Quasi-Minimal Residual Variants of the COCG and COCR Methods for Complex Symmetric Linear Systems in Electromagnetic Simulations

Xian-Ming Gu, Ting-Zhu Huang, Liang Li, Hou-Biao Li, Tomohiro Sogabe, and Markus Clemens, *Senior Member, IEEE*

***Abstract*—The conjugate orthogonal conjugate gradient (COCG) method has been considered an attractive part of the Lanczos-type Krylov subspace method for solving complex symmetric linear systems. However, it is often faced with apparently irregular convergence behaviors in practical electromagnetic simulations. To avoid such a problem, the symmetric quasi-minimal residual (QMR) method has been developed. On the other hand, the conjugate A-orthogonal conjugate residual (COCR) method, which can be regarded as an extension of the conjugate residual method, also had been established. It shows that the COCR often gives smoother convergence behavior than the COCG method. The purpose of this paper is to apply the QMR approaches to the COCG and COCR to derive two new methods (including their preconditioned versions), and to report the benefits of the modified methods by some practical examples arising in electromagnetic simulations.**

***Index Terms*—Complex symmetric matrices, conjugate orthogonal conjugate gradient (COCG), electromagnetic simulations, Krylov subspace methods, quasi-minimal residual (QMR).**

I. Introduction

We are interested in the efficient solutions of nonsingular linear systems of the form $A\boldsymbol{x} = \boldsymbol{b}$, where A is an $N \times N$ complex symmetric matrix ($A = A^T \in \mathbb{C}^{N\times N}$). Such complex-value systems of linear equations, which are strongly required to solve efficiently, often come from many valuable electromagnetic applications such as modeling waveguide discontinuities [1] and simulating the radar cross section (RCS) [2]. In order to solve such systems efficiently, van der Vorst and Mellissen [3] exploited and extended the conjugate gradient (CG) method [4] to establish the well-known conjugate orthogonal conjugate gradient (COCG) method. Meanwhile, Li *et al.* [2], [5] capitalized upon incomplete Cholesky factorization preconditioning strategies combined with the COCG method to solve 3-D electromagnetic scattering problems. Besides, other relatively complicated, but robust solvers such as quasi-minimal residual (QMR) [6], complex symmetric method (CSYM) [7], conjugate A-orthogonal conjugate residual (COCR) [8], [9], symmetric complex bi-conjugate gradient (SCBiCG(Γ, n)) method [10], [11] and bi-conjugate gradient conjugate residual-type (BiCGCR) method [11], [12] are also favorable. QMR is derived from the complex symmetric Lanczos process [6, Algorithm 2.1]. The COCR method proposed as an extension of the conjugate residual (CR) method is derived from the conjugate A-orthogonalization process [9, pp. 54–55]. CSYM is derived from the combination of QMR and tridiagonalization of A by Householder reflections and it does not belong to the Krylov subspace methods (refer to [2] and [7] for this discussion). Moreover, BiCGCR and SCBiCG(Γ, n) can be obtained from the special cases in the bi-conjugate gradient (BiCG) method [13], [14], which has been proposed for solving non-Hermitian linear systems. There are certainly also other iterative methods [14]–[20] for solving complex symmetric linear systems, which we will not pursue here.

The COCG method based on the complex symmetric Lanczos process often exhibits rather irregular convergence behavior, typical for the BiCG method from which it is derived. Moreover, the implicit decomposition of the reduced tridiagonal system may not exist, resulting in breakdown of the algorithm. A related algorithm, the complex symmetric quasi-minimal residual (denoted as QMR-SYM [21, p. 112]) method of Freund [6] can be applied to overcome these problems. The main idea behind this algorithm is to solve the reduced tridiagonal system in a least squares sense, similar to the approach followed in the generalized minimal residual (GMRES) method [22, pp. 164–185]. Since the constructed basis for the Krylov subspace is conjugate orthogonal, rather than orthogonal as in GMRES, the obtained solution is viewed as a QMR solution, which explains the name. Additionally, symmetric QMR uses look-ahead techniques to avoid breakdowns in the underlying complex symmetric Lanczos process, which makes it promise to be more robust than COCG (refer to [6]). Recently, based on a new QMR method with the A-biorthogonalization (QMR_AB) process and the QMR_AB based on the bi-conjugate residual (QMRBi-CR) method are

Manuscript received March 05, 2014; revised June 24, 2014, September 28, 2014; accepted October 19, 2014. Date of publication November 07, 2014; date of current version December 02, 2014.This work was supported by the National Science Foundation of China (NSFC) under Grant 61370147, Grant 61170311, Grant 11101071, Grant 11301057, and Grant 61170309, by the 973 Program (2013CB329404), and by Fundamental Research Funds for the Central Universities (ZYGX2013J106).

X.-M. Gu, T.-Z. Huang, L. Li, and H.-B. Li are with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China (e-mail: guxianming@live.cn; x.m.gu@rug.nl; tingzhuhuang@126.com).

T. Sogabe is with the Graduate School of Information Science and Technology, Aichi Prefectural University, Nagakute, Aichi 480-1198, Japan.

M. Clemens is with the Chair of Electromagnetic Theory, Bergische Universität Wuppertal, Wuppertal 42119, Germany (e-mail: clemens@uni-wuppertal.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMTT.2014.2365472

proposed by Minami *et al.* in [23]. Moreover, it is shown that the QMR based on the BiCG (QMRBi-CG) method [24] and QMRBi-CR are the mathematically equivalent, but numerically improved generalized versions of the QMR and QMR_AB, respectively.

In this paper, we first propose a new QMR method with a conjugate A-orthogonalization process (QMR_CA) method based on the conjugate A-orthogonalization process by recalling Sogabe's derivation of QMR_SYM (referred to as QMR_SYM [9, pp. 21–23]). We then use the strategies of [23] to improve the COCG and COCR for solving complex symmetric linear systems based on the derivations of QMRBi-CG and QMRBi-CR. Since the QMR principle results in a QMR property, the modified COCG and COCR variants, called QMRCOCG and QMRCOCR, respectively, can be expected to give smoother convergence behaviors than COCG and COCR in the residual norm. From a more general point of view, the algorithm of QMRCOCR is also regarded as a particular case in QMRBi-CR, which has been proposed for solving non-Hermitian linear systems, and this is similar to the relation between QMRCOCG and QMRBi-CG. It is very useful to note that QMRCOCG and QMRCOCR are the mathematically equivalent, but numerically improved generalized versions of the QMR_SYM and QMR_AB for solving complex symmetric linear systems, respectively.

This paper is organized as follows. In Section II, we review the principles and derivation of the COCG and QMR_SYM for solving complex symmetric linear systems. Some remarks about the two algorithms are also given there. In Section III, we review the COCR and its orthogonality properties, and then derive the QMR_CA method. In Section IV, we first observe a way to derive the algorithms of preconditioned QMRCOCG (PQMRCOCG). We then derive preconditioned QMRCOCR (PQMRCOCR) from the observation. In Section V, we report some numerical examples arising in some real problems, such as numerical simulation of electromagnetic scattering. Finally, this paper closes with some conclusions in Section VI.

In this paper, the symbol $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ denotes the inner product given by $\sum_{i=1}^{N} \bar{x}_i y_i$, and $\|\boldsymbol{x}\|$ is the 2-norm of vector $\boldsymbol{x}$, i.e., $\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$. The $\bar{A}, A^T$ and $A^H = \bar{A}^T$ denote the complex conjugate, transpose, and conjugate transpose matrices of matrix A. The symbol $\mathcal{K}_n(A, \boldsymbol{r}_0)$ then denotes the n-D Krylov subspace defined by span$\{\boldsymbol{r}_0,\ A\boldsymbol{r}_0, \ldots, A^{n-1}\boldsymbol{r}_0\}$.

II. COCG and QMR_SYM Methods

As we know, both COCG and QMR_SYM are based on the well-known complex symmetric Lanczos process, which is a special case for the Lanczos biorthogonalization process [22, pp. 217–219] when the coefficient matrix is complex symmetric. This algorithm (see, e.g., [6, Algorithm 2.1]) can be given as the following pseudocode.

Algorithm 1: The complex symmetric Lanczos process

1. Set $\beta_0 = 0,\ \boldsymbol{v}_0^L = \boldsymbol{0},\ \boldsymbol{r}_0 \neq \boldsymbol{0} \in \mathbb{C}^N$.
2. Set $\boldsymbol{v}_1^L = \boldsymbol{r}_0 / \|\boldsymbol{r}_0\|$.
3. For $n = 1, 2, \ldots, m-1$, Do:
4. $\quad \boldsymbol{s}_n^L = A\boldsymbol{v}_n^L, \quad \alpha_n = \langle \bar{\boldsymbol{v}}_n^L, \boldsymbol{s}_n^L \rangle$.[1]
5. $\quad \tilde{\boldsymbol{v}}_{n+1}^L = \boldsymbol{s}_n^L - \alpha_n \boldsymbol{v}_n^L - \beta_{n-1}\boldsymbol{v}_{n-1}$.
6. $\quad \beta_n = \langle \tilde{\boldsymbol{v}}_{n+1}^L, \tilde{\boldsymbol{v}}_{n+1}^L \rangle^{1/2}$.
7. $\quad \boldsymbol{v}_{n+1}^L = \tilde{\boldsymbol{v}}_{n+1}^L / \beta_n$.
8. EndDo

If breakdown does not occur at step 7, the above algorithm generates a conjugate orthogonal basis of the Krylov subspace such that $\langle \bar{\boldsymbol{v}}_i^L, \boldsymbol{v}_j^L \rangle = \delta_{ij}$. If we set $V^L = (\boldsymbol{v}_1^L, \ldots, \boldsymbol{v}_n^L)$, similar to the Lanczos biorthogonalization process, Algorithm 1 can also be written in matrix form. The matrix form of Algorithm 1 is given by $AV_n^L = V_{n+1}^L \tilde{T}_n^L = V_n^L T_n^L + \beta_n \boldsymbol{v}_{n+1}^L \boldsymbol{e}_n^T$, where $\boldsymbol{e}_n = (0, 0, \ldots, 1)^T \in \mathbb{R}^n$, $(V_n^L)^T V_n^L = I_n$, $\tilde{T}_n^L$, and T_n^L are the $(n+1) \times n$ and $n \times n$ tridiagonal matrices whose diagonal entries are $\alpha_1, \alpha_2, \ldots$ and subdiagonal entries are $\beta_1, \beta_2, \ldots$ representing the recurrence coefficients of the complex symmetric Lanczos process. From the above, we see that T_n^L is also complex symmetric. In the interest of counteraction against such breakdowns, refer oneself to remedies such as so-called look-ahead techniques [21], [22], which can enhance stability while increasing cost modestly or others, but that is outside the scope of this paper and we shall not pursue that (including other subsequent solvers in this study) here.

A. COCG Method

The COCG method is proposed as a useful Krylov subspace method for solving complex symmetric linear systems, and it is derived from the complex Lanczos process (see Algorithm 1) of Krylov subspace. Here we describe below the COCG algorithm when applied to the system $A\boldsymbol{x} = \boldsymbol{b}$ with a complex symmetric matrix A. Its pseudocodes are given as follows.

Algorithm 2: The COCG method

1. Give an initial guess $\boldsymbol{x}_0^G$ and compute $\boldsymbol{r}_0^G = \boldsymbol{b} - A\boldsymbol{x}_0^G$.
2. Set $\boldsymbol{p}_0^G = \boldsymbol{r}_0^G, \rho_0 = \langle \bar{\boldsymbol{r}}_0^G, \boldsymbol{r}_0^G \rangle$.
3. For $n = 1, 2, \ldots$, until convergence, Do:
4. $\quad \boldsymbol{q}_{n-1}^G = A\boldsymbol{p}_{n-1}^G, \quad \alpha_{n-1} = \rho_{n-1} / \langle \bar{\boldsymbol{p}}_{n-1}^G, \boldsymbol{q}_{n-1}^G \rangle$.
5. $\quad \boldsymbol{x}_n^G = \boldsymbol{x}_{n-1}^G + \alpha_{n-1}\boldsymbol{p}_{n-1}^G$.
6. $\quad \boldsymbol{r}_n^G = \boldsymbol{r}_{n-1}^G - \alpha_{n-1}\boldsymbol{q}_{n-1}^G$.
7. $\quad \rho_n = \langle \bar{\boldsymbol{r}}_n^G, \boldsymbol{r}_n^G \rangle,\ \beta_{n-1} = \rho_n / \rho_{n-1}$.
8. $\quad \boldsymbol{p}_n^G = \boldsymbol{r}_n^G + \beta_{n-1}\boldsymbol{p}_{n-1}^G$.
9. EndDo

If breakdown does not occur at steps 4 or 7, COCG residuals satisfy $\langle \bar{\boldsymbol{r}}_i^G, \boldsymbol{r}_j^G \rangle = 0$ for $i \neq j$. From the above algorithm, we see that if the coefficient matrix is Hermitian, then COCG is equivalent to the CG method [4].

B. QMR_SYM Method

The QMR_SYM method is also useful for solving complex symmetric linear systems, and it can be derived from the complex symmetric Lanczos process described in Algorithm 1. Here, we give the derivation process of the QMR_SYM method.

[1] In our paper, we should specify that the dot product $\langle \bar{\boldsymbol{a}}, \boldsymbol{b} \rangle$ is evaluated via $\boldsymbol{a}^T\boldsymbol{b}$ where $\boldsymbol{a}$ and $\boldsymbol{b}$ are two arbitrary complex vectors.

Let V_n be the conjugate orthonormal basis of $\mathcal{K}_n(A, \boldsymbol{r}_0)$. Since $\boldsymbol{x}_n$ lies in the affine space $\boldsymbol{x}_0 + \mathcal{K}_n(A, \boldsymbol{r}_0)$, we then have

$$\boldsymbol{x}_n = \boldsymbol{x}_0 + V_n^L \boldsymbol{y}_n \quad \boldsymbol{y}_n \in \mathbb{C}^n. \tag{1}$$

The corresponding residual vector is $\boldsymbol{r}_n = \boldsymbol{r}_0 - AV_n^L \boldsymbol{y}_n$. From the complex symmetric Lanczos process, it follows that

$$\begin{aligned} \|\boldsymbol{r}_n\| &= \left\| \boldsymbol{r}_0 - V_{n+1}^L \tilde{T}_n^L \boldsymbol{y}_n \right\| \\ &= \left\| V_{n+1}^L \left(\boldsymbol{g}^{(n+1)} - \tilde{T}_n^L \boldsymbol{y}_n \right) \right\| \\ &= \left\| V_{n+1}^L (\Omega_{n+1}^L)^{-1} \left(\omega_1 \boldsymbol{g}^{(n+1)} - \Omega_{n+1}^L \tilde{T}_n^L \boldsymbol{y}_n \right) \right\| \end{aligned} \tag{2}$$

where $\boldsymbol{g}^{(n+1)} = \|\boldsymbol{r}_0\| \cdot (1, 0, \ldots, 0)^T \in \mathbb{R}^{n+1}$ and Ω_{n+1} is defined as the scaling matrix $\Omega_{n+1}^L = \mathrm{diag}(\omega_1, \omega_2, \ldots, \omega_{n+1})$, $\omega_j > 0, j = 1, 2 \ldots, n+1$. In fact, we often take $\omega_j \equiv 1$ $(j = 1, 2, \ldots)$ for convenience.

We see that the above derivation process is very similar to the one of the minimal residual (MINRES) method [29]. However, in this case, if we choose $\boldsymbol{y}_n$ such that the norm of the residual is minimized, then it may lead to a large amount of computational costs due to $(V_n^L)^H V_n^L \neq I_n$. Instead, QMR_SYM chooses $\boldsymbol{y}_n$ such that

$$\boldsymbol{y}_n := \arg \min_{\boldsymbol{y} \in \mathbb{C}^n} \left\| \omega_1 \boldsymbol{g}^{(n+1)} - \Omega_{n+1}^L \tilde{T}_n^L \boldsymbol{y} \right\|. \tag{3}$$

Here, we note that if V_n^L satisfies the relation $(V_n^L)^H V_n^L = I_n$, then the above choice leads to minimization of the norm of the residuals. Unfortunately, when matrix A is complex symmetric, V_n^L does not satisfy the relation $(V_n^L)^H V_n = I_n$ except in some special cases (see [9, p. 22]). To achieve the minimization (3), the Givens rotations G_j $(j = 1, \ldots, n)$ described in [25] play an important role. Multiplying $\omega_1 \boldsymbol{g}^{(n+1)} - \Omega_{n+1}^L \tilde{T}_n^L \boldsymbol{y}_n$ via $Q_n = G_n \cdots G_1$ such that $Q_n^H Q_n = I_n$ and $Q_n(\Omega_{n+1}^L \tilde{T}_n^L) = R_n$, we obtain $\min_{\boldsymbol{y} \in \mathbb{C}^n} \|\omega_1 \boldsymbol{g}^{(n+1)} - \Omega_{n+1}^L \tilde{T}_n^L \boldsymbol{y}\| = \min_{\boldsymbol{y} \in \mathbb{C}^n} \|\boldsymbol{\zeta}_n - R_n \boldsymbol{y}\|$, where $\boldsymbol{\zeta}_n = \omega_1 Q_n \boldsymbol{g}^{(n+1)}$. Thus, we have $\boldsymbol{y}_n = R_n^{-1} \boldsymbol{\zeta}_n$. From this and (1), we obtain the desired approximate solution $\boldsymbol{x}_n = \boldsymbol{x}_0 + V_n^L R_n^{-1} \boldsymbol{\zeta}_n$. It is then not hard to obtain the pseudocode of this practical algorithm, but we do not pursue that here (refer to [9, pp. 21–23] for details).

III. COCR AND QMR_CA

Analogically, the COCR method can be derived from the conjugate A-orthogonalization process, which is similar to the complex symmetric Lanczos process. The conjugate A-orthogonalization process of $\mathcal{K}_n(A, \boldsymbol{r}_0)$ and $\mathcal{K}_n(\bar{A}, \bar{\boldsymbol{r}}_0)$ can then be reformulated below.

Algorithm 3: The conjugate A-orthogonalization process

1. Set $\beta_0 = 0$, $\boldsymbol{v}_0^A = \boldsymbol{0}$, $\boldsymbol{r}_0 \neq \boldsymbol{0} \in \mathbb{C}^N$.
2. Set $\boldsymbol{v}_1^A = \boldsymbol{r}_0 / \|\boldsymbol{r}_0\|$, $\boldsymbol{t}_1^A = A\boldsymbol{v}_1^A$.
3. For $n = 1, 2, \ldots,$ Do:
4. $\quad \alpha_n = \langle \bar{\boldsymbol{t}}_n^A, \boldsymbol{t}_n^A \rangle, \tilde{\boldsymbol{v}}_{n+1}^A = \boldsymbol{t}_n^A - \alpha_n \boldsymbol{v}_n^A - \beta_{n-1} \boldsymbol{v}_{n-1}^A.$
5. $\quad \boldsymbol{q}_n^A = A\boldsymbol{t}_n^A, \ \tilde{\boldsymbol{u}}_{n+1}^A = \boldsymbol{q}_n^A - \alpha_n \boldsymbol{t}_n^A - \beta_{n-1} \boldsymbol{t}_{n-1}^A.$
6. $\quad \beta_n = \langle \bar{\tilde{\boldsymbol{v}}}_{n+1}^A, \tilde{\boldsymbol{u}}_{n+1}^A \rangle^{\frac{1}{2}}.$
7. $\quad \boldsymbol{v}_{n+1}^A = \tilde{\boldsymbol{v}}_{n+1}^A / \beta_n, \ \boldsymbol{t}_{n+1}^A = \tilde{\boldsymbol{q}}_{n+1}^A / \beta_n.$
8. EndDo

From Algorithm 3 it is clear that if breakdown does not occur at step 7, the vector sequences $\boldsymbol{v}_0^A, \boldsymbol{v}_1^A, \ldots$ satisfy the following conjugate A-orthogonality property: $\langle \bar{\boldsymbol{v}}_i^A, A\boldsymbol{v}_j^A \rangle = \delta_{ij}$. If we set $V^A = (\boldsymbol{v}_1^A, \ldots, \boldsymbol{v}_n^A)$, similar to the complex symmetric Lanczos process, Algorithm 3 can be also written in matrix form, i.e., $AV_n^A = V_{n+1}^A \tilde{T}_n^A = V_n^A T_n^A + \beta_n \boldsymbol{v}_{n+1}^A \boldsymbol{e}_n^T$, where $\boldsymbol{e}_n = (0, 0, \ldots, 1)^T \in \mathbb{R}^n$, $(V_n^L)^T A V_n^L = I_n$, $\tilde{T}_n^A$, and T_n^A are the $(n+1) \times n$ and $n \times n$ tridiagonal matrices whose diagonal entries are $\alpha_1, \alpha_2, \ldots$ and subdiagonal entries are $\beta_1, \beta_2, \ldots$ that are recurrence coefficients of the conjugate A-orthogonalization process. From the above, we note that T_n^A is also complex symmetric.

A. COCR Method

The COCR method is an efficient Krylov subspace method for solving complex symmetric linear systems, and it is derived from the conjugate A-orthogonalization process (see Algorithm 3) of the Krylov subspace. Here we describe below the COCR algorithm when applied to the system $A\boldsymbol{x} = \boldsymbol{b}$ with a complex symmetric matrix A.

Algorithm 4: The COCR method

1. Give an initial guess $\boldsymbol{x}_0^R$ and compute $\boldsymbol{r}_0^R = \boldsymbol{b} - A\boldsymbol{x}_0^R$.
2. Set $\boldsymbol{p}_0^R = \boldsymbol{r}_0^R, \boldsymbol{u}_0^R = A\boldsymbol{p}_0^R, \boldsymbol{s}_0^R = \boldsymbol{u}_0^R, \ \rho_0 = \langle \bar{\boldsymbol{r}}_0^R, \boldsymbol{s}_0^R \rangle.$
3. For $n = 1, 2, \ldots,$ until convergence, Do:
4. $\quad \alpha_{n-1} = \rho_{n-1} / \langle \bar{\boldsymbol{u}}_{n-1}^R, \boldsymbol{u}_{n-1}^R \rangle.$
5. $\quad \boldsymbol{x}_n^R = \boldsymbol{x}_{n-1}^R + \alpha_{n-1} \boldsymbol{p}_{n-1}^R.$
6. $\quad \boldsymbol{r}_n^R = \boldsymbol{r}_{n-1}^R - \alpha_{n-1} \boldsymbol{u}_{n-1}^R.$
7. $\quad \boldsymbol{s}_n^R = A\boldsymbol{r}_n^R, \ \rho_n = \langle \bar{\boldsymbol{r}}_n^R, \boldsymbol{s}_n^R \rangle, \ \beta_{n-1} = \rho_n / \rho_{n-1}.$
8. $\quad \boldsymbol{p}_n^R = \boldsymbol{r}_n^R + \beta_{n-1} \boldsymbol{p}_{n-1}^R.$
9. $\quad \boldsymbol{u}_n^R = \boldsymbol{s}_n^R + \beta_{n-1} \boldsymbol{u}_{n-1}^R.$
10. EndDo

It is shown in [8] and [9] that if breakdown does not occur at steps 4 or 7, then the nth residual vector of COCR satisfies $\boldsymbol{r}_n^R \perp \bar{A}\mathcal{K}_n(\bar{A}, \bar{\boldsymbol{r}}_0^R)$, which leads to A-conjugate orthogonality $\langle \bar{\boldsymbol{r}}_i^R, A\boldsymbol{r}_j^R \rangle = 0$, for $i \neq j$. The COCR method corresponds to the CR method [22, p. 194] when the coefficient matrix A is Hermitian.

B. QMR_CA Method

The QMR_CA method based on the conjugate A-orthogonalization process can be derived using a similar strategy of QMR_SYM, which is based on the complex symmetric Lanczos process. More precisely, the corresponding residual vector is $\boldsymbol{r}_n = \boldsymbol{r}_0 - AV_n^A \boldsymbol{y}_n$ and scaling matrix Ω_{n+1}^A. From the conjugate A-orthogonalization process, it follows that

$$\begin{aligned} \boldsymbol{r}_n &= \boldsymbol{r}_0 - AV_n^A \boldsymbol{y}_n \\ &= \boldsymbol{r}_0 - V_{n+1}^A \tilde{T}_n^A \boldsymbol{y}_n \\ &= V_{n+1}^A \left(\Omega_{n+1}^A \right)^{-1} \left(\omega_1 \boldsymbol{g}_1^{(n+1)} - \Omega_{n+1}^A \tilde{T}_n^A \boldsymbol{y}_n \right). \end{aligned} \tag{4}$$

It is natural to determine $\boldsymbol{y}_n$ such that all residual norm $\|\boldsymbol{r}_n\|$ are minimized. However, this choice is not useful in practical implementation due to large computational costs and memory requirement. Hence, the vector $\boldsymbol{y}_n$ is determined via using an

alternative approach, i.e., by solving the following least squares problems:

$$\boldsymbol{y}_n := \arg\min_{\boldsymbol{y}\in\mathbb{C}^n} \left\| \omega_1 \boldsymbol{g}_1^{(n+1)} - \Omega_{n+1}^A \tilde{T}_n^A \boldsymbol{y} \right\|. \tag{5}$$

It can obtain the pseudocode of the QMR_CA method via using the similar derivation of the QMR_AB method [23]. For the sake of brevity, we do not give the detailed pseudocode of QMQ_CA here.

IV. QMRCOCG AND QMRCOCR

In Section III, we have described both the QMR_SYM method and QMR_CA method, the accuracy of them is often not as good as the COCG method and COCR method as can be shown in numerical comparison experiments. This is mainly because both the QMR_SYM method and QMR_CA method are based on the three-term recurrence. Comparing those with the coupled two-term recurrence-based algorithms, the algorithms based on the three-term recurrences are sensitively influenced by roundoff errors [23], [26]. Thus, the QMR_SYM method and QMR_CA method are rarely used in practical numerical simulations. In this section, firstly the new QMRCOCG method can be derived from the coupled two-term recurrence-based COCG method, and then we will derive the efficient QMRCOCR method from the COCR method.

A. QMRCOCG Method

It is well known that the approximate solution $\boldsymbol{x}_n^G$ obtained by the COCG method can be written as

$$\boldsymbol{x}_n^G = \boldsymbol{x}_0 + V_n^L \boldsymbol{y}_n^G \tag{6}$$

by using the Petrov–Galerkin method, we have

$$\begin{aligned} \boldsymbol{0} &= \left(V_n^L\right)^T \boldsymbol{r}_n^G \\ &= \left(V_n^L\right)^T \boldsymbol{r}_0 - \left(V_n^L\right)^T A V_n^L \boldsymbol{y}_n^G \\ &= \left(V_n^L\right)^T \|\boldsymbol{r}_0\| \boldsymbol{v}_1 - T_n^L \boldsymbol{y}_n^G = \boldsymbol{g}_1^{(n)} - T_n^L \boldsymbol{y}_n^G \end{aligned}$$

where $\boldsymbol{g}_1^{(n)} = \|\boldsymbol{r}_0\| \ldots (1, 0, \cdots, 0)^T \in \mathbb{R}^n$. According to the above identity, we obtain the form of $\boldsymbol{y}_n^G$

$$\boldsymbol{y}_n^G = \left(T_n^L\right)^{-1} \boldsymbol{g}_1^{(n)}. \tag{7}$$

From the conclusion of [27, Lemma 4.1] proposed by Freund, it is not hard to prove that $\boldsymbol{y}_n^{QG}$ (3) of the QMRCOCG method and $\boldsymbol{y}_n^G$ (7) of the COCG method are connected by

$$\boldsymbol{y}_n^{QG} = \left(1 - c_n^2\right) \boldsymbol{y}_{n-1}^{QG} + c_n^2 \boldsymbol{y}_n^G \tag{8}$$

where the coefficients are defined by

$$c_n = \frac{1}{\sqrt{1+\theta_n^2}} \quad \theta_n = \frac{1}{\tau_{n-1}} \left\| \omega_1 \boldsymbol{g}_1^{(n+1)} - \Omega_{n+1}^L \tilde{T}_n^L \boldsymbol{y}_n^G \right\|. \tag{9}$$

Moreover, the scalar coefficients τ_i can be updated from step to step by setting

$$\tau_0 := \omega_1 \|\boldsymbol{r}_0\|, \text{ and } \tau_n = \tau_{n-1} \theta_n c_n, \qquad n = 1, 2, \ldots. \tag{10}$$

We remark that c_n is just the cosine of the nth Given rotation in an QR decomposition of the upper Hessenberg matrix $\Omega_{n+1}^L \tilde{T}_n^L$.

The norm in the second equation in (9) is directly connected with the norm of the COCG residual vector, and it holds

$$\left\| \omega_1 \boldsymbol{g}_1^{(n+1)} - \Omega_{n+1}^L \tilde{T}_n^L \boldsymbol{y}_n^G \right\| = \omega_{n+1} \left\| \boldsymbol{r}_n^G \right\| / \left\| v_{n+1}^L \right\|. \tag{11}$$

To verify (11), note that, from (2) and (7), we have

$$\begin{aligned} &\omega_1 \boldsymbol{g}_1^{(n+1)} - \Omega_{n+1}^L \tilde{T}_n^L \boldsymbol{y}_n^G \\ &\quad = \omega_1 \boldsymbol{g}_1^{(n+1)} - \Omega_{n+1}^L T_n^L \left(T_n^L\right)^{-1} \boldsymbol{g}_1^{(n+1)} \\ &\qquad - \omega_{n+1} \xi_n \left\langle \boldsymbol{y}_n^G, \boldsymbol{e}_n^{(n)} \right\rangle \boldsymbol{e}_{n+1}^{(n+1)} \\ &\qquad = \delta_n \boldsymbol{e}_{n+1}^{(n+1)} \end{aligned} \tag{12}$$

where $\delta_n := -\omega_{n+1} \xi_n \left\langle \boldsymbol{y}_n^G, \boldsymbol{e}_n^{(n)} \right\rangle$ and

$$\begin{aligned} \boldsymbol{r}_n^G &= V_{n+1}^L \left(\Omega_{n+1}^L\right)^{-1} \left(\omega_1 \boldsymbol{g}_1^{(n+1)} - \Omega_{n+1}^L \tilde{T}_n^L \boldsymbol{y}_n^G \right) \\ &= \frac{\delta_n}{\omega_{n+1}} \boldsymbol{v}_{n+1}^L. \end{aligned} \tag{13}$$

The relation (11) then follows via taking the norms in (12) and (13).

Next, we show how the QMR_SYM iterates can be updated by means of quantities from the COCG algorithm. In the following, let $n \geq 1$. With (6) and (8), we obtain that

$$\boldsymbol{x}_n^{QG} = \left(1 - c_n^2\right) \boldsymbol{x}_{n-1}^{QG} + c_n^2 \boldsymbol{x}_n^G. \tag{14}$$

We then set $\boldsymbol{d}_n^{QG} := \boldsymbol{x}_n^{QG} - \boldsymbol{x}_{n-1}^{QG}$, the solution of QMRCOCG can be represented by following identity:

$$\boldsymbol{x}_n^{QG} = \boldsymbol{x}_{n-1}^{QG} + \boldsymbol{d}_n^{QG}. \tag{15}$$

Using (14), the definition of $\boldsymbol{d}_n^{QG}$, and the relation $\boldsymbol{x}_n^G = \boldsymbol{x}_{n-1}^G + \alpha_{n-1} \boldsymbol{p}_{n-1}^G$ of the COCG method (see Algorithm 2), it follows that

$$\begin{aligned} \boldsymbol{x}_n^{QG} &= -c_n^2 \boldsymbol{x}_{n-1}^{QG} + c_n^2 \left(\boldsymbol{x}_{n-1}^G + \alpha_{n-1} \boldsymbol{p}_{n-1}^G \right) \\ &= c_n^2 \left(\boldsymbol{x}_{n-1}^G - \boldsymbol{x}_{n-1}^{QG} \right) + c_n^2 \alpha_{n-1} \boldsymbol{p}_{n-1}^G \\ &= c_n^2 \theta_{n-1}^2 \boldsymbol{d}_{n-1}^{QG} + c_n^2 \alpha_{n-1} \boldsymbol{p}_{n-1}^G. \end{aligned} \tag{16}$$

The residual vector can then be updated by (14) as

$$\begin{aligned} \boldsymbol{r}_n^{QG} &= \boldsymbol{b} - A \boldsymbol{x}_n^{QG} \\ &= \left(1 - c_n^2\right) \left(\boldsymbol{b} - A \boldsymbol{x}_{n-1}^{QG} \right) + c_n^2 \left(\boldsymbol{b} - A \boldsymbol{x}_n^G \right) \\ &= \left(1 - c_n^2\right) \boldsymbol{r}_{n-1}^{QG} + c_n^2 \boldsymbol{r}_n^G. \end{aligned} \tag{17}$$

Therefore, by putting these relations (8)–(10) and (15)–(17) in the classical COCG algorithm, we can obtain the following QMRCOCG method without look-ahead. For efficient implementation, we directly give the preconditioned version of resulted algorithm (see Algorithm 5) by using the derivation of the preconditioned conjugate gradient (PCG) method [22, pp. 262–265]. Here we added some auxiliary quantities to reduce the floating point arithmetic for the square of scalar coefficients, i.e., $\tilde{\theta}_n := \theta_n^2,\ \tilde{c}_n := c_n^2$ and $\tilde{\tau}_n := \tau_n^2$.

Algorithm 5: The preconditioned QMRCOCG method

1. Give an initial guess $\boldsymbol{x}_0^{QG}$, then $\boldsymbol{r}_0^{QG} := \boldsymbol{b} - A\boldsymbol{x}_0^{QG}$.
2. Set $\boldsymbol{r}_0^G = \boldsymbol{r}_0^{QG}$ and solve $M\boldsymbol{z}_0^G = \boldsymbol{r}_0^G$. % M is the preconditioner
3. Set $\boldsymbol{p}_0^G = \boldsymbol{z}_0^G,\ \boldsymbol{d}_0^{QG} = \boldsymbol{0},\ \tilde{\tau}_0 = \omega_1^2\langle \boldsymbol{r}_0^G, \boldsymbol{z}_0^G\rangle,\ \tilde{\theta}_0 = 0,\ \rho_0 = \langle \bar{\boldsymbol{r}}_0^G, \boldsymbol{z}_0^G\rangle$.
4. For $n = 1, 2, \ldots$, until convergence, Do:
   (COCG part1)
5. $\boldsymbol{q}_{n-1}^G = A\boldsymbol{p}_{n-1}^G,\ \ \alpha_{n-1} = \rho_{n-1}/\langle \bar{\boldsymbol{p}}_{n-1}^G, \boldsymbol{q}_{n-1}^G\rangle$.
6. $\boldsymbol{r}_n^G = \boldsymbol{r}_{n-1}^G - \alpha_{n-1}\boldsymbol{q}_{n-1}^G$.
7. Solve $M\boldsymbol{z}_n^G = \boldsymbol{r}_n^G$.
   (QMR approach)
8. $\tilde{\theta}_n = \frac{\omega_{n+1}^2\langle \boldsymbol{r}_n^G, \boldsymbol{z}_n^G\rangle}{\tilde{\tau}_{n-1}},\quad \tilde{c}_n = \frac{1}{1+\tilde{\theta}_n},\quad \tilde{\tau}_n = \tilde{\tau}_{n-1}\tilde{\theta}_n\tilde{c}_n$.
9. $\boldsymbol{d}_n^{QG} = \tilde{c}_n\tilde{\theta}_{n-1}\boldsymbol{d}_{n-1}^{QG} + \tilde{c}_n\alpha_{n-1}\boldsymbol{p}_{n-1}^G$.
10. $\boldsymbol{x}_n^{QG} = \boldsymbol{x}_{n-1}^{QG} + \boldsymbol{d}_n^{QG}$.
11. $\boldsymbol{r}_n^{QG} = (1-\tilde{c}_n)\boldsymbol{r}_{n-1}^{QG} + \tilde{c}_n\boldsymbol{r}_n^G$.
    (COCG part2)
12. $\rho_n = \langle \bar{\boldsymbol{r}}_n^G, \boldsymbol{z}_n^G\rangle,\quad \beta_{n-1} = \rho_n/\rho_{n-1}$.
13. $\boldsymbol{p}_n^G = \boldsymbol{z}_n^G + \beta_{n-1}\boldsymbol{p}_{n-1}^G$.
14. EndDo

We remark that, except for the additional updated in-lines 8–11, this algorithm is just the classical preconditioned COCG (PCOCG) method [11].

B. QMRCOCR Method

The approximate solution $\boldsymbol{x}_n^G$ obtained by the COCR method satisfies the following identity $\boldsymbol{r}_n^R = \boldsymbol{r}_0 - AV_n^A\boldsymbol{y}_n^R$ via using the Petrov–Galerkin method, and we have

$$\begin{aligned}\boldsymbol{0} &= \left(V_n^A\right)^T A\boldsymbol{r}_n^R \\ &= \left(V_n^A\right)^T A\boldsymbol{r}_0 - \left(V_n^A\right)^T A^2 V_n^A\boldsymbol{y}_n^R \\ &= \left(V_n^A\right)^T A\|\boldsymbol{r}_0\|\boldsymbol{v}_1^A - T_n^A\boldsymbol{y}_n^R \\ &= \boldsymbol{g}_1^{(n)} - T_n^A\boldsymbol{y}_n^R.\end{aligned}$$

According to the above identity, we obtain the form of $\boldsymbol{y}_n^G$

$$\boldsymbol{y}_n^R = \left(T_n^A\right)^{-1}\boldsymbol{g}_1^{(n)}. \tag{18}$$

Moreover, it follows that

$$\begin{aligned}&\omega_1\boldsymbol{g}_1^{(n+1)} - \Omega_{n+1}^A\tilde{T}_n^A\boldsymbol{y}_n^R \\ &\quad= \omega_1\boldsymbol{g}_1^{(n+1)} - \Omega_{n+1}^A T_n^A\left(T_n^A\right)^{-1}\boldsymbol{g}_1^{(n+1)} \\ &\qquad- \omega_{n+1}\xi_n\left\langle \boldsymbol{y}_n^R, \boldsymbol{e}_n^{(n)}\right\rangle \boldsymbol{e}_{n+1}^{(n+1)} \\ &\quad= \delta_n\boldsymbol{e}_{n+1}^{(n+1)}\end{aligned} \tag{19}$$

where $\delta_n := -\omega_{n+1}\xi_n\langle \boldsymbol{y}_n^R, \boldsymbol{e}_n^{(n)}\rangle$. From (4), we obtain that

$$\begin{aligned}\boldsymbol{r}_n^R &= V_{n+1}^A\left(\Omega_{n+1}^A\right)^{-1}\left(\omega_1\boldsymbol{g}_1^{(n+1)} - \Omega_{n+1}^A\tilde{T}_n^A\boldsymbol{y}_n^R\right) \\ &= \frac{\delta_n}{\omega_{n+1}}\boldsymbol{v}_{n+1}^A.\end{aligned} \tag{20}$$

It then also follows that

$$\left\|\omega_1\boldsymbol{g}_1^{(n+1)} - \Omega_{n+1}^A\tilde{T}_n^A\boldsymbol{y}_n^R\right\| = \omega_{n+1}\left\|\boldsymbol{r}_n^R\right\|/\|v_{n+1}^A\|. \tag{21}$$

Moreover, by using a similar strategy of derivation of the QMRCOCG method, we can put (9) and (10) and (15)–(17) in the COCR algorithm and obtain the following QMRCOCR method. For efficient implementation, we directly give the preconditioned version of the resulted algorithm (see Algorithm 6) by derived strategy of the PCG method.

Algorithm 6: The preconditioned QMRCOCR method

1. Give an initial guess $\boldsymbol{x}_0^{QR}$, then $\boldsymbol{r}_0^{QR} := \boldsymbol{b} - A\boldsymbol{x}_0^{QR}$.
2. Set $\boldsymbol{r}_0^R = \boldsymbol{r}_0^{QR}$ and solve $M\boldsymbol{z}_0^R = \boldsymbol{r}_0^R$. % M is the preconditioner
3. Set $\boldsymbol{p}_0^R = \boldsymbol{z}_0^R, \boldsymbol{d}_0^{QR} = \boldsymbol{0}, \tilde{\tau}_0 = \omega_1^2\langle \boldsymbol{r}_0^R, \boldsymbol{z}_0^R\rangle, \boldsymbol{u}_0^R = A\boldsymbol{p}_0^R, \theta_0 = 0, \boldsymbol{s}_0^R = \boldsymbol{u}_0^R, \rho_0 = \langle \boldsymbol{r}_0^R, \boldsymbol{s}_0^R\rangle$ and solve $M\boldsymbol{t}_0^R = \boldsymbol{u}_0^R$.
4. For $n = 1, 2, \ldots$, until convergence, Do:
   (COCR part1)
5. $\alpha_{n-1} = \rho_{n-1}/\langle \bar{\boldsymbol{u}}_{n-1}^R, \boldsymbol{t}_{n-1}^R\rangle$.
6. $\boldsymbol{r}_n^R = \boldsymbol{r}_{n-1}^R - \alpha_{n-1}\boldsymbol{u}_{n-1}^R$.
7. $\boldsymbol{z}_n^R = \boldsymbol{z}_{n-1}^R - \alpha_{n-1}\boldsymbol{t}_{n-1}^R$.
   (QMR approach)
8. $\tilde{\theta}_n = \frac{\omega_{n+1}^2\langle \boldsymbol{r}_n^R, \boldsymbol{z}_n^R\rangle}{\tilde{\tau}_{n-1}},\ \tilde{c}_n = \frac{1}{1+\tilde{\theta}_n},\quad \tilde{\tau}_n = \tilde{\tau}_{n-1}\tilde{\theta}_n\tilde{c}_n$.
9. $\boldsymbol{d}_n^{QR} = \tilde{c}_n\tilde{\theta}_{n-1}\boldsymbol{d}_{n-1}^{QR} + \tilde{c}_n\alpha_{n-1}\boldsymbol{p}_{n-1}^R$.
10. $\boldsymbol{x}_n^{QR} = \boldsymbol{x}_{n-1}^{QR} + \boldsymbol{d}_n^{QR}$.
11. $\boldsymbol{r}_n^{QR} = (1-\tilde{c}_n)\boldsymbol{r}_{n-1}^{QR} + \tilde{c}_n\boldsymbol{r}_n^R$.
    (COCR part2)
12. $\boldsymbol{s}_n^R = A\boldsymbol{z}_n^R,\ \rho_n = \langle \bar{\boldsymbol{z}}_n^R, \boldsymbol{s}_n^R\rangle,\ \beta_{n-1} = \rho_n/\rho_{n-1}$.
13. $\boldsymbol{p}_n^R = \boldsymbol{z}_n^R + \beta_{n-1}\boldsymbol{p}_{n-1}^R$.
14. $\boldsymbol{u}_n^R = \boldsymbol{s}_n^R + \beta_{n-1}\boldsymbol{u}_{n-1}^R$.
15. EndDo

We remark that except for the additional updated in-line 8–11, this algorithm is just the classical preconditioned COCR (PCOCR) method [9, p. 53].

At the end of this section, we show the storage and computational cost for COCG, QMRCOCG, COCR, QMRCOCR, and complex symmetric MINRES using the QLP decomposition (CS-MINRES-QLP) [20] in Table I. "Vectors" denotes the number of vectors that are required to save per iteration step. "MVs" denotes the number of matrix-vector products in each iteration step. "4 or 5" denotes "4" for the unpreconditioned

TABLE I
SUMMARY OF THE COMPUTATIONAL COSTS OF DIFFERENT ITERATIVE SOLVERS IN EACH ITERATION STEP

Solver	Dot product	$\alpha\boldsymbol{x}$	$\boldsymbol{x} \pm \boldsymbol{y}$	Vectors	MVs	Preconditioner solver
COCG	2	3	3	4	1	1
QMRCOCG	3	6	5	7	1	1
COCR	2	4 or 5	4 or 5	5 or 6	1	1
QMRCOCR	3	6 or 7	6 or 7	7 or 8	1	1
QMR-SYM	2	7	5	6	1	1
CS-MINRES-QLP	2	11	8	8	1	1

COCR and "5" for the preconditioned one as well as other similar conventions.

V. NUMERICAL EXPERIMENTS

In this section, we report three numerical examples with COCG, QMRCOCG, COCR, QMRCOCR, QMR-SYM, and CS-MINRES-QLP. We evaluate all methods in aspects of the number of iterations (referred to as *Iters*), CPU consuming time in seconds (referred to as CPU), and $\log_{10}$ of true relative residual 2-norm (referred to as *TRR*) defined as $\log_{10}\|\boldsymbol{b}-A\boldsymbol{x}_n\|/\|\boldsymbol{b}-A\boldsymbol{x}_0\|$. All the tests are performed under Windows XP (64-bit) and MATLAB R2010a running on a DELL desktop workstation with an Intel Xeon CPU E5504 at 2.0 GHz and 48 GB of memory. In all cases, the iteration was started with $\boldsymbol{x}_0 = \boldsymbol{0}$, and the stopping criterion was $\|\boldsymbol{r}_n\|_2/\|\boldsymbol{r}_0\|_2 < 10^{-6}$. Here we only add the simple $\mathrm{SSOR}(\tau)$ preconditioners [22, p. 285] to ensure the convergence within a reasonable amount of iterations in all examples. Moreover, we set the weighted coefficients $\omega_j \equiv 1\ (j = 1, 2, \ldots)$ for simplicity. The convergence plots are shown $\log_{10}$ of the relative residual 2-norm, $\log_{10}\|\boldsymbol{r}_n\|/\|\boldsymbol{b}\|$ (on the vertical axis) versus iterations (on the horizontal axis).

Example 1: We first consider a model problem consisting of the propagation of a plane wave in the vacuum. It can be governed via using the following 3-D time–harmonic Maxwell's equations

$$\begin{cases} \mathrm{i}\omega\varepsilon_r\mathbf{E} - \mathbf{curl}\mathbf{H} = -\mathbf{J}, & \text{in } \Omega, \quad \mathrm{i} = \sqrt{-1} \\ \mathrm{i}\omega\mu_r\mathbf{H} + \mathbf{curl}\mathbf{E} = 0, & \text{in } \Omega \\ \mathbf{n}\times\mathbf{E} = 0, & \text{on } \Gamma_m \\ \mathbf{n}\times\mathbf{E} + \mathbf{n}\times(\mathbf{n}\times\mathbf{H}) = \mathbf{n}\times\mathbf{E}^{\mathrm{int}} + \mathbf{n}\times(\mathbf{n}\times\mathbf{H}^{\mathrm{inc}}), & \text{on } \Gamma_a \end{cases}$$

where the more details of these notations $\mathbf{n}$, $\mathbf{E}$, $\mathbf{H}$, $\mathbf{J}$, and $(\mathbf{E}^{\mathrm{int}}, \mathbf{H}^{\mathrm{inc}})$ can be found in [30]. The boundary of the computational domain Ω is $\partial\Omega = \Gamma_m \cup \Gamma_a$ with $\Gamma_m \cap \Gamma_a = \emptyset$. The boundary condition on Γ_m indicates a metallic boundary condition (i.e., a perfect electrical conductor (PEC) boundary condition), while the second relation states a Silver–Müller condition on Γ_a. Here, the computational domain is chosen to be the unit cube $\Omega = (-0.5; 0.5)^3$ and the Silver–Müller absorbing boundary condition is imposed on the whole $\partial\Omega$. The electromagnetic parameters ε_r and μ_r are set to be 1 everywhere, and the angular frequency is $\omega = 2\pi$. Finally, the penalty parameter τ is set to be 1.2. The "HDG-P_2" method [30] is used to discrete the above Maxwell's equations. We take a system of linear equations with size 33696 and 2643840 nonzero elements in this example (see [30] for details of this model). The numerical results of different iterative solvers with SSOR(0.8) preconditioners are reported in Table II.

TABLE II
NUMERICAL RESULTS FOR EXAMPLE 1

Solver	*Iters*	*TRR*	CPU (s)
COCG	1301	-6.4222	63.92
QMRCOCG	1199	-6.0193	60.96
COCR	1168	-6.0303	58.76
QMRCOCR	1158	-6.0024	61.06
QMR-SYM	1199	-6.0146	90.14
CS-MINRES-QLP	†	†	†

Firstly, the "†" (similarly hereinafter) means that the CS-MINRES-QLP fails to accept the indefinite preconditioners and converge to the desired residual tolerance. According to Table II, it shows that the good potential of the QMR strategy to smooth the residual and to improve the performance of the PCOCG and PCOCR to some extent. Moreover, the preconditioned QMRCOCR method (PQMRCOCR) with smooth convergence outperforms the other methods in terms of the number of iterations, whereas it needs more CPU time than PCOCR and preconditioned QMRCOCG method (PQMRCOCG). The PCOCR with irregular convergence is the best solver among all other methods in terms of CPU time. Compared with PCOCR, the accelerating benefit of PQMRCOCR does not seem very evident, but it still converges faster than PCOCG and the preconditioned QMR-SYM (PQMR-SYM) method in aspects of CPU time. Let us show the convergence behaviors of these methods in Fig. 1. From Fig. 1, we observe that PQMRCOCG and PQMRCOCR smoothed the convergence behaviors of PCOCG and PCOCR and the 2-norms of relative residuals per iteration step were smaller than those of PCOCG and PCOCR. The most remarkable thing is that the convergence curves of PQMRCOCG and PQMRCOCR were almost monotone decreasing. Take full account of the numerical results, the PQMRCOCG and PQMRCOCR are considerably recommended solving the complex symmetric linear systems in Example 1.

Example 2: Secondly, we calculate the RCS of a metallic sphere coated via a lossy dielectric layer using an edge-based finite-element method (FEM). The radius of the metallic sphere is 0.8 cm. The relative complex permittivity of the lossy dielectric layer is taken as $\epsilon_r = 4 - \mathrm{i}$ with a thickness of 0.2 cm. The incident plane wave is taken as x-polarized with incident angles $\phi = 0°$ and $\theta = 180°$ and its frequency is 3 GHz. Due to the symmetry of the fields and geometry, one-half of the structure is applied for the numerical model. The first-order tetrahedral element is exploited to discrete the computational

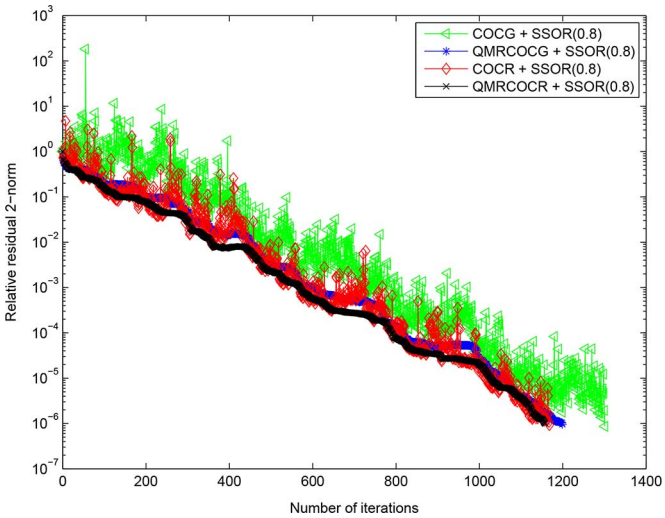


Fig. 1. Convergence histories of different iterative solvers with SSOR(0.8) preconditioners for Example 1.

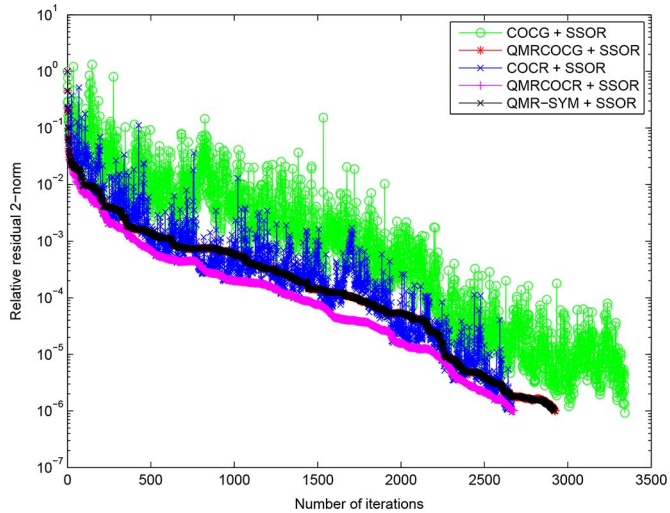


Fig. 2. Convergence histories of different SSOR preconditioned iterative solvers for Example 2.

TABLE III
NUMERICAL RESULTS FOR EXAMPLE 2

Solver	*Iters*	*TRR*	CPU (s)
COCG	3342	-6.0341	80.28
QMRCOCG	2925	-6.0013	79.76
COCR	2666	-6.0160	69.25
QMRCOCR	2677	-6.0002	79.00
QMR-SYM	2921	-6.0002	137.51
CS-MINRES-QLP	$>$ **4000**	**-2.2861**	$>$ **119.81**

TABLE IV
NUMERICAL RESULTS FOR EXAMPLE 3

Solver	*Iters*	*TRR*	CPU (s)
COCG	4952	-6.0315	2768.27
QMRCOCG	4355	-6.0006	2654.23
COCR	3899	-6.0185	2260.10
QMRCOCR	3980	-6.0006	2535.20
QMR-SYM	4433	-6.0001	4174.65
CS-MINRES-QLP	†	†	†

domain. The PML absorbing material encloses the modeled domain, except on the symmetric planes. For improving the accuracy, a three-element buffer is placed between the dielectric sphere and the equivalent source surface. In addition, a three-element buffer is placed between the equivalent source surface and the PML interface. Therefore, we obtain a system of linear equations of size 51611 with a complex coefficient matrix containing 428 771 nonzero elements in the upper triangular (see [2] for details for this model issue). Here we set the maximal iteration number `maxit = 4000`. The numerical results of different preconditioned iterative solvers are given in Table III.

Convergence histories of the different preconditioned iterative solvers are illustrated in Fig. 2. We see from Fig. 2 that the relative residual 2-norm of PQMRCOCR was often less than that of PCOCG, PQMRCOCG, PQMR-SYM, and PCOCR in each iteration step. On the other hand, although the convergence curve of PCOCR is slightly smoother than that of PCOCG, but the convergence behaviors of PCOCG and PCOCR still seem considerably irregular. Moreover, both PQMRCOCG and PQMRCOCR often give much smoother convergence behaviors than PCOCG and PCOCR, respectively. Fig. 2 shows that both PQMRCOCG and PQMRCOCR yield the rather monotone convergence behaviors with compared to PCOCG and PCOCR. In Example 2, these solvers (except the precondtioned CS-MINRES-QLP) give almost the same accuracy on *TRR* at each final iteration step (see Table III).

As observed from Table III, both PCOCR and PQMRCOCR require less number of iterations and CPU time than PCOCG, PQMR-SYM, and PQMRCOCG, whereas the convergence behavior of PCOCR is still rather irregular. Moreover, the accelerating benefit of PQMRCOCR does not seem very evident via comparing with PCOCR (irregular convergence behavior). It appears from the results that QMR variants improve the COCG and COCG with respects to smooth convergence behaviors, respectively. In addition, we find that the PQMR-SYM and PQMRCOCG have almost the same convergence behaviors, but the QMR-SYM is more expensive. Based on the overall evaluation of numerical results, these QMR variants can be regarded as two robust solvers for complex symmetric linear systems in Example 2.

Example 3: At last we consider a test problem from the University of Florida Sparse Matrix Collection [28]. The test matrix "dielFilterV2clx" came from analysis of a fourth-pole dielectric resonator [31] generated with the finite-element method. For the sake of simplicity, the linear system $A\boldsymbol{x} = \boldsymbol{b}$ is defined via choosing a discrete solution $\boldsymbol{u}$ with real and imaginary parts consisting of uniformly distributed random numbers in the interval $[-1, 1]$, and the right-hand side is then computed as $\boldsymbol{b} = A\boldsymbol{u}$. The numerical results of different iterative methods for solving this problem are shown in Table IV.

As seen from Table IV, the numerical results that are required for convergence of the different preconditioned iterative solvers are shown for Example 3. It appears from the results that both PCOCR and PQMRCOCR outperform PCOCG, PQMR-SYM, and PQMRCOCG in terms of the number of iterations and CPU time, whereas the convergence behavior of PCOCR is still rather irregular. From the point of view of smooth convergence

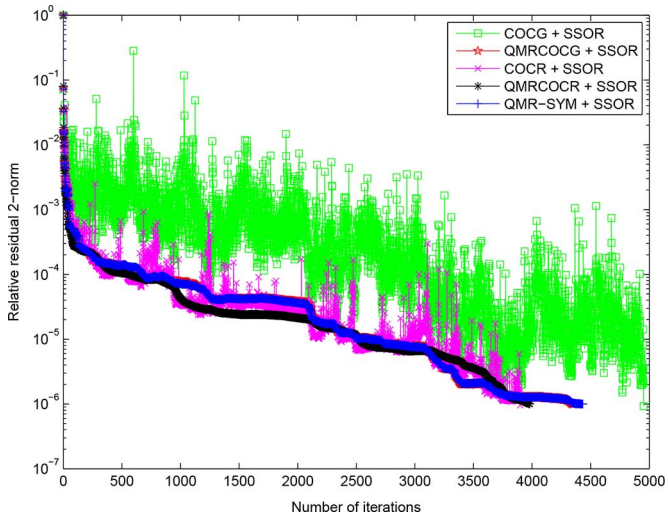


Fig. 3. Convergence histories of different iterative solvers with SSOR preconditioners for Example 3.

curves, the PQMRCOCG and PQMRCOCR are still preferred for solving the complex symmetric linear systems in Example 3. Compared with PCOCR (irregular convergence behavior), the accelerating benefit of PQMRCOCR does not seem very evident. The convergence histories of the different preconditioned iterative solvers for Example 3 are shown in Fig. 3. As seen from Fig. 3, the convergence behaviors of both PCOCG and PCOCR were considerably jagged, whereas the behaviors with preconditioned versions of QMR-SYM, QMRCOCG, and QMRCOCR were preferably smoother. In addition, from Table IV and Fig. 3, we also find that PQMRCOCG is the mathematically equivalent, but numerically improved generalized versions of the PQMR-SYM because the PQMR-SYM is based on the three-term recurrences, which is often sensitive to the rounding error.

VI. Conclusion

In this paper, we first review the COCG and QMR_SYM, which can be derived from the complex symmetric Lanczos process. We then derived the QMRCOCG via combining the COCG and QMR_SYM, and it is a mathematically equivalent, but numerically improved generalized version of QMR_SYM. Furthermore, it also can be regarded as the improved version of COCG. We derived a new algorithm of QMR_CA from the conjugate A-orthogonalization process. At the same time, we obtain the QMRCOCR by using the similar derived idea of the QMRCOCG. These QMR variants offer smoother convergence over COCG and COCR with little additional cost due to additional inner product per iterative step, respectively (see Table I). In addition, the numerical examples demonstrate that QMRCOCG and QMRCOCR often require almost the same computing time as that required by COCR and COCG, whereas the convergence behaviors of QMRCOCG and QMRCOCR are rather monotonous compared to COCG and COCR. Hence, we conclude that QMRCOCG and QMRCOCG can be viewed as two efficient solvers for complex symmetric linear systems arising in some electromagnetic simulations. Future work should include a more detailed study of convergence behaviors of QMRCOCG and QMRCOCR. Some efficient preconditioning strategies [2], [5], [11] can also be worth considering to accelerate the convergence of the novel iterative solvers.

Acknowledgment

The authors would like to thank Dr. Y.-H. Li for her helpful discussions. The authors are also grateful to the anonymous referees and editors Prof. D. Schreurs and Dr. F. Mesa for giving us so many valuable and important comments and suggestions for revising this paper's manuscript.

References

[1] K. Ise, K. Inoue, and M. Koshiba, "Three-dimensional finite-element method with edge elements for electromagnetic waveguide discontinuities," *IEEE Trans. Microw. Theory Techn.*, vol. 39, no. 8, pp. 1289–1295, Aug. 1991.

[2] L. Li, T.-Z. Huang, and Z.-G. Ren, "A preconditioned COCG method for solving complex symmetric linear systems arising from scattering problems," *J. Electromagn. Waves Appl.*, vol. 22, no. 14–15, pp. 2023–2034, 2008.

[3] H. A. van der Vorst and J. B. M. Melissen, "A Petrov-Galerkin type method for solving $A\boldsymbol{x} = \boldsymbol{b}$, where A is symmetric complex," *IEEE Trans. Magn.*, vol. 26, no. 2, pp. 706–708, Mar. 1990.

[4] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Nat. Bureau Standards*, vol. 49, no. 6, pp. 409–436, Dec. 1952.

[5] L. Li, T.-Z. Huang, Y.-F. Jing, and Y. Zhang, "Application of the incomplete Cholesky factorization preconditioned Krylov subspace method to the vector finite element method for 3-D electromagnetic scattering problems," *Comput. Phys. Commun.*, vol. 181, no. 2, pp. 271–276, Feb. 2010.

[6] R. W. Freund, "Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices," *SIAM J. Sci. Statist. Comput.*, vol. 13, no. 1, pp. 425–448, Jan. 1992.

[7] A. Bunse-Gerstner and R. Stöver, "On a conjugate gradient-type method for solving complex symmetric linear systems," *Linear Algebra Appl.*, vol. 287, no. 1–3, pp. 105–123, Jan. 1999.

[8] T. Sogabe and S.-L. Zhang, "A COCR method for solving complex symmetric linear systems," *J. Comput. Appl. Math.*, vol. 199, no. 2, pp. 297–303, Feb. 2007.

[9] T. Sogabe, "Extensions of the conjugate residual method" Ph.D. dissertation, Dept. Appl. Phys., The Univ. Tokyo, Tokyo, Japan, 2006. [Online]. Available: http://www.ist.aichi-pu.ac.jp/person/sogabe/thesis.pdf

[10] M. Clemens and T. Weiland, "Iterative methods for the solution of very large complex-symmetric linear systems of equations in electromagnetics," in *11th Copper Mountain Iterative Methods Conf.*, T. A. Manteuffel and S. F. McCormick, Eds., 1996, Pt. 2.

[11] X.-M. Gu, M. Clemens, and T.-Z. Huang, "The SCBiCG class of algorithms for complex symmetric systems with applications in several electromagnetic model problems," *Comput. Phys. Commun.*, pp. 1–31, Jul. 2014, revised.

[12] M. Clemens, T. Weiland, and U. van Rienen, "Comparison of Krylov-type methods for complex linear systems applied to high-voltage problems," *IEEE Trans. Magn.*, vol. 34, no. 5, pp. 3335–3338, Sep. 1998.

[13] R. Fletcher, "Conjugate gradient methods for indefinite systems," in *Lecture Notes in Mathematics*. Berlin, Germany: Springer, 1976, vol. 506, pp. 73–89.

[14] D. A. H. Jacobs, "A generalization of the conjugate gradient method to solve complex systems," *IMA J. Numer. Anal.*, vol. 6, no. 4, pp. 447–452, Oct. 1986.

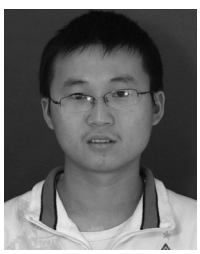
[15] Z.-Z. Bai, M. Benzi, and F. Chen, "Modified HSS iteration methods for a class of complex symmetric linear systems," *Computing*, vol. 87, no. 3–4, pp. 93–111, May 2010.

[16] O. Axelsson and A. Kucherov, "Real valued iterative methods for solving complex symmetric linear systems," *Numer. Linear Algebra Appl.*, vol. 7, no. 4, pp. 197–218, May 2000.

[17] G. Markham, "Conjugate gradient type methods for indefinite, asymmetric and complex systems," *IMA J. Numer. Anal.*, vol. 10, no. 2, pp. 155–170, Apr. 1990.

[18] P. Joly and G. Meurant, "Complex conjugate gradient methods," *Numer. Algorithms*, vol. 4, no. 3, pp. 379–406, 1993.

[19] A. Shiode, K. Abe, and S. Fujino, "Application of MRTR method for solving complex symmetric linear systems," in *Syst. Modeling and Simulation: Theory and Appl., Asian Simulation Conf.*, Hirakawa, Japan, 2007, pp. 108–112, Springer.
[20] S.-C. Choi, "Minimal residual methods for complex symmetric, skew symmetric, and skew Hermitian systems," Comput. Inst., Univ. Chicago, Chicago, IL, USA, Rep. ANL/MCS-P3028-0812, 2013, p. 26.
[21] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[22] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA, USA: SIAM, 2003.
[23] S. Minami, T. Sogabe, M. Sugihara, and S.-L. Zhang, "On an application of the QMR approach to the BiCR method," (in Japanese) *Trans. Jpn. Soc. Ind. Appl. Math.*, vol. 17, no. 3, pp. 301–317, 2007.
[24] R. W. Freund and T. Szeto, "A quasi-minimal residual squared algorithm for non-hermitian linear systems," in *Proc. 2nd Copper Mountain Iterative Methods Conf.*, Apr. 1992, UCLA-CAM Tech. Rep. 92–19.
[25] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.
[26] R. W. Freund and N. M. Nachtigal, "An implementation of the QMR method based on coupled two-term recurrences," *SIAM J. Sci. Comput.*, vol. 15, no. 2, pp. 313–337, Mar. 1994.
[27] R. W. Freund, "A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems," *SIAM J. Sci. Comput.*, vol. 14, no. 2, pp. 470–482, Mar. 1993.
[28] T. Davis, "The University of Florida sparse matrix collection," *ACM Trans. Math. Softw.*, vol. 38, no. 1, p. 25, Nov. 2011, Art. 1.
[29] C. C. Paige and M. A. Saunders, "Solution of sparse indefinite systems of linear equations," *SIAM J. Numer. Anal.*, vol. 12, no. 4, pp. 617–629, Sep. 1975.
[30] L. Li, S. Lanteri, and R. Perrussel, "A hybridizable discontinuous Galerkin method combined to a Schwarz algorithm for the solution of 3d time-harmonic Maxwell's equation," *J. Comput. Phys.*, vol. 256, pp. 563–581, Jan. 2014.
[31] F. Alessandri, M. Chiodetti, A. Giugliarelli, D. Maiarelli, G. Martirano, D. Schmitt, L. Vanni, and F. Vitulli, "The electric-field integral-equation method for the analysis and design of a class of rectangular cavity filters loaded by dielectric and metallic cylindrical pucks," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 8, pp. 1790–1797, Aug. 2004.



Xian-Ming Gu is currently working toward the Ph.D. degree at the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China.

He is currently with the School of Mathematical Sciences, UESTC. He also has taken part in the joint-supervision Ph.D. program (Sandwich Ph.D. position from August 2014 to July 2016) of the Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen (RUG), Groningen, The Netherlands. He has authored or coauthored over 12 research papers. His research interests include iterative solvers of (complex-value) linear systems and preconditioning techniques with applications in computational electromagnetics, matrix analysis with applications, fast algorithms for structured matrices, time integration methods, and numerical solutions of partial/fractional differential equations.

Ting-Zhu Huang received the B.S., M.S., and Ph.D. degrees in computational mathematics from Xi'an Jiaotong University, Xi'an, China, in 1986, 1992, and 2000, respectively.

He is currently a Professor with the School of Mathematical Sciences, University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China. He is currently an Editor for numerous publications, including *The Scientific World Journal*, *Advances in Numerical Analysis*, *Journal of Applied Mathematics*, *Journal of Pure and Applied Mathematics: Advances and Applications*, and *Journal of Electronic Science and Technology of China*. His current research interests include numerical linear algebra and scientific computation with applications, iterative methods of linear systems with application to computational electromagnetics, numerical algorithms for image processing, control theory, and matrix analysis with applications.

Liang Li received the B.S., M.Sc., and Ph.D. degrees in computational mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China, in 2004, 2007, and 2009, respectively.

During 2010, he was a Visiting Scholar with the Université Libre de Bruxelles, Bruxelles, Belgium. From 2010 to 2012, he was a Postdoctoral Fellow with INRIA Sophia Antipolis-NACHOS, Sophia Antipolis, France. He is currently an Associate Professor with the School of Mathematical Sciences, UESTC. His current research interests include iterative methods and preconditioning techniques for solving large linear systems, hybridizable discontinuous Galerkin methods, domain decomposition, and (algebraic) multigrid methods.

Hou-Biao Li received the M.Sc. and Ph.D. degrees in computational and applied mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China, in 2005 and 2007, respectively.

He currently is an Associate Professor with the School of Mathematical Sciences, UESTC. He has authored or coauthored over 30 research papers. His research interests involve numerical linear algebra, preconditioning techniques, matrix analysis with applications, numerical methods for image processing, and computational mathematics.

Tomohrio Sogabe received the B.E., M.E., and Ph.D. degrees from the University of Tokyo, Tokyo, Japan, in 2001, 2003, and 2006, respectively.

He is currently an Associate Professor with the Graduate School of Information Science and Technology, Aichi Prefectural University, Nagakute, Aichi, Japan. He is currently a Secretary Editor of *JSIAM Letters* and an Editor of the *Transactions of JSIAM*. His research interests include numerical linear algebra and scientific computing.

Markus Clemens (M'99–SM'07) was born in Wittlich, Germany, in 1968. He received the University Diploma in Mathematical Engineering ("Technomathematik") from the University of Kaiserslautern, Kaiserslautern, Germany, in 1995, the Ph.D. degree in the field of computational electromagnetics and venia legendi degree in electromagnetic theory and scientific computing from the Technische Universität Darmstadt, Darmstadt, Germany, in 1998 and 2003, respectively.

In 2004, he became Head of the Chair for Theory in Electrical Engineering and Computational Electromagnetics, Helmut-Schmidt-University, Hamburg, Germany. Since October 2009, he has been the Head of the Chair for Electromagnetic Theory, Bergische Universität Wuppertal, Wuppertal, Germany. His research activities specifically involve the development and application of numerical simulation methods for computational electromagnetics and computational multiphysics.