ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational and Applied Mathematics 199 (2007) 297–303

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

A COCR method for solving complex symmetric linear systems

Tomohiro Sogabe*, Shao-Liang Zhang

Department of Applied Physics, The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan

Received 17 December 2004

Abstract

The Conjugate Orthogonal Conjugate Gradient (COCG) method has been recognized as an attractive Lanczos-type Krylov subspace method for solving complex symmetric linear systems; however, it sometimes shows irregular convergence behavior in practical applications. In the present paper, we propose a Conjugate A-Orthogonal Conjugate Residual (COCR) method, which can be regarded as an extension of the Conjugate Residual (CR) method. Numerical examples show that COCR often gives smoother convergence behavior than COCG.

Keywords: COCG; COCR; Complex symmetric matrices; Krylov subspace methods

1. Introduction

We consider the solution of nonsingular complex symmetric linear systems of the form $A\boldsymbol{x} = \boldsymbol{b}$, where A is an $N \times N$ non-Hermitian but symmetric matrix ($A \neq \bar{A}^{\mathrm{T}}$, $A = A^{\mathrm{T}}$). Such systems arise in many important applications such as numerical computations in quantum chemistry, eddy current problems, and numerical solutions of the complex Helmholtz equation. Hence, there is a strong need for the fast solution of complex symmetric linear systems. For solving such systems efficiently, van der Vorst and Mellissen [12] proposed the conjugate orthogonal conjugate gradient (COCG) method, which is regarded as an extension of the Conjugate Gradient (CG) method [8]. Relatively complicated but robust algorithms such as QMR [6], CSYM [3], and Bi-CGCR [4] are also useful. QMR is derived from the complex symmetric Lanczos algorithm, CSYM is obtained from the idea of QMR and tridiagonalization of A by Householder reflections, and Bi-CGCR is derived from a particular case in Bi-CG [5] for solving non-Hermitian linear systems.

In the present paper, we extend the conjugate residual (CR) method described in [7,10] to complex symmetric linear systems based on an observation of deriving CG, CR, and COCG. Since CR holds a minimal residual property, the extended algorithm, named COCR, can be expected to give smoother convergence behavior than COCG in the residual norm. From a more general point of view, the algorithm of COCR is also obtained from a particular case in Bi-CR [11] which has been proposed for solving non-Hermitian linear systems, and this is similar to the relation between COCG and Bi-CG. This paper is organized as follows: in the next section, first, we observe a way to derive the algorithms of CG, CR, and COCG. Second, we derive COCR from the observation, and its orthogonality properties are discussed. In Section 3, we report some numerical examples. Finally, we make some concluding remarks in Section 4.

* Corresponding author.
E-mail address: tsogabe@zzz.t.u-tokyo.ac.jp (T. Sogabe).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.07.032

In the paper, the symbol $(\boldsymbol{x}, \boldsymbol{y})$ denotes the inner product given by $\sum_{i=0}^{N} \bar{x}_i y_i$, and the symbol $K_n(A, \boldsymbol{r}_0)$ denotes the n-dimensional Krylov subspace given by span$\{\boldsymbol{r}_0, A\boldsymbol{r}_0, \ldots, A^{n-1}\boldsymbol{r}_0\}$.

2. An extension of CR to complex symmetric linear systems

2.1. An observation of deriving CG, CR, and COCG

In this subsection, we discuss a way to obtain CG, CR, and COCG. Let $\boldsymbol{x}_n$ be the nth approximate solution in the methods. Then, the corresponding nth residual vector $\boldsymbol{r}_n (:= \boldsymbol{b} - A\boldsymbol{x}_n)$ and search direction $\boldsymbol{p}_n$ are given by the following coupled two-term recurrences:

$$\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0, \quad \boldsymbol{p}_0 = \boldsymbol{r}_0, \tag{1}$$

$$\boldsymbol{r}_n = \boldsymbol{r}_{n-1} - \alpha_{n-1} A\boldsymbol{p}_{n-1}, \tag{2}$$

$$\boldsymbol{p}_n = \boldsymbol{r}_n + \beta_{n-1}\boldsymbol{p}_{n-1}, \quad \text{for } n = 1, 2, \ldots . \tag{3}$$

The differences among the algorithms of three methods are computational formulas of α_{n-1} and β_{n-1} in the recurrences (2)–(3), and these parameters are determined by the following orthogonality conditions:

$$\boldsymbol{r}_n \perp W \quad \text{and} \quad A\boldsymbol{p}_n \perp W. \tag{4}$$

When A is Hermitian (positive definite),

- $W = K_n(A, \boldsymbol{r}_0)$ leads to CG;
- $W = AK_n(A, \boldsymbol{r}_0)$ leads to CR.

When A is complex symmetric,

- $W = K_n(\bar{A}, \bar{\boldsymbol{r}}_0)$ leads to COCG.

2.2. A derivation of COCR

In this subsection, we derive the algorithm of COCR, and discuss its orthogonality properties.

Let $\boldsymbol{r}_n$ and $\boldsymbol{p}_n$ be the nth residual vector and search direction of COCR, and also given by the recurrences (1)–(3) as well as the three methods. Then, $\boldsymbol{r}_n$ and $\boldsymbol{p}_n$ can be computed by determining α_{n-1} and β_{n-1}. Hence, as we see in (4), a choice of subspace W is needed to determine these parameters. Compared with the subspaces for CG and COCG, the main difference is the complex conjugate, and thus it is natural from the subspace for CR that we take the following choice:

- $W = \bar{A}K_n(\bar{A}, \bar{\boldsymbol{r}}_0)$.

Hence, the following orthogonality conditions are chosen for COCR:

$$\boldsymbol{r}_n \perp \bar{A}K_n(\bar{A}, \bar{\boldsymbol{r}}_0) \quad \text{and} \quad A\boldsymbol{p}_n \perp \bar{A}K_n(\bar{A}, \bar{\boldsymbol{r}}_0). \tag{5}$$

Now, we show a process for obtaining α_{n-1} and β_{n-1} using the recurrences (1)–(3) and the orthogonality conditions (5). For determining α_{n-1}, it follows from (2) that the inner product of $\bar{A}^n\bar{\boldsymbol{r}}_0$ and $\boldsymbol{r}_n$ is computed as

$$(\bar{A}^n\bar{\boldsymbol{r}}_0, \boldsymbol{r}_n) = (\bar{A}^n\bar{\boldsymbol{r}}_0, \boldsymbol{r}_{n-1}) - \alpha_{n-1}(\bar{A}^n\bar{\boldsymbol{r}}_0, A\boldsymbol{p}_{n-1}).$$

Since $\bar{A}^n\bar{\boldsymbol{r}}_0$ belongs to $\bar{A}K_n(\bar{A},\bar{\boldsymbol{r}}_0)$, it follows $(\bar{A}^n\bar{\boldsymbol{r}}_0,\boldsymbol{r}_n)=0$ from the conditions (5). Hence, we obtain

$$\alpha_{n-1}=\frac{(\bar{A}^n\bar{\boldsymbol{r}}_0,\boldsymbol{r}_{n-1})}{(\bar{A}^n\bar{\boldsymbol{r}}_0,A\boldsymbol{p}_{n-1})}. \tag{6}$$

Next, for determining β_{n-1} it follows from (3) that the inner product of $\bar{A}^n\bar{\boldsymbol{r}}_0$ and $A\boldsymbol{p}_n$ is computed as

$$(\bar{A}^n\bar{\boldsymbol{r}}_0,A\boldsymbol{p}_n)=(\bar{A}^n\bar{\boldsymbol{r}}_0,A\boldsymbol{r}_n)+\beta_{n-1}(\bar{A}^n\bar{\boldsymbol{r}}_0,A\boldsymbol{p}_{n-1}).$$

From the conditions (5) it follows $(\bar{A}^n\bar{\boldsymbol{r}}_0,A\boldsymbol{p}_n)=0$, thus we obtain

$$\beta_{n-1}=-\frac{(\bar{A}^n\bar{\boldsymbol{r}}_0,A\boldsymbol{r}_n)}{(\bar{A}^n\bar{\boldsymbol{r}}_0,A\boldsymbol{p}_{n-1})}=-\alpha_{n-1}\frac{(\bar{A}^n\bar{\boldsymbol{r}}_0,A\boldsymbol{r}_n)}{(\bar{A}^n\bar{\boldsymbol{r}}_0,\boldsymbol{r}_{n-1})}. \tag{7}$$

Here, let us consider obtaining practical formulas for α_{n-1} and β_{n-1} from (6) and (7). Note that from the recurrences (1)–(3) two vectors $\bar{A}\bar{\boldsymbol{r}}_{n-1}$ and $\bar{A}\bar{\boldsymbol{p}}_{n-1}$ can be written as

$$\bar{A}\bar{\boldsymbol{r}}_{n-1}=\bar{c}_{n-1}\bar{A}^n\bar{\boldsymbol{r}}_0+\bar{A}\bar{\boldsymbol{z}}_1,\quad \bar{A}\bar{\boldsymbol{z}}_1\in\bar{A}K_{n-1}(\bar{A},\bar{\boldsymbol{r}}_0), \tag{8}$$

$$\bar{A}\bar{\boldsymbol{p}}_{n-1}=\bar{c}_{n-1}\bar{A}^n\bar{\boldsymbol{r}}_0+\bar{A}\bar{\boldsymbol{z}}_2,\quad \bar{A}\bar{\boldsymbol{z}}_2\in\bar{A}K_{n-1}(\bar{A},\bar{\boldsymbol{r}}_0), \tag{9}$$

where $c_{n-1}=(-1)^{n-1}\prod_{i=0}^{n-2}\alpha_i$. Then, from (8), (9), and the conditions (5), the formula α_{n-1} in (6) can be rewritten by

$$\alpha_{n-1}=\frac{(\bar{A}\bar{\boldsymbol{r}}_{n-1},\boldsymbol{r}_{n-1})-(\bar{A}\bar{\boldsymbol{z}}_1,\boldsymbol{r}_{n-1})}{(\bar{A}\bar{\boldsymbol{p}}_{n-1},A\boldsymbol{p}_{n-1})-(\bar{A}\bar{\boldsymbol{z}}_2,A\boldsymbol{p}_{n-1})}=\frac{(\bar{A}\bar{\boldsymbol{r}}_{n-1},\boldsymbol{r}_{n-1})}{(\bar{A}\bar{\boldsymbol{p}}_{n-1},A\boldsymbol{p}_{n-1})}. \tag{10}$$

Similarly, the formula β_{n-1} in (7) can be rewritten by

$$\beta_{n-1}=\frac{(\bar{A}\bar{\boldsymbol{r}}_n,\boldsymbol{r}_n)}{(\bar{A}\bar{\boldsymbol{r}}_{n-1},\boldsymbol{r}_{n-1})}. \tag{11}$$

Finally, we give an update formula of the nth approximate solution $\boldsymbol{x}_n$. From the relation (2), and recall $\boldsymbol{r}_n=\boldsymbol{b}-A\boldsymbol{x}_n$, we obtain

$$\boldsymbol{x}_n=\boldsymbol{x}_{n-1}+\alpha_{n-1}\boldsymbol{p}_{n-1}. \tag{12}$$

From (1)–(3) and (10)–(12), the algorithm of COCR is obtained as follows:

Algorithm 1. COCR

$\boldsymbol{x}_0$ *is an initial guess,* $\boldsymbol{r}_0=\boldsymbol{b}-A\boldsymbol{x}_0$,
set $\boldsymbol{p}_{-1}=\boldsymbol{0}$, $\beta_{-1}=0$,
for $n=0,1,\ldots,$ *until* $\|\boldsymbol{r}_n\|\leqslant\varepsilon\|\boldsymbol{b}\|$ do:

$$\boldsymbol{p}_n=\boldsymbol{r}_n+\beta_{n-1}\boldsymbol{p}_{n-1},$$

$$(A\boldsymbol{p}_n=A\boldsymbol{r}_n+\beta_{n-1}A\boldsymbol{p}_{n-1},)$$

$$\alpha_n=\frac{(\bar{\boldsymbol{r}}_n,A\boldsymbol{r}_n)}{(\bar{A}\bar{\boldsymbol{p}}_n,A\boldsymbol{p}_n)},$$

$$\boldsymbol{x}_{n+1}=\boldsymbol{x}_n+\alpha_n\boldsymbol{p}_n,$$

$$\boldsymbol{r}_{n+1}=\boldsymbol{r}_n-\alpha_nA\boldsymbol{p}_n,$$

$$\beta_n=\frac{(\bar{\boldsymbol{r}}_{n+1},A\boldsymbol{r}_{n+1})}{(\bar{\boldsymbol{r}}_n,A\boldsymbol{r}_n)}.$$

Table 1
Summary of operations per iteration step, where AXPY: $a\boldsymbol{x}+\boldsymbol{y}$, IP: Inner product, MV: Matrix–vector product

Name	IP	AXPY	MV
QMR	2	6	1
COCG	2	3	1
COCR	2	4	1

It follows from Algorithm 1 that we obtain the three results: first, it is clear that COCR is equivalent to CR when A is real symmetric, and thus COCR can be considered as an extension of CR for symmetric linear systems to complex symmetric ones; second, similar to COCG, COCR may break down, i.e., $(\bar{A}\bar{\boldsymbol{p}}_n, A\boldsymbol{p}_n)=0$ or $(\bar{\boldsymbol{r}}_n, A\boldsymbol{r}_n)=0$ with $\boldsymbol{r}_n \neq \boldsymbol{0}$. On the other hand, it is possible for QMR to evade such breakdowns by using *look-ahead* strategy [6, Section 4]; third, if breakdown does not occur, COCR holds the following properties:

$$(\bar{\boldsymbol{r}}_i, A\boldsymbol{r}_j)=0, \quad i \neq j, \tag{13}$$

$$(\bar{A}\bar{\boldsymbol{p}}_i, A\boldsymbol{p}_j)=0, \quad i \neq j. \tag{14}$$

For the proof of (13) and (14), it is sufficient to consider the case $j<i$, and the proof is given by induction. Since the trivial case is obvious, we assume that properties (13) and (14) hold for $j<i\leqslant k$. Then, we show

$$(\bar{\boldsymbol{r}}_{k+1}, A\boldsymbol{r}_j)=0, \tag{15}$$

$$(\bar{A}\bar{\boldsymbol{p}}_{k+1}, A\boldsymbol{p}_j)=0. \tag{16}$$

First, we show (15). For the case $j<k$ it follows that $(\bar{\boldsymbol{r}}_{k+1}, A\boldsymbol{r}_j)=(\bar{\boldsymbol{r}}_k, A\boldsymbol{r}_j)-\alpha_k(\bar{A}\bar{\boldsymbol{p}}_k, A\boldsymbol{r}_j)=-\alpha_k(\bar{A}\bar{\boldsymbol{p}}_k, A\boldsymbol{r}_j)=-\alpha_k(\bar{A}\bar{\boldsymbol{p}}_k, A\boldsymbol{p}_j-\beta_{j-1}A\boldsymbol{p}_{j-1})=0$ from the assumption. For the case $j=k$ we obtain $(\bar{\boldsymbol{r}}_{k+1}, A\boldsymbol{r}_k)=(\bar{\boldsymbol{r}}_k, A\boldsymbol{r}_k)-\alpha_k(\bar{A}\bar{\boldsymbol{p}}_k, A\boldsymbol{r}_k)=(\bar{\boldsymbol{r}}_k, A\boldsymbol{r}_k)-\alpha_k(\bar{A}\bar{\boldsymbol{p}}_k, A\boldsymbol{p}_k-\beta_{k-1}A\boldsymbol{p}_{k-1})=(\bar{\boldsymbol{r}}_k, A\boldsymbol{r}_k)-\alpha_k(\bar{A}\bar{\boldsymbol{p}}_k, A\boldsymbol{p}_k)=0$ from the formula of α_k.

Next, we show (16). For the case $j<k$ it follows that $(\bar{A}\bar{\boldsymbol{p}}_{k+1}, A\boldsymbol{p}_j)=(\bar{A}\bar{\boldsymbol{r}}_{k+1}+\bar{\beta}_k\bar{A}\bar{\boldsymbol{p}}_k, A\boldsymbol{p}_j)=(\bar{A}\bar{\boldsymbol{r}}_{k+1}, A\boldsymbol{p}_j)=(1/\alpha_j)(\bar{A}\bar{\boldsymbol{r}}_{k+1}, \boldsymbol{r}_j-\boldsymbol{r}_{j+1})=0$ from the first result of the proof. For the case $j=k$ we obtain $(\bar{A}\bar{\boldsymbol{p}}_{k+1}, A\boldsymbol{p}_k)=(\bar{A}\bar{\boldsymbol{r}}_{k+1}, A\boldsymbol{p}_k)+\beta_k(\bar{A}\bar{\boldsymbol{p}}_k, A\boldsymbol{p}_k)=(1/\alpha_k)(\bar{A}\bar{\boldsymbol{r}}_{k+1}, \boldsymbol{r}_k-\boldsymbol{r}_{k+1})+\beta_k(\bar{A}\bar{\boldsymbol{p}}_k, A\boldsymbol{p}_k)=-(1/\alpha_k)(\bar{A}\bar{\boldsymbol{r}}_{k+1}, \boldsymbol{r}_{k+1})+\beta_k(\bar{A}\bar{\boldsymbol{p}}_k, A\boldsymbol{p}_k)=0$ from the formulas of α_k and β_k.

Since Algorithm 1 holds conjugate A-orthogonal property (13) and it is similar to the algorithm of CR, we named it Conjugate A-Orthogonal Conjugate Residual (COCR) method. The computational cost for QMR, COCG, and COCR at each iteration step is shown in Table 1.

At the end of this subsection, let us consider another set of formulas for α_n and β_n in Algorithm 1. From (13) and (14), it is easily verified that $\alpha_n=(\bar{A}\bar{\boldsymbol{p}}_n, \boldsymbol{r}_n)/(\bar{A}\bar{\boldsymbol{p}}_n, A\boldsymbol{p}_n)$ and $\beta_n=-(\bar{A}\bar{\boldsymbol{p}}_n, A\boldsymbol{r}_{n+1})/(\bar{A}\bar{\boldsymbol{p}}_n, A\boldsymbol{p}_n)$. This set of formulas leads to the algorithm of BiCGCR, but causes one more inner product per iteration step.

3. Numerical examples

In this section, we report some numerical examples with QMR, COCG, and COCR. We evaluate both two methods in aspects of the number of iterations (Its) and $\log_{10}$ of true relative residual 2-norm (TRR) defined as $\log_{10}\|\boldsymbol{b}-A\boldsymbol{x}_n\|/\|\boldsymbol{b}\|$. All tests were performed on an ALPHA work station with 750 MHz processor using double precision arithmetic. Codes were written in Fortran 77 and compiled with the optimization option -$O4$. In all cases the iteration was started with $\boldsymbol{x}_0=\boldsymbol{0}$, and the stopping criterion was $\|\boldsymbol{r}_n\|/\|\boldsymbol{b}\|\leqslant 10^{-6}$. The preconditioner was IC(0) [9]. For the complex symmetric structure of A, IC(0) produces LDL^{T}. If the diagonal matrix D and the lower triangular matrix L are nonsingular, then the preconditioned matrix $D^{-1/2}L^{-1}AL^{-\mathrm{T}}D^{-1/2}$ is also a nonsingular complex symmetric matrix. Thus, in this case, we can use LDL^{T} as a preconditioner. The convergence plots show $\log_{10}$ of the relative residual 2-norm, $\log_{10}\|\boldsymbol{r}_n\|/\|\boldsymbol{b}\|$, (on the vertical axis) versus Its (on the horizontal axis).

Table 2
Numerical results for Example 1, where N: order of matrix, Its: number of iterations, TRR: $\log_{10}$ of final true relative residual 2-norm

Matrix	N	Its			TRR		
		QMR	COCG	COCR	QMR	COCG	COCR
DWG961B	961	1523	1365	1388	−6.00	−6.10	−6.05
QC2534	2534	1008	628	403	−6.14	−6.06	−6.41

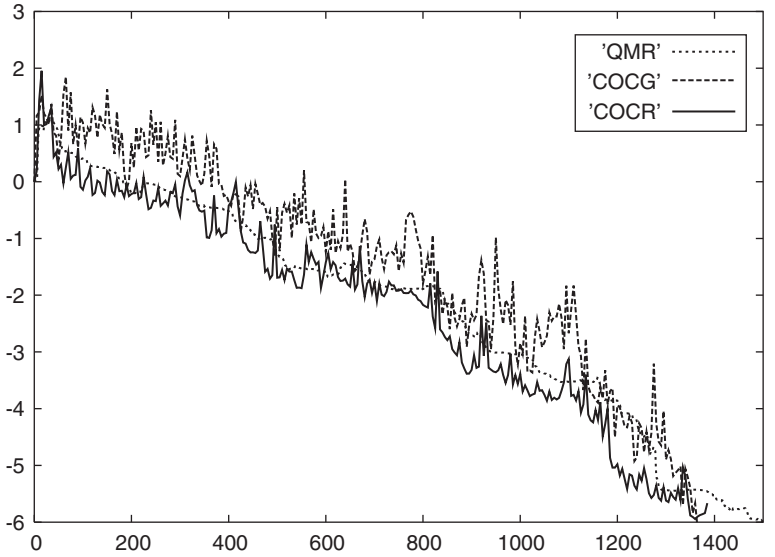

Fig. 1. The convergence history of Example 1 (DW961B).

3.1. Example 1

In the first numerical example, we consider problems from NEP collection [1]. We chose two matrices from electrical engineering (DWG961B) and quantum chemistry (QC2534). Numerical results for each test problem are given in Table 2. The right-hand side $\boldsymbol{b}$ was chosen as $(1+i,\ldots,1+i)^{\mathrm{T}}$.

Convergence histories for DWG961B and QC2534 are shown in Figs. 1 and 2. We see from Fig. 1 that COCG and COCR give similar convergence behavior, and the residual norm of COCR is often less than that of COCG and QMR at each iteration step. On the other hand, QMR often gives much smoother convergence behavior than COCG and COCR; however QMR requires a larger number of iterations than the other two methods. Fig. 2 shows that COCR converges considerably faster than COCG and QMR. In the two cases, the three methods give almost the same accuracy on TRR at each final iteration step (see Table 2).

3.2. Example 2

In the second numerical example, we consider a complex symmetric linear system arising from the 200×200 central difference discretization of the Helmholtz equation described in [2]: $u_{xx}+u_{yy}+\sigma^2 u=0$ over $[0,\pi]\times[0,\pi]$, with Dirichlet condition $u=0$ along $y=\pi$, Neumann conditions $u_x=i\sqrt{\sigma^2-\frac{1}{4}}\cos(y/2)$ along $x=0$ and $u_y=0$ along $y=0$, and radiation condition $u_x-i\sqrt{\sigma^2-\frac{1}{4}}u=0$ along $x=\pi$. This leads to a linear system with 201×200 unknowns. Here, we consider the two cases of $\sigma=2.0, 4.0$. These numerical results are shown in Figs. 3 and 4.

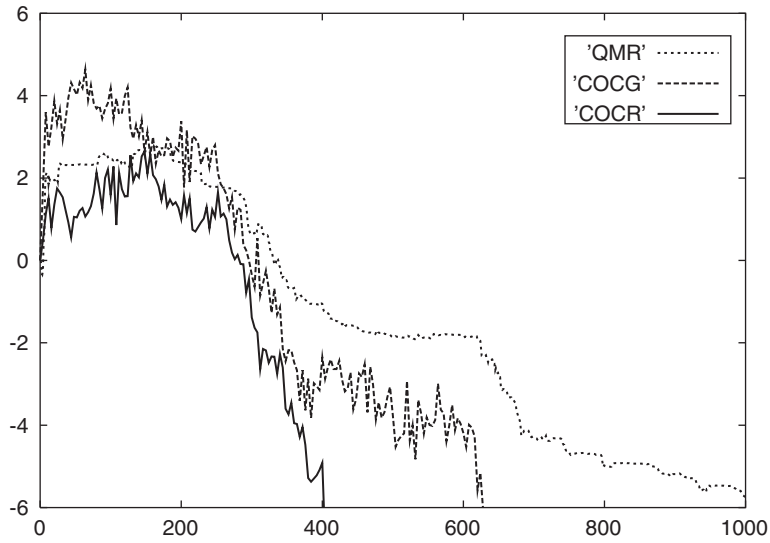

Fig. 2. The convergence history of Example 1 (QC2534).

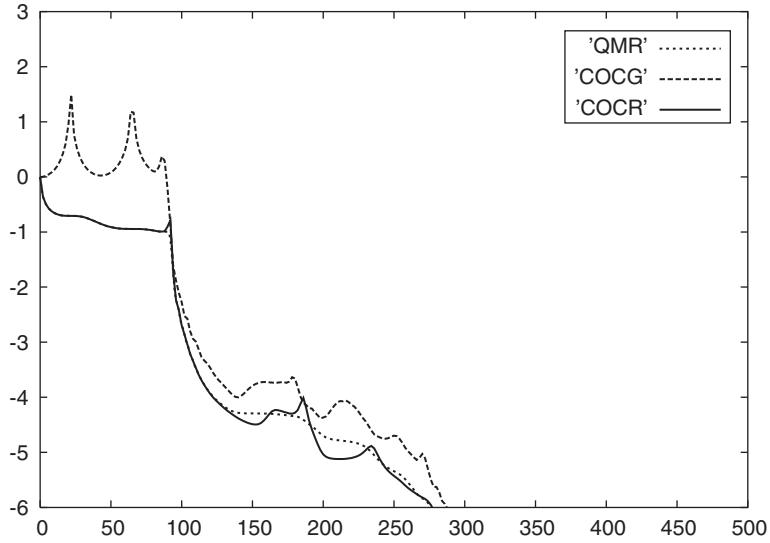

Fig. 3. The convergence history of Example 2 ($\sigma = 2.0$).

From Figs. 3 and 4 we observe that COCR gives smoother convergence behavior than COCG in the early phase, and then the two methods show similar convergence behavior. We also see that QMR has an advantage over COCG and COCR in that its residual 2-norm decreases almost monotonically. In the two cases ($\sigma = 2.0, 4.0$), they give almost the same accuracy on TRR at each final iteration step (see Table 3).

4. Concluding remarks

In this paper, first, we observed a way for obtaining CG, CR, and COCG. Second, we derived the algorithm of COCR from the observation and showed some orthogonality properties. From numerical examples we have learned that COCR tends to give smoother convergence behavior than COCG, and it sometimes converges faster than QMR in

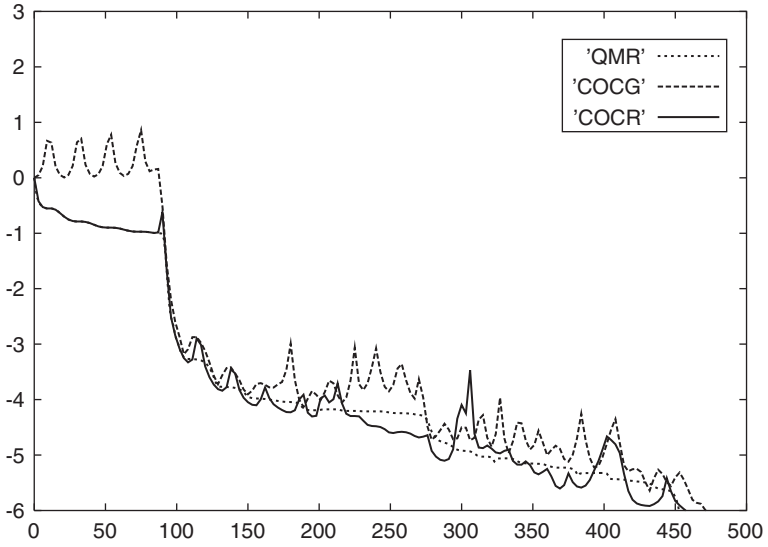

Fig. 4. The convergence history of Example 2 ($\sigma = 4.0$).

Table 3
Numerical results for Example 2, where N: order of matrix, Its: number of iterations, TRR: $\log_{10}$ of final true relative residual 2-norm

Matrix	N	Its			TRR		
		QMR	COCG	COCR	QMR	COCG	COCR
$\sigma = 2.0$	40200	276	288	278	−6.01	−6.03	−6.04
$\sigma = 4.0$	40200	453	473	458	−6.06	−6.01	−6.01

terms of the number of iterations. Hence, we conclude that COCR as well as QMR and COCG may become a useful tool for solving complex symmetric linear systems.

References

[1] Z. Bai, D. Day, J. Demmel, J. Dongarra, A test matrix collection for non-Hermitian eigenvalue problems, Technical Report CS-97-355, University of Tennessee, Knoxville, TN, 1997.
[2] A. Bayliss, C.I. Goldstein, E. Turkel, An iterative method for the Helmholtz equation, J. Comput. Phys. 49 (1983) 443–457.
[3] A. Bunse-Gerstner, R. Stöver, On a conjugate gradient-type method for solving complex symmetric linear systems, Lin. Alg. Appl. 287 (1999) 105–123.
[4] M. Clemens, T. Weiland, Comparison of Krylov-type methods for complex linear systems applied to high-voltage problems, IEEE Trans. Mag. 34 (5) (1998) 3335–3338.
[5] R. Fletcher, Conjugate gradient methods for indefinite systems, Lecture Notes in Mathematics, vol. 506, Springer, Berlin, 1976, pp. 73–89.
[6] R.W. Freund, Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices, SIAM J. Sci. Statist. Comput. 13 (1992) 425–448.
[7] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore, 1996.
[8] M.R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Standards 49 (1952) 409–436.
[9] J.A. Meijerink, H.A. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, Math. Comput. 31 (1977) 148–162.
[10] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., SIAM, Philadelphia, PA, 2003.
[11] T. Sogabe, S.-L. Zhang, Extended conjugate residual methods for solving nonsymmetric linear systems, in: Y. Yuan (Ed.), Numerical Linear Algebra and Optimization, Science Press, Beijing, 2004, pp. 88–99.
[12] H.A. van der Vorst, J.B.M. Melissen, A Petrov–Galerkin type method for solving $A\boldsymbol{x} = \boldsymbol{b}$, where A is symmetric complex, IEEE Trans. Mag. 26 (2) (1990) 706–708.