ELSEVIER Linear Algebra and its Applications 287 (1999) 105–123

LINEAR ALGEBRA AND ITS APPLICATIONS

On a conjugate gradient-type method for solving complex symmetric linear systems

Angelika Bunse-Gerstner *, Ronald Stöver

Universität Bremen, Fachbereich 3 - Mathematik und Informatik, Postfach 330 440, 28334 Bremen, Germany

Received 5 January 1998; accepted 18 May 1998

Submitted by V. Mehrmann

Dedicated to Ludwig Elsner on the occasion of his 60th birthday

Abstract

We consider large sparse linear systems $Ax = b$ with complex symmetric coefficient matrices $A = A^{\mathrm{T}}$ which arise, e.g., from the discretization of partial differential equations with complex coefficients. For the solution of such systems we present a new conjugate gradient-type iterative method, CSYM, which is based on unitary equivalence transformations of A to symmetric tridiagonal form. An analysis of CSYM shows that its convergence depends on the singular values of A and that it has both, the minimal residual property and constant costs per iteration step. We compare the algorithm with other methods for solving large sparse complex symmetric systems. © 1999 Elsevier Science Inc. All rights reserved.

1. Introduction

The problem of solving large sparse nonsingular systems of linear equations

$$Ax = b \quad \text{with } A \in \mathbb{C}^{n,n},\ \ x, b \in \mathbb{C}^n \tag{1}$$

arises for example when partial differential equations that involve complex coefficient functions or complex boundary conditions are discretized. Due to the high dimension n and the sparsity of the system iterative methods are appropriate

[*] Corresponding author. E-mail: angelika@math.uni-bremen.de.

PII: S0024-3795(98)10091-5

for such problems, but typically the coefficient matrix A is non-Hermitian, such that the classical CG algorithm cannot be applied. Nevertheless in many examples A still exhibits some structure: it is complex symmetric, i.e.,

$$A = A^{\mathrm{T}} \iff a_{ij} = a_{ji} \quad \text{for all } 1 \leqslant i, j \leqslant n\,.$$

The most common approaches for solving Eq. (1) are either to solve the normal equation $A^H Ax = A^H b$ by the preconditioned CG algorithm or to transform (1) into a real system of dimension $2n$, which can be solved by some CG-like method. While these two approaches can be used for any linear system, Freund [4] proposed a modified QMR method, a conjugate gradient-type method with quasi-minimal residuals based on the Lanczos recursion [7] that in fact exploits the special structure.

In the method presented here we create a sequence of vectors $q_1, q_2, \ldots$ by a three term recurrence relation, similar to the Lanczos method, such that for all $k = 1, 2, \ldots$ and $Q_k = [q_1, q_2, \ldots, q_k]$ we get

$$AQ_k = \overline{Q_k} T_k + w_{k+1} e_k^{\mathrm{T}}, \tag{2}$$

where T_k is a $k \times k$ complex symmetric tridiagonal matrix, $w_{k+1} \in \mathbb{C}^n$ and $\overline{Q_k}$ denotes the conjugate (element-wise) of the matrix Q_k. The column vectors of Q_k are computed as

$$q_{k+1} = \frac{\overline{w_{k+1}}}{\sqrt{w_{k+1}^H w_{k+1}}} \tag{3}$$

yielding an orthonormal sequence $q_1, q_2, \ldots$, i.e., Q_k has orthonormal columns. Here the superscript ^{H} denotes the conjugate transpose, so we work with the standard scalar product in $\mathbb{C}^n$. Since $w_{k+1}^H w_{k+1} = 0$ only if $w_{k+1} = 0$ there is in particular no danger of a serious-breakdown or a near serious-breakdown in the creation of this vector sequence. As approximations x_k to the desired solution we choose the vectors which have minimal residuals over the subspace spanned by the columns of Q_k.

The outline of the paper is as follows. In Section 2, we introduce the tridiagonalization process which computes the vectors q_j and the tridiagonal matrix T_k from (2) and we briefly discuss the procedure for determining the approximations x_k. The two methods are combined to the algorithm CSYM.

In Section 3, our algorithm is analyzed and some theoretical statements are given. We will see that it is not a Krylov subspace method, but could be viewed in a certain way as an interlacing of implicit conjugate gradient methods for $A^H A$ with starting vector $\overline{r_0}$, the conjugate of the first residual, and with starting vector $A^H r_0$.

The original complex symmetric problem can be reformulated as a real problem of twice the dimension in two different ways. At first sight CSYM may seem to be implicitly one of the Krylov subspace methods for these real problems, which both are known to have very unfavorable spectral properties [4]. In Section 4, we show why CSYM is fundamentally different from these Krylov subspace methods. Also CSYM is theoretically and numerically compared with the modified QMR method of Freund and the CGNR, i.e. the conjugate gradient method for the normal equations minimizing the residuals.

Throughout the paper, all vectors and matrices are assumed to be complex, unless stated otherwise. The coefficient matrix A is always $n \times n$, nonsingular and, unless stated otherwise, complex symmetric. $\overline{M}$ is the complex conjugate (element-wise) of M, $\operatorname{Re} M = (M + \overline{M})/2$ is the real part and $\operatorname{Im} M = (M - \overline{M})/(2\imath)$ is the imaginary part of the matrix M. By $\|x\| = \sqrt{x^H x}$ we denote the Euclidean vector norm and

$$\Pi_k := \left\{ p(\lambda) = \sum_{j=0}^{k} \gamma_j \lambda^j \mid \gamma_j \in \mathbb{C} \right\}$$

is the set of all complex polynomials of degree at most k. With e_k we denote the kth canonical unit vector of the corresponding vector space.

2. Tridiagonalizing and computing minimal residuals

In this paper we base the method CSYM on a unitary transformation $T = Q^{\mathrm{T}} A Q$ of the complex symmetric A to complex symmetric tridiagonal form T. This is similar to the Lanczos procedure which for general matrices $A \in \mathbb{C}^{n \times n}$ computes from two arbitrarily chosen unit length vectors v_1 and w_1 subsequently vectors $v_2, v_3, \ldots$ and $w_2, w_3, \ldots$ by three term recurrence relations. The vectors are such that for $V = [v_1, v_2, \ldots, v_n]$ and $W = [w_1, w_2, \ldots, w_n]$ we would achieve in general a transformation to tridiagonal form $S = W^{\mathrm{T}} A V$, where $W^{\mathrm{T}} V$ is diagonal. If A is real symmetric or complex symmetric then it suffices to compute one vector sequence $v_1, v_2, v_3, \ldots$, and work and storage requirements are reduced by one half. In this case S is real or complex symmetric, respectively. This is the basis of Freund's modified QMR method [4] for complex symmetric A, see Section 4. The crucial difference to the basis of CSYM is the fact that here we have $Q^H Q = I$, i.e., a unitary instead of a complex orthogonal transformation with the matrix V, satisfying $V^{\mathrm{T}} V = I$.

A unitary transformation $T = Q^{\mathrm{T}} A Q$ can always be achieved, which is easily seen by a construction using Householder reflections: Let

$$A = \begin{bmatrix} a_{11} & a^{\mathrm{T}} \\ a & A_{n-1} \end{bmatrix}, \qquad U_1 = \begin{bmatrix} 1 & 0 \\ 0 & I - \gamma_1 v_1 v_1^H \end{bmatrix}$$

with

$$v_1 = \frac{\widetilde{v_1}}{\|\widetilde{v_1}\|}, \qquad \widetilde{v_1} = a - \|a\| e_1, \qquad \gamma_1 = 1 + \frac{a^H v_1}{v_1^H a}.$$

If $\tilde{v}_1 = 0$ we would set $U_1 = I$. Then

$$U_1 A U_1^{\mathrm{T}} = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \beta_2 & & & & \\ 0 & & & & \\ \vdots & & & \widetilde{A_{n-1}} & \\ 0 & & & & \end{bmatrix} \qquad \text{with } \beta_2 = \|a\| > 0, \quad \alpha_1 \in \mathbb{C}$$

and repeating this process leads to

$$\underbrace{U_{n-1} \cdots U_1}_{=Q^{\mathrm{T}}} \cdot A \cdot \underbrace{U_1^{\mathrm{T}} \cdots U_{n-1}^{\mathrm{T}}}_{=:Q} = \underbrace{\begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix}}_{=:T} .$$

Note that $Q \in \mathbb{C}^{n,n}$ has orthonormal columns, the first column of Q being e_1 and that T is complex symmetric. (Without loss of generality we have assumed here that all β_k are nonzero. Otherwise we can split the problem into two smaller dimensional ones with the same structure.) Obviously the first column vector e_1 of Q can be replaced by any other unit vector with a preprocessing transformation by a suitable U_0. We can rewrite the last equation as

$$A \cdot Q = (Q^{\mathrm{T}})^H \cdot T = \overline{Q} \cdot T . \tag{4}$$

It is easily shown that if we fix the first column vector of Q and the subdiagonal elements to be real positive then the transformation is uniquely determined.

By evaluating the last equation column-wise we get recursion formulas for the columns q_k of Q and the α_k, β_k: Given q_1 it is easily seen that

$$\begin{aligned} &\alpha_1 = q_1^{\mathrm{T}} A q_1, \quad w_2 := A q_1 - \alpha_1 \overline{q_1} = \beta_2 \overline{q_2} \\ &\quad \Rightarrow \quad \beta_2 = \|w_2\|, \quad q_2 = \overline{w_2}/\beta_2 \end{aligned}$$

and for $k \geqslant 2$ with known $q_1, \ldots, q_k, \alpha_1, \ldots, \alpha_{k-1}, \beta_2, \ldots, \beta_k$:

$$\begin{aligned} &\alpha_k = q_k^{\mathrm{T}} A q_k, \quad w_{k+1} := A q_k - \alpha_k \overline{q_k} - \beta_k \overline{q_{k-1}} = \beta_{k+1} \overline{q_{k+1}} \\ &\quad \Rightarrow \quad \beta_{k+1} = \|w_{k+1}\|, \quad q_{k+1} = \overline{w_{k+1}}/\beta_{k+1} . \end{aligned}$$

This results in the following algorithm:

Tridiagonalization of the complex symmetric matrix A

```
(1) Initialize
    – Choose $0 \neq r_0 \in \mathbb{C}^n$
    – Set $w = r_0$, $q_0 = 0$
(2) For $k = 1, 2, \ldots$
    – $\beta_k = \|w\|$
    – if $\beta_k = 0$ STOP
    – $q_k = \overline{w}/\beta_k$
    – $\alpha_k = q_k^T A q_k$
    – $w = Aq_k - \alpha_k \overline{q_k} - \beta_k \overline{q_{k-1}}$
```

The following proposition, which can be proved by simple inductions, lists properties of the algorithm.

Proposition 1. *Let*

$$Q_l = [q_1, \ldots, q_l] \in \mathbb{C}^{n,l},$$

$$T_l = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_l \\ & & \beta_l & \alpha_l \end{bmatrix} \in \mathbb{C}^{l,l},$$

$$\widetilde{T}_l = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_l \\ & & \beta_l & \alpha_l \\ & & & \beta_{l+1} \end{bmatrix} \in \mathbb{C}^{l+1,l}.$$

(a) *In exact arithmetic, the algorithm terminates after $m \leqslant n$ steps, i.e., $\beta_k \neq 0$ for $k = 1, \ldots, m$ and $\beta_{m+1} = 0$.*

(b) *For $l < m$: $AQ_l = \overline{Q_l}T_l + \beta_{l+1}(\overline{q_{l+1}}e_l^T) = \overline{Q_{l+1}}\widetilde{T}_l$.*

(c) *For $j, k \leqslant l \leqslant m$: $q_j^H q_k = \delta_{jk}$, i.e. Q_l has orthonormal columns;*
for $j \leqslant l$: $\beta_j = q_j^T A q_{j-1} = q_{j-1}^T A q_j$;
for $k < j - 1, j \leqslant l$: $q_j^T A q_k = q_k^T A q_j = 0$.

We remark that this tridiagonalization can be derived as a special case of the tridiagonalization proposed by Saunders et al. [10] for general non-Hermitian matrices:

$$A \cdot Q = P \cdot T,$$

where $P, Q \in \mathbb{C}^{n,n}$ are unitary matrices and $T \in \mathbb{C}^{n,n}$ tridiagonal. By choosing $p_1 = \overline{q_1}, q_1 = \overline{r_0}/\|r_0\|$ we get $P = \overline{Q}$ and T complex symmetric, i.e., the tridiagonalization (4).

For solving $Ax = b$ we choose a starting vector $x_0 \in \mathbb{C}^n$ and construct approximations from the vector space spanned by the columns of Q_k:

$$\text{for } k = 1, 2, \ldots, : \; x_k = x_0 + Q_k z_k \; \in \; x_0 + \operatorname{span}\{q_1, \ldots, q_k\}.$$

We choose $z_k \in \mathbb{C}^k$ to minimize the norm of the corresponding residual vector $r_k = b - Ax_k$, which by using the above properties of the tridiagonalization is

$$\begin{aligned} r_k &= b - Ax_k = b - Ax_0 - AQ_k z_k = r_0 - \overline{Q_{k+1}}\widetilde{T}_k z_k \\ &= \|r_0\|\overline{q_1} - \overline{Q_{k+1}}\widetilde{T}_k z_k = \overline{Q_{k+1}}(\|r_0\|e_1 - \widetilde{T}_k z_k) \\ &\Rightarrow \|r_k\| = \left\| \|r_0\|e_1 - \widetilde{T}_k z_k \right\| \end{aligned}$$

since Q_{k+1} has orthonormal columns.

Consequently we choose z_k as the solution of the least-squares-problem

$$\|r_k\| = \min_{z \in \mathbb{C}^k} \left\| \|r_0\|e_1 - \widetilde{T}_k z \right\|,$$

which is uniquely determined because $\beta_j > 0$ for $j = 2, \ldots, k+1$ and thus $\operatorname{rank}(\widetilde{T}_k) = k$.

In order to get an efficient algorithm we solve the least-squares-problem by an updated QR-decomposition of $\widetilde{T}_k$ (following a standard approach, see e.g. [4] and the references therein): Let

$$V_k \widetilde{T}_k = \widetilde{R}_k = \begin{bmatrix} R_k \\ 0 \end{bmatrix} = \begin{bmatrix} \gamma_1 & \eta_2 & \theta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \theta_k \\ & & & \ddots & \eta_k \\ & & & & \gamma_k \\ & & & & 0 \end{bmatrix}$$

with unitary $V_k \in \mathbb{C}^{k+1,k+1}$ and $\gamma_j \neq 0$ (recall that $\operatorname{rank}(R_k) = \operatorname{rank}(\widetilde{T}_k) = k$). Then by writing

$$V_{k+1} = \begin{bmatrix} I_k & & \\ & c_{k+1} & \overline{s_{k+1}} \\ & -s_{k+1} & c_{k+1} \end{bmatrix} \begin{bmatrix} V_k & \\ & 1 \end{bmatrix} \quad \text{with } |s_{k+1}|^2 + c_{k+1}^2 = 1$$

we derive recursion formulas for $c_k, s_k, \gamma_k, \eta_k, \theta_k$ such that $V_{k+1}\tilde{T}_{k+1} = \tilde{R}_{k+1}$.

From this we get

$$\begin{aligned}
z_k &= \|r_0\| R_k^{-1}\widetilde{v}_k \quad \text{with } \widetilde{v}_k = [\zeta_1,\dots,\zeta_k]^{\mathrm{T}} \in \mathbb{C}^k \\
&\qquad\qquad\qquad\quad \text{and } [\zeta_1,\dots,\zeta_{k+1}]^{\mathrm{T}} = V_k e_1 \in \mathbb{C}^{k+1}, \\
p_k &= (q_k - \eta_k p_{k-1} - \theta_k p_{k-2})/\xi_k \quad \text{with } [p_1,\dots,p_k] := Q_k R_k^{-1}, \\
\tau_k &= \|r_0\|(-1)^{k-1} s_1 \cdots s_{k-1}, \\
x_k &= x_{k-1} + \tau_k c_k p_k
\end{aligned}$$

and thus an algorithm with constant costs per iteration step.

We remark that explicit computation of $\|r_k\|$ is not necessary since

$$\begin{aligned}
\|r_k\| &= \| \|r_0\| e_1 - \widetilde{T}_k z_k \| \\
&= \|r_0\| \|e_1 - \widetilde{T}_k R_k^{-1} \widetilde{v}_k\| \\
&= \|r_0\| \|v_k - V_k \widetilde{T}_k R_k^{-1} \widetilde{v}_k\| \\
&= \|r_0\| \left\| v_k - \begin{bmatrix} R_k \\ 0 \end{bmatrix} R_k^{-1} \widetilde{v}_k \right\| \\
&= \|r_0\| \|v_k - \begin{bmatrix} \widetilde{v}_k \\ 0 \end{bmatrix} \| \\
&= \|r_0\| \cdot |e_{k+1}^{\mathrm{T}} v_k| \\
&= \|r_0\| \cdot |s_1, \dots, s_k| \\
&= |\tau_{k+1}| \, .
\end{aligned}$$

Our iterative algorithm for approximating the solution of the system $Ax = b$ with A complex symmetric thus reads as follows:

Algorithm CSYM

```
(1) Initialize
      – Choose $x_0 \in \mathbb{C}^n$, $r_0 = b - Ax_0$,
        $tol \in \mathbb{R}_+$ (tolerance for termination),
        max$_{\text{iter}}$ (maximal number of iteration steps)
      – Set $q_0 = 0$, $q_1 = \overline{r_0}/\|r_0\|$,
        $\alpha_1 = q_1^{\mathrm{T}} A q_1$, $\beta_1 = 0$,
        $c_{-1} = 0$, $s_{-1} = 0$, $c_0 = 1$, $s_0 = 0$,
        $p_{-1} = p_0 = 0$,
        $\tau_1 = \|r_0\|$
(2) For $k = 1, 2, \dots,$ max$_{\text{iter}}$
      – $\eta_k = c_{k-2} c_{k-1} \beta_k + \overline{s_{k-1}} \alpha_k$
      – $\theta_k = \overline{s_{k-2}} \beta_k$
      – $\gamma_k = c_{k-1} \alpha_k - c_{k-2} s_{k-1} \beta_k$
```

```
- $w = Aq_k - \alpha_k\overline{q_k} - \beta_k\overline{q_{k-1}}$
- $\beta_{k+1} = \|w\|$
- if $\beta_{k+1} = 0$ STOP
- $q_{k+1} = \overline{w}/\beta_{k+1}$
- $\alpha_{k+1} = q_{k+1}^{\mathrm{T}}Aq_{k+1}$
- if $\gamma_k \neq 0$
  * $c_k = \frac{|\gamma_k|}{\sqrt{|\gamma_k|^2+\beta_{k+1}^2}}$
  * $s_k = \frac{\overline{\gamma_k}}{|\gamma_k|}\frac{\beta_{k+1}}{\sqrt{|\gamma_k|^2+\beta_{k+1}^2}}$
  * $\xi_k = \frac{\gamma_k}{|\gamma_k|}\sqrt{|\gamma_k|^2 + \beta_{k+1}^2}$
  else
  * $c_k = 0$
  * $s_k = 1$
  * $\xi_k = \beta_{k+1}$
- $p_k = (q_k - \eta_k p_{k-1} - \theta_k p_{k-2})/\xi_k$
- $x_k = x_{k-1} + \tau_k c_k p_k$
- $\tau_{k+1} = -s_k\tau_k$
- $\|r_k\| = |\tau_{k+1}|$
- if $\|r_k\| < tol$ STOP
```

3. An analysis of CSYM

In this section we analyze the given algorithm CSYM and show some theoretical results.

There are two ways in which the algorithm can stop. Either $\beta_{k+1} = 0$ and this is equivalent to $A^{-1}b \in x_0 + \mathrm{span}\{q_1, \dots, q_k\}$, i.e., we found the solution in a k-dimensional subspace of $\mathbb{C}^n$, or $\|r_k\| <$ tol (the user-defined tolerance), i.e., x_k is a sufficiently exact approximation of $A^{-1}b$, where the quality depends on tol and the condition number of A.

Proposition 2 gives a representation of the subspace $K_k := \mathrm{span}\{q_1, \dots, q_k\}$.

Proposition 2

$$K_k = \begin{cases} \mathrm{span}\left\{\overline{r_0}, \overline{A}r_0, \overline{A}A\overline{r_0}, \dots, (\overline{A}A)^{(k-2)/2}\,\overline{A}r_0\right\} & \textit{for } k \textit{ even} \\ \mathrm{span}\left\{\overline{r_0}, \overline{A}r_0, \overline{A}A\overline{r_0}, \dots, (\overline{A}A)^{(k-1)/2}\,\overline{r_0}\right\} & \textit{for } k \textit{ odd} \end{cases}$$

$$= \left\{p_1(\overline{A}A)\overline{r_0} + p_2(\overline{A}A)\overline{A}r_0 \mid p_1 \in \Pi_{\lfloor (k-1)/2 \rfloor}, p_2 \in \Pi_{\lfloor (k-2)/2 \rfloor}\right\}.$$

With $\lfloor x \rfloor$ we denote the largest integer less or equal to x. This space is not a Krylov subspace of the usual kind $K_m(B, v) = \{p(B)v \mid p \in \Pi_{m-1}\}$ because **two** polynomials according to the two vectors $\overline{r_0}$ and $\overline{A}r_0$ are involved. We avoided the use of the indefinite bilinear form $(x, y) = y^{\mathrm{T}}x$ for the price of having complex conjugates involved, thus resulting in a more complicated subspace.

By using the standard inner product not only breakdowns resulting form the occurrence of isotropic vectors are avoided, but we get a tridiagonalization with an orthonormal matrix Q from which we derive a minimal residual property of CSYM.

Proposition 3

$$\|r_k\| = \min_{z\in\mathbb{C}^k}\left\|\|r_0\|e_1 - \widetilde{T}_k z\right\| = \min_{z\in\mathbb{C}^k}\left\|\overline{Q_{k+1}}(\|r_0\|e_1 - \widetilde{T}_k z)\right\|$$

$$= \min_{z\in\mathbb{C}^k}\|r_0 - AQ_k z\| = \min_{x\in x_0+K_k}\|b - Ax\|.$$

Note that CSYM is an algorithm with constant costs per iteration step and the minimal residual property. Faber and Manteuffel proved (see [3]) that CG-like Krylov subspace methods satisfying these two properties exist only for the very special class of matrices:

$$A = \exp(\imath\theta)(T + \sigma I_n), \quad T = T^H, \;\; \theta \in \mathbb{R}, \;\; \sigma \in \mathbb{C}.$$

For the class of shifted unitary matrices an optimal Krylov subspace method using coupled recursions was described in [6]. This is another way to generalize the CG method.

While in the analysis of CG-like algorithms the spectrum of A plays an important role, in our case the singular values take over this part. The special structure of complex symmetry leads to a special singular value decomposition of A, called Takagi SVD (see e.g. [5], Corollary 4.4.4, pp. 204/205, see also [1]): There exists $V \in \mathbb{C}^{n,n}$ unitary such that

$$A = V\Sigma V^{\mathrm{T}},$$

where $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n), \sigma_j \geqslant 0$. By inserting this into the representation

$$r_k = r_0 - p_1(A\overline{A})A\overline{r_0} - p_2(A\overline{A})A\overline{A}r_0$$

for some polynomials $p_1 \in \Pi_{\lfloor(k-1)/2\rfloor}, p_2 \in \Pi_{\lfloor(k-2)/2\rfloor}$ we get

$$r_k = V\left(\left(I - \Sigma^2 p_2(\Sigma^2)\right)V^H r_0 - \Sigma p_1(\Sigma^2)V^{\mathrm{T}}\overline{r_0}\right).$$

Let $y := V^H r_0$, then $V^{\mathrm{T}}\overline{r_0} = \bar{y}$. Writing the condition $r_k = 0$ row-wise we get

$$\sigma_l\left(\sum_{j=0}^{\lfloor(k-1)/2\rfloor} a_j\sigma_l^{2j}\right)\overline{y_l} + \sigma_l^2\left(\sum_{j=0}^{\lfloor(k-2)/2\rfloor} b_j\sigma_l^{2j}\right)y_l = y_l \quad (l = 1, \ldots, n),$$

where a_j, b_j denote the coefficients of p_1 and p_2, respectively.

This leads to the equivalent linear system (shown for k even)

$$\begin{bmatrix} \overline{y_1} & \sigma_1 y_1 & \sigma_1^2\overline{y_1} & \cdots & \sigma_1^{k-2}\overline{y_1} & \sigma_1^{k-1}y_1 \\ \vdots & & & & & \vdots \\ \overline{y_n} & \sigma_n y_n & \sigma_n^2\overline{y_n} & \cdots & \sigma_n^{k-2}\overline{y_n} & \sigma_n^{k-1}y_n \end{bmatrix} \begin{bmatrix} a_0 \\ b_0 \\ \vdots \\ a_{k/2-1} \\ b_{k/2-1} \end{bmatrix} = \begin{bmatrix} \frac{y_1}{\sigma_1} \\ \vdots \\ \frac{y_n}{\sigma_n} \end{bmatrix}$$

Let's look at the simplest case $\Sigma = \sigma I_n$. Then, in general, CSYM will not terminate in one step as one may expect, because the system $\bar{y}a_0 = 1/\sigma y$ is solvable if and only if the vector $y = V^H r_0$ is either real or pure imaginary. But of course CSYM stops after two steps.

By generalizing these considerations we get a statement about convergence of CSYM in exact arithmetic. We are looking for the minimal number k for which polynomials $p_1 \in \Pi_{\lfloor (k-1)/2 \rfloor}, p_2 \in \Pi_{\lfloor (k-2)/2 \rfloor}$ exist such that

$$(I - \Sigma^2 p_2(\Sigma^2))y - (\Sigma p_1(\Sigma^2))\bar{y} = 0.$$

Examining this equation component-wise, the following becomes obvious:

- If $y_l = e_l^T y = 0$ we have no conditions for p_1, p_2.
- In the case of a simple singular value σ_l we get a condition either for p_1 or for p_2.
- For a multiple singular value $\sigma_l = \cdots = \sigma_{l+m}$ a necessary condition

$$(1 - \sigma_l^2 p_2(\sigma_l^2))z - (\sigma_l p_1(\sigma_l^2))\bar{z} = 0, \; z := \begin{bmatrix} y_l \\ \vdots \\ y_{l+m} \end{bmatrix} \in \mathbb{C}^{m+1}$$

occurs. We have two cases: If $z, \bar{z}$ are linearly independent, we get a condition for p_1 and p_2, otherwise a condition for either p_1 or p_2.

Proposition 4. *In exact arithmetic, CSYM converges in at most $2M + N$ steps, where M is the number of multiple singular values and N the number of simple singular values of A.*

Note that additional conditions for r_0 may decrease the number of iterations.

Finally we get an upper bound for the convergence rate which we derive from the well-known estimation

$$\|x^* - x_k\|_B \leqslant 2\left(\frac{\sqrt{\kappa(B)} - 1}{\sqrt{\kappa(B)} + 1}\right)^k \cdot \|x^* - x_0\|_B$$

for the CG algorithm applied to a Hermitian matrix B (where x^* is the exact solution and $\|y\|_B = \sqrt{y^H B y}$).

Proposition 5

$$\|r_k^{\mathrm{CSYM}}\| \leqslant 2\left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^{\lfloor k/2 \rfloor} \cdot \|r_0\|$$

where $\kappa(A)$ is the condition number corresponding to $\|\cdot\| = \|\cdot\|_2$.

Proof. Applying the CG method to the normal equation we get

$$\begin{aligned}\|r_k^{\mathrm{CGNR}}\| &= \|A(x^* - x_k^{\mathrm{CGNR}})\| = \|x^* - x_k^{\mathrm{CGNR}}\|_{A^HA} \\ &\leqslant 2\left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^k \cdot \|x^* - x_0\|_{A^HA} = 2\left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^k \|r_0\|\end{aligned}$$

because $\kappa(A^HA) = \kappa(A)^2$. Since

$$\begin{aligned}\|r_{2k+1}^{\mathrm{CSYM}}\| \leqslant \|r_{2k}^{\mathrm{CSYM}}\| &= \min_{p_1, p_2 \in \Pi_{k-1}} \|r_0 - A\bar{A}p_2(A\bar{A})r_0 - p_1(A\bar{A})A\bar{r_0}\| \\ &\leqslant \min_{p \in \Pi_{k-1}} \|r_0 - A\bar{A}p(A\bar{A})r_0\| = \|r_k^{\mathrm{CGNR}}\|\end{aligned}$$

the inequality holds true. □

This bound is quite unfavorable because $\kappa(A)$ instead of $\sqrt{\kappa(A)}$ and the power $\lfloor k/2 \rfloor$ instead of k occurs. But equality is in general not achieved since the polynomial p_1 was set to 0 for the derivation of this bound. Looking at the numerical examples in Section 4.2 one can see that in all these cases $\|r_k^{\mathrm{CSYM}}\| < \|r_{\lfloor k/2 \rfloor}^{\mathrm{CGNR}}\|$ holds true. It can be shown, however, that the bound is sharp in the following sense: For any complex symmetric matrix and any k there exists an initial residual such that in the kth step the norm of the residual is equal to $\|r_{\lfloor k/2 \rfloor}^{\mathrm{CGNR}}\|$.

4. Comparison to other approaches

There are (at least) three other appropriate methods for solving large sparse complex symmetric systems. We compare them with CSYM by some theoretical investigations. Numerical examples are displayed to illustrate the results and to show the typical numerical performance.

4.1. Theoretical investigations

Any linear system can be solved by the CG algorithm applied to the normal equations $A^HAx = A^Hb$, where in every step the residual is minimized (CGNR method). For complex symmetric systems this results in

$$r_k^{\mathrm{CGNR}} = (I - A\bar{A}p(A\bar{A}))r_0 = V(I - \Sigma^2 p(\Sigma^2))y$$

for a polynomial $p \in \Pi_{k-1}$ (where $A = V\Sigma V^{\mathrm{T}}$ is the Takagi SVD and $y = V^H r_0$). We see that convergence depends on the squares of the singular values of A, corresponding to the spectrum $\sigma(\overline{A}A) = \{\sigma^2 \mid \sigma \text{singular value of} A\}$. Thus we see that

- CGNR does not use the complex symmetric structure.
- The convergence depends on the squares of the singular values of A, often causing a rather slow convergence.
- In exact arithmetic CGNR converges in at most $M + N$ steps (M the number of multiple and N the number of simple singular values, see Proposition 4). Thus fewer steps than for CSYM are needed, because each multiple singular value requires only one step of CGNR (but two steps of CSYM).
- Every iteration step is twice as expensive in terms of computational work as in CSYM or QMR due to two matrix–vector products with A and $\overline{A}$, respectively.

An obvious way to solve complex linear systems is to rewrite them as real linear systems of twice the dimension and to apply some CG-like methods to the resulting systems. There are two possibilities:

$$Ax = b \iff \underbrace{\begin{bmatrix} \operatorname{Re} A & -\operatorname{Im} A \\ \operatorname{Im} A & \operatorname{Re} A \end{bmatrix}}_{=:A_*} \begin{bmatrix} \operatorname{Re} x \\ \operatorname{Im} x \end{bmatrix} = \begin{bmatrix} \operatorname{Re} b \\ \operatorname{Im} b \end{bmatrix}$$

$$\iff \underbrace{\begin{bmatrix} \operatorname{Re} A & \operatorname{Im} A \\ \operatorname{Im} A & -\operatorname{Re} A \end{bmatrix}}_{=:A_{**}} \begin{bmatrix} \operatorname{Re} x \\ -\operatorname{Im} x \end{bmatrix} = \begin{bmatrix} \operatorname{Re} b \\ \operatorname{Im} b \end{bmatrix}.$$

Freund [4] examined the correspondence between Krylov subspace methods for the complex system and the two real systems and pointed out that the real systems should not be used because A_*, A_{**} have quite unfavorable spectral properties. The kth iterate of a Krylov subspace method for A_{**} can be written [4] as

$$x_k = x_0 + R(\overline{A}A)\overline{r_0} + S(\overline{A}A)\overline{A}r_0,$$

where R and S are real polynomials of degree $\lfloor (k-1)/2 \rfloor$ and $\lfloor (k-2)/2 \rfloor$, respectively. In view of Proposition 2 it seems now at first sight that CSYM is just a complex version of a Krylov subspace method for A_{**}. Here we show that CSYM is not equivalent to either of these real methods.

Proposition 6. *The transformation $\phi: \mathbb{C}^n \to \mathbb{R}^{2n}$, $\phi(x) := [\operatorname{Re} x, \operatorname{Im} x]^{\mathrm{T}}$ has the following properties*:

- *ϕ is a one-to-one-correspondence, $\phi^{-1}(y_1, y_2) = y_1 + \imath y_2$.*
- *ϕ is isometric: $\|\phi(x)\| = \|x\|$.*
- *$\phi(\overline{x}) = P\phi(x)$ with*

$$P = \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix}$$

- $\phi(x+y) = \phi(x) + \phi(y)$.
- $\phi(\alpha x) = \begin{bmatrix} \mathrm{Re}\,\alpha I_n & -\mathrm{Im}\,\alpha I_n \\ \mathrm{Im}\,\alpha I_n & \mathrm{Re}\,\alpha I_n \end{bmatrix} \phi(x)$ *for* $\alpha \in \mathbb{C}$.
- $\phi(Ax) = A_*\phi(x) = A_{**}P\phi(x)$.

With ϕ we can "translate" CSYM into methods for the two real linear systems. It is sufficient to look at the corresponding residuals and to note that $A_*P = A_{**}$. It is not difficult to prove the following.

Proposition 7. *If*

$$r_k = r_0 - A\overline{A}\left(\sum_{j=0}^{\lfloor (k-2)/2 \rfloor} b_j (A\overline{A})^j\right) r_0 - \left(\sum_{j=0}^{\lfloor (k-1)/2 \rfloor} c_j (A\overline{A})^j\right) A\overline{r_0}$$

then

$$\begin{aligned}\phi(r_k) &= \left(I - \sum_{j=0}^{k-1} D_j (A_*P)^{j+1}\right) \phi(r_0) \\ &= \left(I - \sum_{j=0}^{k-1} D_j A_{**}^{j+1}\right) \phi(r_0),\end{aligned}$$

where

$$D_j := \begin{bmatrix} \mathrm{Re}\, d_j I_n & -\mathrm{Im}\, d_j I_n \\ \mathrm{Im}\, d_j I_n & \mathrm{Re}\, d_j I_n \end{bmatrix}, \qquad d_j = \begin{cases} c_{\frac{j}{2}} & \textit{for } j \textit{ even} \\ b_{(j-1)/2} & \textit{for } j \textit{ odd}. \end{cases}$$

We see that CSYM is fundamentally different because the polynomials p_1, p_2 in CSYM can have complex coefficients which leads to the strange polynomial with coefficients $D_j \in \mathbb{R}^{2n,2n}$ in the representation of $\phi(r_k)$. Due to the doubled dimension and the unfavorable eigenstructures of A_*, A_{**}:

$$\sigma(A_*) = \sigma(A) \cup \overline{\sigma(A)}\ , \ \sigma(A_{**}) = \{\pm\sigma \mid \sigma \text{ singular value of} A\}$$

(see [4]), CSYM is in general much more efficient than any Krylov subspace method for A_* or A_{**}.

By the transformation ϕ it is possible to translate CSYM into a real version applicable to matrices of the form

$$\begin{bmatrix} B & C \\ C & -B \end{bmatrix} \text{ or } \begin{bmatrix} B & -C \\ C & B \end{bmatrix}$$

with $B, C \in \mathbb{R}^{n,n}$ symmetric but we recommend to go the other direction: apply CSYM on the complex symmetric matrix $A = B + \imath C$.

The modified quasi-minimal residual method proposed by Freund [4] is based on the Lanczos recursion and creates a sequence of vectors $p_1, p_2, \dots$ by a three term recurrence relation such that for all $k = 1, 2, \dots$ and $P_k = [p_1, p_2, \dots, p_k]$ we get

$$AP_k = P_k S_k + \tilde{p}_{k+1} e_k^{\mathrm{T}},$$

where S_k is a $k \times k$ complex symmetric tridiagonal matrix, e_k is the kth unit vector of dimension k, $\tilde{p}_{k+1} \in \mathbb{C}^n$,

$$p_k^{\mathrm{T}} p_j = 0 \quad \text{for } j \neq k \quad \text{and} \quad p_k^{\mathrm{T}} p_k = 1 \quad \text{for all } k, j.$$

Note that for a vector $p \in \mathbb{C}^n$ the indefinite inner product $p^{\mathrm{T}} p$ may be zero even if $p \neq 0$. Thus P_k does not have orthonormal columns, with respect to the standard scalar product. Moreover, in the three term recursion one has to divide by $\tilde{p}_{k+1}^{\mathrm{T}} \tilde{p}_{k+1}$. If this term is zero the recursion process breaks down. In [4] Freund suggests in the near serious-breakdown case, i.e., in situations in which $\tilde{p}_{k+1}^{\mathrm{T}} \tilde{p}_{k+1}$ is very small, while $\|\tilde{p}_{k+1}\|$ is not, to use look-ahead Lanczos steps.

The approximations $x_k = x_0 + P_k z_k$ are constructed such that z_k is the solution of the minimization problem

$$\min_{z \in \mathbb{C}^k} \left\| \omega_1 \|r_0\| e_1 - \Omega_{k+1} \widetilde{S}_k z \right\|$$

with

$$\widetilde{S}_k = \begin{bmatrix} & S_k & \\ 0 \quad \cdots \quad 0 & & \sqrt{\tilde{p}_{k+1}^{\mathrm{T}} \tilde{p}_{k+1}} \end{bmatrix} \in \mathbb{C}^{k+1,k}.$$

The weight-matrix $\Omega_{k+1} = \operatorname{diag}(\omega_1, \dots, \omega_{k+1})$ is usually chosen by $\omega_j = \|p_j\|$. Since $\operatorname{span}\{p_1, \dots, p_k\} = K_k(A, r_0)$ QMR is a Krylov subspace method with residuals

$$r_k^{\mathrm{QMR}} = r_0 - Ap(A)r_0, \quad p \in \Pi_{k-1}$$

but $\|r_k^{\mathrm{QMR}}\|$ is not minimal over the Krylov subspace. In comparison to CSYM we can say:

- QMR exploits the special structure of A in the same way as CSYM, since due to the complex symmetry only one vector sequence in the Lanczos process is required.
- Because the indefinite bilinear form $(x, y) = y^{\mathrm{T}} x$ is used breakdowns are possible when a division by $\tilde{v}^{\mathrm{T}} \tilde{v}$ for an isotropic vector $\tilde{v}$ occurs.
- QMR is a Krylov subspace method, so the convergence depends on the Jordan structure of A. But every complex-valued matrix is similar to a complex symmetric one, i.e. spectrum and Jordan structure are not special at all.

- The number of iteration steps cannot be compared with that of CSYM in general, because convergence of the first method depends on the eigenvalues and convergence of the other on the singular values.
- The work per iteration step is nearly identical. One matrix-vector multiplication and the solution of a least-squares-problem by an updated QR-factorization is done in both algorithms.

4.2. Numerical examples

Since the disadvantages of any CG-like method applied to A_* or A_{**} are obvious after the theoretical investigations, we restricted the numerical examples to the three methods CSYM, QMR, CGNR and chose five examples which shall give a good impression about performance of these methods.

With each example we present a computed convergence curve of the number of matrix–vector products versus the relative residual norms $\|r_k\|/\|r_0\|$ in a logarithmic scale (instead of the matrix–vector products one can look at the number of iteration steps: for CSYM and QMR this is the same, CGNR needs two multiplications per step). The initial vector x_0 and the right-hand side b are chosen, unless stated otherwise, as random complex vectors with independent normally distributed elements of zero mean and variance 1, the tolerance for termination is always $10^{-6} \cdot \|r_0\|$ and the maximum number of iteration steps is chosen appropriately.

All computations were done on a SPARC ULTRA 1 Sun workstation with MATLAB 5.1 [8].

Example 1. We start with an example given by Freund [4] which results from the discretization of the Helmholtz equation. The complex symmetric matrix

$$A = A_0 - 100h^2 I_n + \imath h^2 \mathrm{diag}(d_1, \ldots, d_n)$$

is of dimension $n = 961$ with $h = 1/(\sqrt{n} + 1)$, $d_j \in [0, 10]$ randomly chosen and A_0 arising from the usual five-point discretization of the Laplace operator. Here $x_0 = 0$ and b has components all equal to $1 + \imath$. We see in Fig. 1 that QMR converges very fast while CSYM and CGNR need more than 1050 and 2100 matrix–vector products, respectively. Note that there are imaginary numbers only on the main diagonal such that the matrix is nearly real symmetric.

Example 2. Another example arising from the Helmholtz equation is the matrix Young1c from the Harwell–Boeing sparse matrix collection [2] which is of dimension $n = 841$ (see Fig. 2). Again QMR is the fastest of the methods but not as fast as in Example 1.

Example 3. In order to give an example with only one eigenvalue but extremely bad singular value distribution [9] we chose

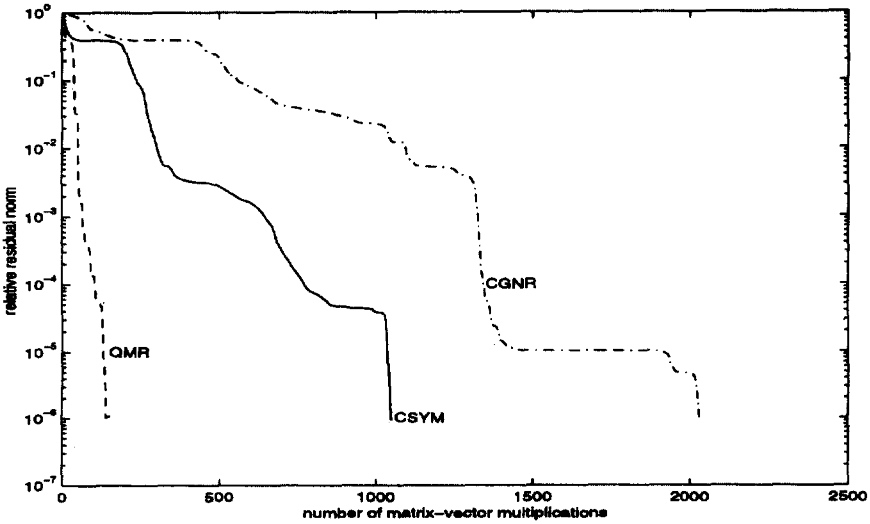

Fig. 1. Example 1.

$$A_k = \begin{bmatrix} 1 & k-1 \\ 0 & 1 \end{bmatrix} \quad \text{for } k = 1,\ldots,\frac{n}{2}, \quad S = \begin{bmatrix} 1 & 0 \\ 1 & i \end{bmatrix},$$

$$A = \operatorname{diag}(S^{-1}A_1S,\ldots,S^{-1}A_{n/2}S),$$

where A is a block-diagonal complex symmetric matrix with $\sigma(A) = \{1\}$ and singular values

$$\sigma = \sqrt{2(k-1)^2 + 1 \pm 2(k-1)\sqrt{(k-1)^2+1}}.$$

The dimension is $n = 100$, for larger n the extremely bad singular value distribution leads to a very slow convergence of CSYM and CGNR. Fig. 3 shows that, as expected, QMR converges in two steps, however CSYM and CGNR need more than 800 respectively 1800 matrix–vector multiplications. We here clearly see the advantages of CSYM versus CGNR.

Example 4. In order to demonstrate the dependence of CSYM-convergence on the multiple singular values we constructed a complex symmetric matrix of dimension $n = 100$ with double singular values. Since in exact arithmetic CSYM needs two iteration steps per singular value and CGNR only one but two matrix-vector products per iteration step the number of matrix–vector products is nearly the same for CSYM and CGNR (see Fig. 4).

Example 5. The last example is a randomly chosen complex symmetric sparse matrix of dimension $n = 1000$ with five nonzero sub- and super-diagonals (i.e.,

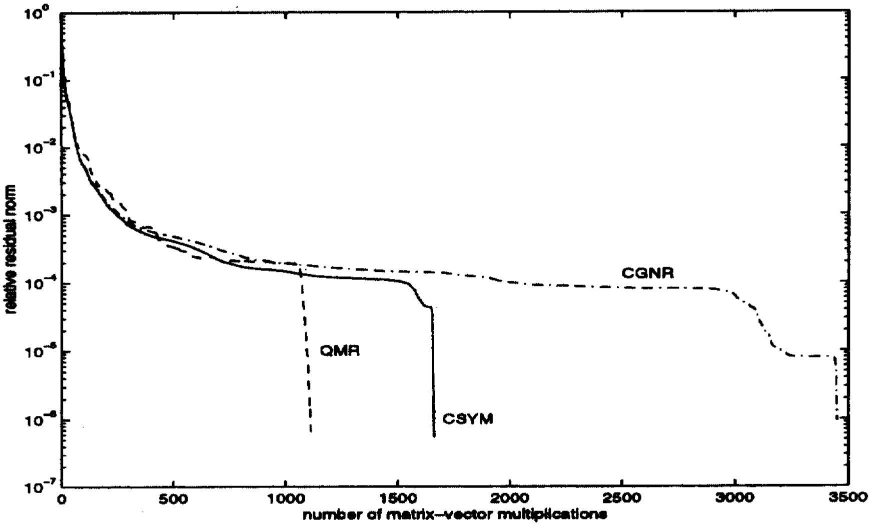

Fig. 2. Example 2.

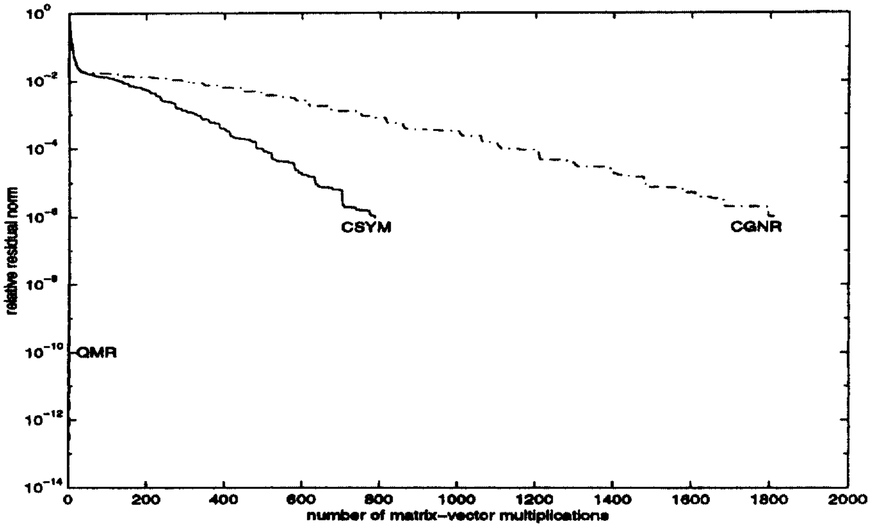

Fig. 3. Example 3.

bandwidth 11). The curves are typical for all examined random examples of large dimension ($500 \leqslant n \leqslant 1000$) and different bandwidth: QMR does not converge within 3000 iteration steps while CSYM and CGNR can handle these matrices (see Fig. 5).

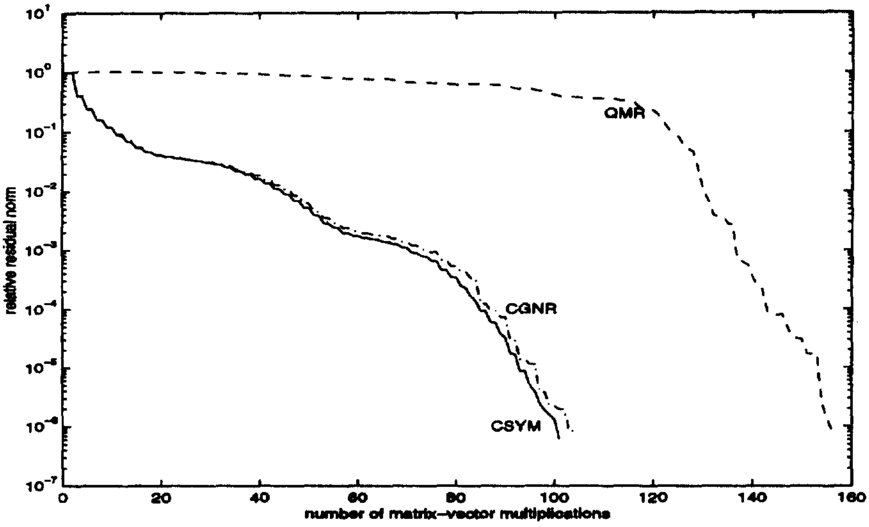

Fig. 4. Example 4.

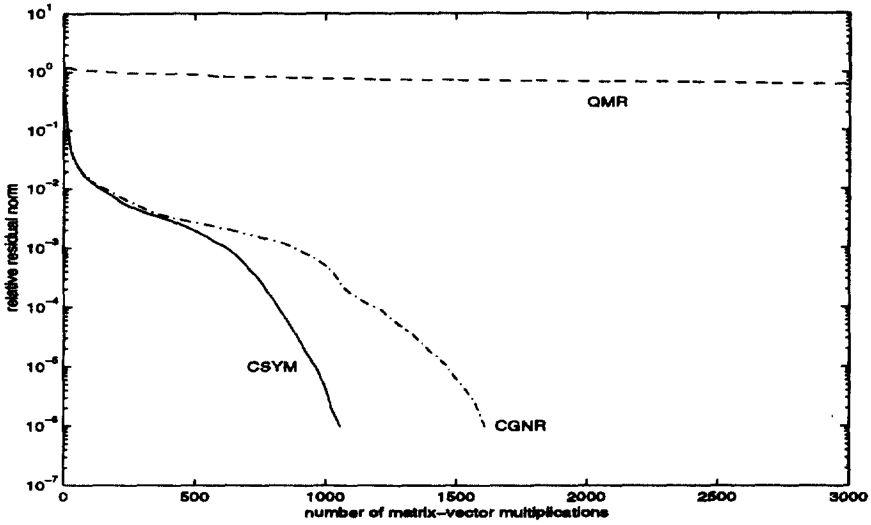

Fig. 5. Example 5.

To summarize the numerical examples we can say:

- There exist extreme examples for which either QMR or CSYM and CGNR converge in one and in two steps, respectively, due to the eigenvalue or singular value structure.

- For general complex symmetric matrices CSYM and CGNR beat QMR by far, while for matrices belonging to the Helmholtz equation QMR beats the other two.
- CSYM performs always better than CGNR, in particular considering the number of matrix–vector products.

Note that in comparing the three methods for our examples we used no preconditioning. In practice CG-like methods have to be used with preconditioning, which usually speeds up convergence dramatically. It remains an open question what the most efficient preconditioner for a complex symmetric system would be. First empirical studies led to unexpected results and more investigations are necessary.

References

[1] A. Bunse-Gerstner, W. Gragg, Singular value decompositions of complex symmetric matrices, J. Comp. Appl. Math. 21 (1988) 41–54.

[2] I.S. Duff, R.G. Grimes, J.G. Lewis, Users' guide for the Harwell–Boeing sparse matrix collection (release I), Technical Report TR/PA/92/86, Cerfacs, Toulouse Cedex, France, October 1992.

[3] V. Faber, T. Manteuffel, Necessary and sufficient conditions for the existence of a conjugate–gradient method, SIAM J. Numer. Anal. 21 (1984) 352–362.

[4] R. Freund, Conjugate–gradient-type methods for linear systems with complex symmetric coefficient matrices, SIAM J. Sci. Stat. Comput. 13 (1992) 425–448.

[5] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, MA, 1985.

[6] C. Jagels, L. Reichel, A fast minimal residual algorithm for shifted unitary matrices, Numer. Linear Algebra Appl. 1 (1994) 555–570.

[7] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Nat. Res. Bur. Standards 45 (1950) 255–282.

[8] Using MATLAB, The Math Works, 24 Prime Park Way, Natick, MA 01760–1500, USA, 1997.

[9] N. Nachtigal, S. Reddy, L. Trefethen, How fast are nonsymmetric matrix iterations?, SIAM J. Matrix Anal. Appl. 13 (1992) 778–795.

[10] M.A. Saunders, H.D. Simon, E.L. Yip, Two conjugate–gradient-type methods for unsymmetric linear equations, SIAM J. Numer. Anal. 25 (1988) 927–940.