Electronic Transactions on Numerical Analysis.
Volume 31, pp. 126-140, 2008.

ISSN 1068-9613.

ON A WEIGHTED QUASI-RESIDUAL MINIMIZATION STRATEGY FOR SOLVING COMPLEX SYMMETRIC SHIFTED LINEAR SYSTEMS*

T. SOGABE†, T. HOSHI‡, S.-L. ZHANG†, AND T. FUJIWARA§

Abstract. We consider the solution of complex symmetric shifted linear systems. Such systems arise in large-scale electronic structure simulations, and there is a strong need of algorithms for their fast solution. With the aim of solving the systems efficiently, we consider a special case of the QMR method for non-Hermitian shifted linear systems and propose its weighted quasi-minimal residual approach. A numerical algorithm, referred to as shifted QMR_SYM(B), is obtained by the choice of a weight which is particularly cost-effective. Numerical examples are presented to show the performance of the shifted QMR_SYM(B) method.

Key words. Complex symmetric matrices, shifted linear systems, Krylov methods, COCG, QMR_SYM.

AMS subject classifications. 65F10.

1. Introduction. In this paper we consider the solution of complex symmetric shifted linear systems of the form

$$(A + \sigma_\ell I)\boldsymbol{x}^{(\ell)} = \boldsymbol{b}, \quad \ell = 1, 2, \dots, m, \tag{1.1}$$

where $A(\sigma_\ell) := A + \sigma_\ell I$ are nonsingular $N \times N$ complex symmetric sparse matrices, i.e., $A(\sigma_\ell) = A(\sigma_\ell)^T \neq \overline{A}(\sigma_\ell)^T$, with scalar shifts $\sigma_\ell \in \mathbb{C}$, I is the $N \times N$ identity matrix, and $\boldsymbol{x}^{(\ell)}$, $\boldsymbol{b}$ are complex vectors of length N. Such systems arise in large-scale electronic structure simulations [15], and there is a strong need of algorithms for their fast solution.

Since the coefficient matrices of the given shifted linear systems (1.1) are sparse, it is natural to use Krylov subspace methods, and moreover since the coefficient matrices are complex symmetric, one of the simplest ways to solve the shifted linear systems consists of employing (preconditioned) Krylov subspace methods for solving complex symmetric linear systems, such as the COCG method [16], the COCR method [13], and the QMR_SYM method [3], to all the shifted linear systems (1.1). On the other hand, denoting the n-dimensional Krylov subspace with respect to A and $\boldsymbol{b}$ as $K_n(A, \boldsymbol{b}) := \mathrm{span}\{\boldsymbol{b}, A\boldsymbol{b}, \dots, A^{n-1}\boldsymbol{b}\}$, we observe that

$$K_n(A, \boldsymbol{b}) = K_n(A(\sigma_\ell), \boldsymbol{b}). \tag{1.2}$$

This implies that once the basis vectors are generated for one of the Krylov subspaces $K_n(A(\sigma_\ell), \boldsymbol{b})$, these basis vectors can be used to solve all the shifted linear systems. In other words, there is no need to generate all Krylov subspaces $K_n(A(\sigma_\ell), \boldsymbol{b})$, and thus the computational cost required by the basis generation, e.g., matrix-vector multiplications, is reduced. Here we give a concrete example: if we apply the conjugate orthogonal conjugate gradient (COCG) method to all the linear systems (1.1), then bases for $K_n(A(\sigma_\ell), \boldsymbol{b})$ are

* Received November 30, 2007. Accepted August 17, 2008. Published online on January 22, 2008. Recommended by Martin H. Gutknecht. This work was partially supported by KAKENHI (Grant No. 18760063, 19560065).

†Department of Computational Science and Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan ({sogabe,zhang}@na.cse.nagoya-u.ac.jp).

‡Department of Applied Mathematics and Physics, Tottori University, Japan, and Core Research for Evolutional Science and Technology, Japan Science and Technology Agency (CREST-JST), 4-1-8 Honcho, Kawaguchi-shi, Saitama 332-0012, Japan (hoshi@damp.tottori-u.ac.jp).

§Center for Research and Development of Higher Education, The University of Tokyo, Hongo, 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan, and Core Research for Evolutional Science and Technology, Japan Science and Technology Agency (CREST-JST), 4-1-8 Honcho, Kawaguchi-shi, Saitama 332-0012, Japan (fujiwara@coral.t.u-tokyo.ac.jp).

ETNA
Kent State University
http://etna.math.kent.edu

generated for $\ell = 1, 2, \ldots, m$. On the other hand, if we apply the COCG method to just one of the shifted linear systems (1.1) (referred to as the "$seed\ system$"), then the Krylov basis vectors are generated from the seed system and these vectors are used to solve the rest of the shifted linear systems.

Based on the observation (1.2), the shifted COCG method [15] has been recently proposed for solving complex symmetric shifted linear systems. The feature of the shifted COCG method is that the method performs COCG on a seed system and makes it possible to complete COCG for all shifted linear systems without further matrix-vector multiplications. The feature is completely different from some of the other well-known shifted linear solvers such as the shifted BiCGStab(ℓ) method [6], or the restarted shifted GMRES method [7], since they perform BiCGStab(ℓ) (or GMRES) on a seed system, but they apply a different method on the shifted linear systems, in order to keep the residuals collinear. The feature of the shifted COCG method plays a very important role in the seed switching technique [14], that avoids a minor problem of the shifted COCG method: one requires the choice of a seed system, and an unsuitable choice may lead to the undesirable result that some shifted linear systems remain unsolved.

There is another approach to solving the shifted linear systems (1.1), which consists of the use of Krylov subspace methods for non-Hermitian shifted linear systems, such as the shifted BiCGStab(ℓ) method [6], the shifted (TF)QMR method [4], the restarted shifted FOM method [10], and the restarted shifted GMRES method [7]; see also, e.g., [11]. We readily see that the relation (1.2) holds not only for complex symmetric matrices, but also for non-Hermitian matrices, and these methods are based on the use of this shift-invariant relation. Therefore, this can be a good approach. However, since these methods do not exploit the property of complex symmetric matrices, their computational cost can be more expensive than that of the shifted COCG method.

In this paper we consider the shifted QMR_SYM method, that is a special case of the QMR method for non-Hermitian shifted linear systems [4], and discuss the most time consuming part of it for a large number of shifted linear systems. Then, in order to reduce the cost, we propose a weighted quasi-minimal residual (WQMR) approach and propose a specific weight. We experimentally show the practical efficiency of the resulting algorithm, referred to as shifted QMR_SYM(B), when the number of shifted linear systems is large enough.

The present paper is organized as follows: in the next section, shifted QMR_SYM is described in order to specify the most time consuming part for a large number of shifted linear systems. In Section 3, we propose a WQMR approach with a specific weight for reducing the cost of the most time consuming part. The resulting algorithm, shifted QMR_SYM(B), and its properties are given. In Section 4, some results of numerical examples from electronic structure simulations are shown to ascertain the performance of the shifted QMR_SYM(B) method. Finally, some concluding remarks are made in Section 5.

Throughout this paper, unless otherwise stated, all vectors and matrices are assumed to be complex, $\overline{M}$, M^T, $M^H = \overline{M}^T$ denote the complex conjugate, transpose, and Hermitian transpose matrices of the matrix M, respectively, and $\|\boldsymbol{v}\|_W$ denotes the W-norm written as $(\boldsymbol{v}^H W \boldsymbol{v})^{1/2}$, where W is Hermitian positive definite.

2. The QMR_SYM method for solving complex symmetric shifted linear systems.

In this section, the shifted QMR_SYM method and its properties for solving complex symmetric shifted linear systems are introduced.

The QMR method for shifted linear systems was first formulated in [4] for the case of a general non-Hermitian matrix. Therefore, by simplifying the non-Hermitian Lanczos process [9], as it is known from other papers, such as [3, 16], a shifted simplified QMR

ETNA
Kent State University
http://etna.math.kent.edu

ALGORITHM 1 (Shifted QMR_SYM).

$\boldsymbol{x}_0^{(\ell)} = \boldsymbol{p}_{-1}^{(\ell)} = \boldsymbol{p}_0^{(\ell)} = \boldsymbol{0},\ \boldsymbol{v}_1 = \boldsymbol{b}/(\boldsymbol{b}^T\boldsymbol{b})^{1/2},\ g_1^{(\ell)} = (\boldsymbol{b}^T\boldsymbol{b})^{1/2},$

for $n = 1, 2, \ldots$ **do:**

(The complex symmetric Lanczos process)

$\alpha_n = \boldsymbol{v}_n^T A \boldsymbol{v}_n,$

$\widetilde{\boldsymbol{v}}_{n+1} = A\boldsymbol{v}_n - \alpha_n \boldsymbol{v}_n - \beta_{n-1}\boldsymbol{v}_{n-1},$

$\beta_n = (\widetilde{\boldsymbol{v}}_{n+1}^T \widetilde{\boldsymbol{v}}_{n+1})^{1/2},$

$\boldsymbol{v}_{n+1} = \widetilde{\boldsymbol{v}}_{n+1}/\beta_n,$

$t_{n-1,n}^{(\ell)} = \beta_{n-1},\ t_{n,n}^{(\ell)} = \alpha_n + \sigma_\ell,\ t_{n+1,n}^{(\ell)} = \beta_n,$

(Solve least squares problems by Givens rotations)

for $\ell = 1, 2, \ldots, m$ **do:**

if $\|\boldsymbol{r}_n^{(\ell)}\|_2/\|\boldsymbol{b}\|_2 > \epsilon,$ **then**

for $i = \max\{1, n-2\}, \ldots, n-1$ **do:**

$$\begin{bmatrix} t_{i,n}^{(\ell)} \\ t_{i+1,n}^{(\ell)} \end{bmatrix} = \begin{bmatrix} c_i^{(\ell)} & s_i^{(\ell)} \\ -\overline{s}_i^{(\ell)} & c_i^{(\ell)} \end{bmatrix} \begin{bmatrix} t_{i,n}^{(\ell)} \\ t_{i+1,n}^{(\ell)} \end{bmatrix},$$

end

$$c_n^{(\ell)} = \frac{|t_{n,n}^{(\ell)}|}{\sqrt{|t_{n,n}^{(\ell)}|^2 + |t_{n+1,n}^{(\ell)}|^2}},$$

$$\overline{s}_n^{(\ell)} = \frac{t_{n+1,n}^{(\ell)}}{t_{n,n}^{(\ell)}} c_n^{(\ell)},$$

$t_{n,n}^{(\ell)} = c_n^{(\ell)} t_{n,n}^{(\ell)} + s_n^{(\ell)} t_{n+1,n}^{(\ell)},$

$t_{n+1,n}^{(\ell)} = 0,$

$$\begin{bmatrix} g_n^{(\ell)} \\ g_{n+1}^{(\ell)} \end{bmatrix} = \begin{bmatrix} c_n^{(\ell)} & s_n^{(\ell)} \\ -\overline{s}_n^{(\ell)} & c_n^{(\ell)} \end{bmatrix} \begin{bmatrix} g_n^{(\ell)} \\ 0 \end{bmatrix},$$

(Update approximate solutions $\boldsymbol{x}_n^{(\ell)}$)

$\boldsymbol{p}_n^{(\ell)} = \boldsymbol{v}_n - (t_{n-2,n}^{(\ell)}/t_{n-2,n-2}^{(\ell)})\boldsymbol{p}_{n-2}^{(\ell)} - (t_{n-1,n}^{(\ell)}/t_{n-1,n-1}^{(\ell)})\boldsymbol{p}_{n-1}^{(\ell)},$

$\boldsymbol{x}_n^{(\ell)} = \boldsymbol{x}_{n-1}^{(\ell)} + (g_n^{(\ell)}/t_{n,n}^{(\ell)})\boldsymbol{p}_n^{(\ell)},$

end if

end

if $\|\boldsymbol{r}_n^{(\ell)}\|_2/\|\boldsymbol{b}\|_2 \le \epsilon$ for all $\ell,$ **then** exit.

end

---

method (shifted QMR_SYM) is readily obtained for the case of a complex symmetric matrix; see Algorithm 1. Algorithm 1 can be regarded as a natural combination of the results given in [3, 4].

In order to know that the numerical solution is accurate enough, one may need to compute the residual 2-norms. In that case, the following computation may be useful.

PROPOSITION 2.1 ([5]). *The 2-norms of the nth residuals of the approximate solutions $\boldsymbol{x}_n^{(\ell)}$ of the shifted QMR_SYM method are given by*

$$\|\boldsymbol{r}_n^{(\ell)}\|_2 = |g_{n+1}^{(\ell)}| \cdot \|\boldsymbol{w}_{n+1}^{(\ell)}\|_2, \quad \textit{for } \ell = 1, 2, \ldots, m,$$

where $\boldsymbol{w}_{n+1}^{(\ell)} = -s_n^{(\ell)}\boldsymbol{w}_n^{(\ell)} + c_n^{(\ell)}\boldsymbol{v}_{n+1}$, and $\boldsymbol{w}_1^{(\ell)} = \boldsymbol{v}_1$.

Proposition 2.1 is a result known to hold for the QMR method [5]. Therefore, it also holds for the above specialized variant. The rest of this section describes some special properties of the shifted QMR_SYM method.

PROPOSITION 2.2 ([2]). *Let $A \in \mathbf{R}^{N\times N}$ be real symmetric, $\sigma_\ell \in \mathbb{C}$ be complex shifts, and $\boldsymbol{b} \in \mathbf{R}^N$. Then the shifted QMR_SYM method (Algorithm 1) enjoys the following properties:*

- *(I) all matrix-vector multiplications can be done in real arithmetic;*
- *(II) an approximate solution at the nth iteration step has minimal residual 2-norms for each ℓ, i.e., the vectors $\boldsymbol{x}_n^{(\ell)}$'s are generated in order to minimize $\|\boldsymbol{r}_n^{(\ell)}\|_2$ over all $\boldsymbol{x}_n^{(\ell)} \in K_n(A, \boldsymbol{b})$;*
- *(III) $\|\boldsymbol{r}_n^{(\ell)}\|_2 = |g_{n+1}^{(\ell)}|$, for $\ell = 1, 2, \ldots, m$, $n \geq 0$.*

The above properties are known, since they have been proved for each individual shift; see [2] for details. These results may be very useful for large-scale electronic structure simulations [15] and a projection approach for eigenvalue problems [12], since there are complex symmetric shifted linear systems to be solved efficiently under the assumption of Proposition 2.2.

3. An iterative method for solving complex symmetric shifted linear systems.

In this section we consider complex symmetric shifted linear systems with a large number of shifts. For such systems, say $m \gg 1$, the most time consuming part of Algorithm 1 consists of generating approximate solutions, since the cost for the recurrences is $6Nm+3m$ per iteration step. In this section we propose a weighted quasi-minimal residual approach with a specific weight: in Subsection 3.1 we discuss the details of the approach, and in Subsection 3.2 we propose a specific weight to achieve the reduction of the cost.

3.1. A weighted quasi-minimal residual approach.

Before we formulate a Weighted Quasi-Minimal Residual (WQMR) approach, let us recall in Algorithm 2 the complex symmetric Lanczos process; see, e.g., Algorithm 2.1 in [3].

ALGORITHM 2 (The complex symmetric Lanczos process).

$\beta_0 = 0,\ \boldsymbol{v}_0 = \boldsymbol{0},\ \boldsymbol{r}_0 \neq \boldsymbol{0} \in \mathbb{C}^N,$
$\boldsymbol{v}_1 = \boldsymbol{r}_0/(\boldsymbol{r}_0^T\boldsymbol{r}_0)^{1/2},$
for $n = 1, 2, \ldots, m-1$ **do:**
  $\alpha_n = \boldsymbol{v}_n^T A\boldsymbol{v}_n,$
  $\widetilde{\boldsymbol{v}}_{n+1} = A\boldsymbol{v}_n - \alpha_n\boldsymbol{v}_n - \beta_{n-1}\boldsymbol{v}_{n-1},$
  $\beta_n = (\widetilde{\boldsymbol{v}}_{n+1}^T\widetilde{\boldsymbol{v}}_{n+1})^{1/2},$
  $\boldsymbol{v}_{n+1} = \widetilde{\boldsymbol{v}}_{n+1}/\beta_n.$
end

The matrix form of Algorithm 2 can be formulated as follows. Let $T_{n+1,n}$ and T_n be the $(n+1)\times n$ and $n\times n$ tridiagonal matrices whose entries are the recurrence coefficients of

ETNA
Kent State University
http://etna.math.kent.edu

the complex symmetric Lanczos process, which are given by

$$T_{n+1,n} := \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \\ & & & \beta_n \end{bmatrix}, \quad T_n := \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix},$$

and let V_n be the $N \times n$ matrix with the Lanczos vectors as columns, i.e.,

$$V_n := (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n).$$

Then, it follows from Algorithm 2 that

$$AV_n = V_{n+1}T_{n+1,n} = V_nT_n + \beta_n \boldsymbol{v}_{n+1}\boldsymbol{e}_n^T, \tag{3.1}$$

where $\boldsymbol{e}_n = (0, 0, \ldots, 1)^T \in \mathbf{R}^n$.

Now we are ready to describe the WQMR approach. Let $\boldsymbol{x}_n^{(\ell)}$ be approximate solutions at the nth iteration step for the systems (1.1), which are given by

$$\boldsymbol{x}_n^{(\ell)} = V_n \boldsymbol{y}_n^{(\ell)}, \quad \ell = 1, 2, \ldots, m, \tag{3.2}$$

where $\boldsymbol{y}_n^{(\ell)} \in \mathbb{C}^n$. Then, from the definition of residual vectors $\boldsymbol{r}_n^{(\ell)} := \boldsymbol{b} - (A + \sigma_\ell I)\boldsymbol{x}_n^{(\ell)}$, the update formulas (3.2), and the matrix form of the complex symmetric Lanczos process (3.1), we readily obtain

$$\boldsymbol{r}_n^{(\ell)} = V_{n+1}\left(g_1\boldsymbol{e}_1 - T_{n+1,n}^{(\ell)}\boldsymbol{y}_n^{(\ell)}\right), \quad \text{where } T_{n+1,n}^{(\ell)} := T_{n+1,n} + \sigma_\ell \begin{bmatrix} I_n \\ \mathbf{0}^T \end{bmatrix}. \tag{3.3}$$

Here, $\boldsymbol{e}_1$=$(1, 0, \ldots, 0)^T$ is the first unit vector, and $g_1 = (\boldsymbol{b}^T\boldsymbol{b})^{1/2}$. It is natural to determine $\boldsymbol{y}_n^{(\ell)}$ such that all residual 2-norms $\|\boldsymbol{r}_n^{(\ell)}\|_2$ are minimized. However, this choice for $\boldsymbol{y}_n^{(\ell)}$ is unfeasible due to large computational costs. Hence, the vectors $\boldsymbol{y}_n^{(\ell)}$ are determined by an alternative approach, i.e., by solving the following weighted least squares problems:

$$\boldsymbol{y}_n^{(\ell)} = \arg \min_{\boldsymbol{z}_n^{(\ell)} \in \mathbb{C}^n} \left\| g_1\boldsymbol{e}_1 - T_{n+1,n}^{(\ell)}\boldsymbol{z}_n^{(\ell)} \right\|_{W_{n+1}^H W_{n+1}}, \tag{3.4}$$

where W_{n+1} is an $(n+1) \times (n+1)$ nonsingular matrix. Thus $W_{n+1}^H W_{n+1}$ can be used as a weight since it is Hermitian positive definite. One of the simplest choices for W_{n+1} is the identity matrix. In this case, from $W_{n+1} = I_{n+1}$ we have

$$\boldsymbol{y}_n^{(\ell)} = \arg \min_{\boldsymbol{z}_n^{(\ell)} \in \mathbb{C}^n} \left\| g_1\boldsymbol{e}_1 - T_{n+1,n}^{(\ell)}\boldsymbol{z}_n^{(\ell)} \right\|_2. \tag{3.5}$$

The vector that is minimized is called quasi-residual. Algorithm 1 is obtained by solving (3.5) using Givens rotations; see, e.g., [8, p. 215].

A slightly generalized choice, proposed in [5], is

$$W_{n+1} = \Omega_{n+1} := \mathrm{diag}(\omega_1, \omega_2, \ldots, \omega_{n+1}),$$

with $\omega_i > 0$ for all i. Then, it follows that

$$\boldsymbol{y}_n^{(\ell)} = \arg \min_{\boldsymbol{z}_n^{(\ell)} \in \mathbb{C}^n} \left\| \omega_1 g_1\boldsymbol{e}_1 - \Omega_{n+1}T_{n+1,n}^{(\ell)}\boldsymbol{z}_n^{(\ell)} \right\|_2.$$

Among the various possible choices for ω_i, a natural one is $\omega_i = \|\boldsymbol{v}_i\|_2$, since then Ω_{n+1} contains the diagonal entries of the upper triangular matrix R_{n+1}, which corresponds to the QR factorization of V_{n+1}. If we choose $W_{n+1} = R_{n+1}$, where $V_{n+1} = Q_{n+1}R_{n+1}$, then from (3.3) and (3.4) we have

$$\begin{aligned}
&\min_{\boldsymbol{z}_n^{(\ell)} \in \mathbb{C}^n} \left\| g_1 \boldsymbol{e}_1 - T_{n+1,n}^{(\ell)} \boldsymbol{z}_n^{(\ell)} \right\|_{R_{n+1}^H R_{n+1}} \\
&\qquad = \min_{\boldsymbol{z}_n^{(\ell)} \in \mathbb{C}^n} \left\| g_1 R_{n+1} \boldsymbol{e}_1 - R_{n+1} T_{n+1,n}^{(\ell)} \boldsymbol{z}_n^{(\ell)} \right\|_2 \\
&\qquad = \min_{\boldsymbol{z}_n^{(\ell)} \in \mathbb{C}^n} \left\| Q_{n+1} R_{n+1} \big(g_1 \boldsymbol{e}_1 - T_{n+1,n}^{(\ell)} \boldsymbol{z}_n^{(\ell)} \big) \right\|_2 \\
&\qquad = \min_{\boldsymbol{z}_n^{(\ell)} \in \mathbb{C}^n} \left\| V_{n+1} \big(g_1 \boldsymbol{e}_1 - T_{n+1,n}^{(\ell)} \boldsymbol{z}_n^{(\ell)} \big) \right\|_2 = \min_{\boldsymbol{z}_n^{(\ell)} \in \mathbb{C}^n} \left\| \boldsymbol{r}_n^{(\ell)} \right\|_2 .
\end{aligned}$$

By solving the above weighted least squares problems, all residual 2-norms are minimized, hence $W_{n+1} = \Omega_{n+1}$ is a reasonable choice. For each individual shift, the resulting algorithm is the same as [3, Algorithm 3.2].

3.2. A choice of the weight suitable for a large number of shifts. In the previous subsection, we have described the WQMR approach and mentioned that the choice of the weight $W_{n+1}^H W_{n+1}$ with $W_{n+1} = I_{n+1}$ leads to the shifted QMR_SYM method (Algorithm 1). Under the assumption of Proposition 2.2, the shifted QMR_SYM method is ideal in the sense of Faber-Manteuffel's Theorem [1], since it enjoys minimal residual property and requires short-term recurrences for updating approximate solutions, and thus one may think that there is no need to choose other possible weights. However, we will show in this subsection that, even under the assumption of Proposition 2.2, there is a suitable weight for the WQMR approach. The motivation for the choice of the weight mainly comes from the freedom of the number m of complex symmetric shifted linear systems.

Now we consider the computational costs of Algorithm 1 for a large number of complex symmetric shifted linear systems, i.e., $m \gg 1$. In this case, we readily see that computing the recurrences for updating approximate solutions in Algorithm 1 is the most time-consuming part, due to a cost of $6Nm + 3m$ per iteration step. Hence, we will now consider a weight to reduce the computational cost for the recurrences of $\boldsymbol{x}_n^{(\ell)}$. To achieve this we propose the following choice:

$$W_{n+1} = L_{n+1}^{(\ell)}, \quad \text{such that } L_{n+1}^{(\ell)} T_{n+1,n}^{(\ell)} = \begin{bmatrix} B_n^{(\ell)} \\ \mathbf{0}^T \end{bmatrix}, \tag{3.6}$$

where $B_n^{(\ell)}$ is an $n \times n$ upper bidiagonal matrix of the form

$$B_n^{(\ell)} := \begin{bmatrix} t_{1,1}^{(\ell)} & t_{1,2}^{(\ell)} & & \\ & t_{2,2}^{(\ell)} & \ddots & \\ & & \ddots & t_{n-1,n}^{(\ell)} \\ & & & t_{n,n}^{(\ell)} \end{bmatrix},$$

and $L_{n+1}^{(\ell)}$ is lower triangular and will be specified below.

Next, we derive recurrence formulas for updating the approximate solutions $\boldsymbol{x}_n^{(\ell)}$. From formula (3.4), with the choice $W_{n+1} = L_{n+1}^{(\ell)}$ of (3.6), it follows that

$$\begin{aligned} \boldsymbol{y}_n^{(\ell)} &= \arg \min_{\boldsymbol{z}_n^{(\ell)} \in \mathbb{C}^n} \left\| g_1 \boldsymbol{e}_1 - T_{n+1,n}^{(\ell)} \boldsymbol{z}_n^{(\ell)} \right\|_{(L_{n+1}^{(\ell)})^H L_{n+1}^{(\ell)}} \\ &= \arg \min_{\boldsymbol{z}_n^{(\ell)} \in \mathbb{C}^n} \left\| g_1 L_{n+1}^{(\ell)} \boldsymbol{e}_1 - L_{n+1}^{(\ell)} T_{n+1,n}^{(\ell)} \boldsymbol{z}_n^{(\ell)} \right\|_2 \\ &= \arg \min_{\boldsymbol{z}_n^{(\ell)} \in \mathbb{C}^n} \left\| \begin{bmatrix} \widetilde{\boldsymbol{g}}_n^{(\ell)} \\ \widetilde{g}_{n+1}^{(\ell)} \end{bmatrix} - \begin{bmatrix} B_n^{(\ell)} \\ \boldsymbol{0}^T \end{bmatrix} \boldsymbol{z}_n^{(\ell)} \right\|_2, \quad \begin{bmatrix} \widetilde{\boldsymbol{g}}_n^{(\ell)} \\ \widetilde{g}_{n+1}^{(\ell)} \end{bmatrix} := g_1 L_{n+1}^{(\ell)} \boldsymbol{e}_1. \end{aligned} \tag{3.7}$$

From the above least squares problems, we readily see that $\boldsymbol{y}_n^{(\ell)} = \left(B_n^{(\ell)}\right)^{-1} \widetilde{\boldsymbol{g}}_n^{(\ell)}$. Hence, it follows from (3.2), using $(\widetilde{\boldsymbol{p}}_1\ \widetilde{\boldsymbol{p}}_2\ \cdots\ \widetilde{\boldsymbol{p}}_n) := V_n (B_n^{(\ell)})^{-1}$, that we have the following coupled two-term recurrence relations:

$$\begin{aligned} \widetilde{\boldsymbol{p}}_n^{(\ell)} &= (\boldsymbol{v}_n - t_{n-1,n}^{(\ell)} \widetilde{\boldsymbol{p}}_{n-1}^{(\ell)}) / t_{n,n}^{(\ell)}, \\ \boldsymbol{x}_n^{(\ell)} &= \boldsymbol{x}_{n-1}^{(\ell)} + \widetilde{g}_n^{(\ell)} \widetilde{\boldsymbol{p}}_n^{(\ell)}. \end{aligned}$$

The cost per iteration is now $5Nm$. Substituting $\boldsymbol{p}_i^{(\ell)} = t_{i,i}^{(\ell)} \widetilde{\boldsymbol{p}}_i^{(\ell)}$ into the above recurrences, we have the even more efficient recurrence formulas

$$\begin{aligned} \boldsymbol{p}_n^{(\ell)} &= \boldsymbol{v}_n - (t_{n-1,n}^{(\ell)} / t_{n-1,n-1}^{(\ell)}) \boldsymbol{p}_{n-1}^{(\ell)}, \\ \boldsymbol{x}_n^{(\ell)} &= \boldsymbol{x}_{n-1}^{(\ell)} + (\widetilde{g}_n^{(\ell)} / t_{n,n}^{(\ell)}) \boldsymbol{p}_n^{(\ell)}. \end{aligned}$$

By this reformulation, the cost becomes $4Nm + 2m$.

Next, let us consider a choice for $L_{n+1}^{(\ell)}$ satisfying (3.6). Let $F_n^{(\ell)}(i)$ be an $n \times n$ matrix of the form

$$F_n^{(\ell)}(i) := \begin{bmatrix} I_{i-1} & & & \\ & 1 & & \\ & f_i^{(\ell)} & 1 & \\ & & & I_{n-i-1} \end{bmatrix}, \tag{3.8}$$

and let T be tridiagonal. Then, by determining $f_i^{(\ell)}$ such that the $(i+1, i)$ entry of the matrix $F_n^{(\ell)}(i) \cdot T$ is zero, we can fulfill (3.6) in the following way:

$$F_{n+1}^{(\ell)}(n) F_{n+1}^{(\ell)}(n-1) \cdots F_{n+1}^{(\ell)}(1) T_{n+1,n}^{(\ell)} = \begin{bmatrix} B_n^{(\ell)} \\ \boldsymbol{0}^T \end{bmatrix}. \tag{3.9}$$

From the above formula, we see that $F_{n+1}^{(\ell)}(n) F_{n+1}^{(\ell)}(n-1) \cdots F_{n+1}^{(\ell)}(1) = L_{n+1}^{(\ell)}$, and thus $L_{n+1}^{(\ell)}$ and $L_n^{(\ell)}$ are related by

$$L_{n+1}^{(\ell)} = F_{n+1}^{(\ell)}(n) \begin{bmatrix} L_n^{(\ell)} & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{bmatrix}, \quad \text{for } n = 2, 3, \ldots, \tag{3.10}$$

where $L_2^{(\ell)} = F_2^{(\ell)}(1)$. We readily see that the matrices $L_{n+1}^{(\ell)}$ are lower triangular with all ones on the diagonals, and this property will be used in the proof of Proposition 3.1. The shifted QMR_SYM method with the weight $(L_{n+1}^{(\ell)})^H L_{n+1}^{(\ell)}$ is referred to as shifted QMR_SYM(B); see Algorithm 3.

ALGORITHM 3 (Shifted QMR_SYM(B)).

$\boldsymbol{x}_0^{(\ell)} = \boldsymbol{p}_{-1}^{(\ell)} = \boldsymbol{p}_0^{(\ell)} = \mathbf{0},\ \boldsymbol{v}_1 = \boldsymbol{b}/(\boldsymbol{b}^T\boldsymbol{b})^{1/2},\ \widetilde{g}_1^{(\ell)} = (\boldsymbol{b}^T\boldsymbol{b})^{1/2},$
for $n = 1, 2, \dots$ **do:**
  (The complex symmetric Lanczos process)
  $\alpha_n = \boldsymbol{v}_n^T A \boldsymbol{v}_n,$
  $\widetilde{\boldsymbol{v}}_{n+1} = A\boldsymbol{v}_n - \alpha_n \boldsymbol{v}_n - \beta_{n-1}\boldsymbol{v}_{n-1},$
  $\beta_n = (\widetilde{\boldsymbol{v}}_{n+1}^T \widetilde{\boldsymbol{v}}_{n+1})^{1/2},$
  $\boldsymbol{v}_{n+1} = \widetilde{\boldsymbol{v}}_{n+1}/\beta_n,$
  $t_{n-1,n}^{(\ell)} = \beta_{n-1},\ t_{n,n}^{(\ell)} = \alpha_n + \sigma_\ell,\ t_{n+1,n}^{(\ell)} = \beta_n,$
  (Solve weighted least squares problems)
  for $\ell = 1, 2, \dots, m$ **do:**
    if $\|\boldsymbol{r}_n^{(\ell)}\|_2 / \|\boldsymbol{b}\|_2 > \epsilon,$ **then**
      for $i = \max\{1, n-1\}, \dots, n-1$ **do:**
        $t_{i+1,n}^{(\ell)} = f_i^{(\ell)} t_{i,n}^{(\ell)} + t_{i+1,n}^{(\ell)},$
      end
      $f_n^{(\ell)} = -\dfrac{t_{n+1,n}^{(\ell)}}{t_{n,n}^{(\ell)}},$
      $t_{n+1,n}^{(\ell)} = 0,$
      $\widetilde{g}_{n+1}^{(\ell)} = f_n^{(\ell)} \widetilde{g}_n^{(\ell)},$
      (Update approximate solutions $\boldsymbol{x}_n^{(\ell)}$)
      $\boldsymbol{p}_n^{(\ell)} = \boldsymbol{v}_n - (t_{n-1,n}^{(\ell)}/t_{n-1,n-1}^{(\ell)})\boldsymbol{p}_{n-1}^{(\ell)},$
      $\boldsymbol{x}_n^{(\ell)} = \boldsymbol{x}_{n-1}^{(\ell)} + (\widetilde{g}_n^{(\ell)}/t_{n,n}^{(\ell)})\boldsymbol{p}_n^{(\ell)},$
    end if
  end
    if $\|\boldsymbol{r}_n^{(\ell)}\|_2/\|\boldsymbol{b}\|_2 \le \epsilon$ for all $\ell,$ **then** exit.
end

As in the case of Proposition 2.1, there is an efficient way to evaluate the residual 2-norms as follows.

PROPOSITION 3.1. *The nth residual 2-norms of the approximate solutions $\boldsymbol{x}_n^{(\ell)}$ for the shifted QMR_SYM(B) method are given by*

$$\|\boldsymbol{r}_n^{(\ell)}\|_2 = |\widetilde{g}_{n+1}^{(\ell)}| \cdot \|\boldsymbol{v}_{n+1}\|_2, \quad \textit{for } \ell = 1, 2, \dots, m.$$

Proof. The proof is similar to that of Proposition 2.1. It follows from (3.3), (3.6), (3.7), and recalling $\boldsymbol{y}_n^{(\ell)} = \left(B_n^{(\ell)}\right)^{-1}\widetilde{\boldsymbol{g}}_n^{(\ell)}$, that we have

$$\boldsymbol{r}_n^{(\ell)} = \widetilde{g}_{n+1}^{(\ell)} V_{n+1} \left(L_{n+1}^{(\ell)}\right)^{-1} \boldsymbol{e}_{n+1}.$$

From (3.8) and (3.9) $L_{n+1}^{(\ell)}$ is a lower triangular matrix with all ones on the diagonals. Thus, $\left(L_{n+1}^{(\ell)}\right)^{-1}$ is also a lower triangular matrix with all ones on the diagonals. It follows that

ETNA
Kent State University
http://etna.math.kent.edu

$\left(L_{n+1}^{(\ell)}\right)^{-1}\boldsymbol{e}_{n+1} = \boldsymbol{e}_{n+1}$, and thus we have $\boldsymbol{r}_n^{(\ell)} = \widetilde{g}_{n+1}^{(\ell)} V_{n+1}\boldsymbol{e}_{n+1} = \widetilde{g}_{n+1}^{(\ell)}\boldsymbol{v}_{n+1}$, which concludes the proof. □

When we solve a large number of shifted systems, the computational cost of the residual 2-norms for the shifted QMR_SYM(B) method is much cheaper than that for the shifted QMR_SYM method, since the former cost is of order N and the latter is of order Nm per iteration step.

Observing the algorithms of the shifted QMR_SYM and the shifted QMR_SYM(B) methods, we see that the work for the weighted least squares problems and for updating the approximate solutions in the shifted QMR_SYM(B) method is less than that in the shifted QMR_SYM method. For the case $m \gg 1$, the most time-consuming part is updating the approximate solutions. In this case, the shifted QMR_SYM(B) method may be more efficient than the shifted QMR_SYM method, since the shifted QMR_SYM(B) method requires $4Nm+2m$ operations per iteration step for the update, while the shifted QMR_SYM method needs $6Nm + 3m$ operations. This difference will be clearer when we will use the results of Propositions 2.1 and 3.1 as a stopping criterion. On the other hand, in terms of number of iterations, the convergence of the shifted QMR_SYM(B) method is worse than that of the shifted QMR_SYM method, but not worse than that of the shifted COCG method, as stated in the following proposition.

PROPOSITION 3.2. *Under the assumption of Proposition 2.2, the shifted QMR_SYM(B) method (Algorithm 3) enjoys the following properties:*

(I) all matrix-vector multiplications can be done in real arithmetic;

(II) if breakdown does not occur and each matrix $T_n + \sigma_\ell I_n$ is nonsingular, then

$$\left\|\boldsymbol{r}_n^{(\ell),SQ(B)}\right\|_2 = \left\|\boldsymbol{r}_n^{(\ell),SCOCG}\right\|_2 \geq \left\|\boldsymbol{r}_n^{(\ell),SQ}\right\|_2, \quad \textit{for } \ell = 1, 2, \ldots m,$$

where the superscripts SQ(B), SCOCG, and SQ are short for shifted QMR_SYM(B), shifted COCG, and shifted QMR_SYM, respectively;

(III) $\|\boldsymbol{r}_n^{(\ell),SQ(B)}\|_2 = |\widetilde{g}_{n+1}^{(\ell)}|$, for $\ell = 1, 2, \ldots, m$, $n \geq 0$.

Proof. The proof of (I) is the same as that of Proposition 2.2, and is based on the fact that, under the assumption of the theorem, the complex symmetric Lanczos process generates real basis vectors.

Next, we give a proof of (II). The nth residuals of the shifted COCG method [15] belong to $\boldsymbol{b} - (A + \sigma_\ell I)K_n(A + \sigma_\ell I, \boldsymbol{b})$. Hence, each $\boldsymbol{r}_n^{(\ell),\mathrm{SCOCG}}$ can be written as

$$\boldsymbol{r}_n^{(\ell),\mathrm{SCOCG}} = \boldsymbol{b} - (A + \sigma_\ell I)V_n\boldsymbol{y}_n^{(\ell),\mathrm{SCOCG}},$$

where V_n is the same matrix as in (3.1). Since each $\boldsymbol{r}_n^{(\ell),\mathrm{SCOCG}}$ is orthogonal to each subspace $K_n(\overline{A} + \overline{\sigma}_\ell I, \overline{\boldsymbol{b}}) = K_n(\overline{A}, \overline{\boldsymbol{b}})$, i.e., $\boldsymbol{r}_n^{(\ell),\mathrm{SCOCG}} \perp K_n(\overline{A}, \overline{\boldsymbol{b}})$, we have

$$V_n^T\boldsymbol{b} - V_n^T(A + \sigma_\ell I)V_n\boldsymbol{y}^{(\ell),\mathrm{SCOCG}} = 0,$$

and thus from (3.1) it follows that

$$\boldsymbol{y}_n^{(\ell),\mathrm{SCOCG}} = (V_n^T(A + \sigma_\ell I)V_n)^{-1}V_n^T\boldsymbol{b} = g_1(T_n^{(\ell)})^{-1}\boldsymbol{e}_1,$$

where $T_n^{(\ell)} := T_n + \sigma_\ell I_n$. Since the shifted QMR_SYM(B) method has the form (3.3), it is sufficient to show that $\boldsymbol{y}_n^{(\ell),\mathrm{SCOCG}} = \boldsymbol{y}_n^{(\ell),\mathrm{SQ}(B)}$. From (3.7) and (3.10) it follows that

$$\begin{aligned}\boldsymbol{y}_n^{(\ell),\mathrm{SQ}(B)} &= (B_n^{(\ell)})^{-1}\widetilde{\boldsymbol{g}}_n^{(\ell)} = g_1(B_n^{(\ell)})^{-1}[I_n|\boldsymbol{0}]L_{n+1}^{(\ell)}[\boldsymbol{e}_1^T|0]^T \\ &= g_1(B_n^{(\ell)})^{-1}L_n^{(\ell)}\boldsymbol{e}_1 = g_1((L_n^{(\ell)})^{-1}B_n^{(\ell)})^{-1}\boldsymbol{e}_1.\end{aligned}$$

Since from (3.9) and (3.10) we can readily confirm the relation $L_n^{(\ell)}T_n^{(\ell)} = B_n^{(\ell)}$, we have $\boldsymbol{y}_n^{(\ell),\mathrm{SQ}(B)} = g_1(T_n^{(\ell)})^{-1}\boldsymbol{e}_1$, which is the same as $\boldsymbol{y}_n^{(\ell),\mathrm{SCOCG}}$. The inequality in (*II*) follows from Proposition 2.2, since under the given assumption the shifted QMR_SYM method enjoys the minimal residual property.

Finally, we give a proof of (*III*). If follows from the proof of (*I*) that $\|\boldsymbol{v}_i\|_2 = 1$ for all i. Thus from Proposition 3.1 we have

$$\|\boldsymbol{r}_n^{(\ell),\mathrm{SQ}(B)}\|_2 = |\widetilde{g}_{n+1}^{(\ell)}| \cdot \|\boldsymbol{v}_{n+1}\|_2 = |\widetilde{g}_{n+1}^{(\ell)}|, \quad \text{for } \ell = 1, 2, \dots, m,\ n \geq 0. \qquad \square$$

We observe that, in property (*II*) of Proposition 3.2, breakdown may occur due to the choice (3.8) of the weighted least squares problems.

From Proposition 3.2 we see that, in terms of the number of iteration steps, the shifted QMR_SYM(B) method never converges faster than the shifted QMR_SYM method, but it converges at the same iteration step as the shifted COCG method does. Since the efficiency of the shifted COCG method has already been shown, and the computational cost of the shifted QMR_SYM(B) method for the case $m \gg 1$ is much less than that of the shifted QMR_SYM method, the shifted QMR_SYM(B) method can also be useful. This is supported by some numerical examples in the next section.

4. Numerical examples. In this section, we report on some numerical examples concerning the shifted COCG method, the shifted QMR_SYM method (Algorithm 1), and the shifted QMR_SYM(B) method (Algorithm 3). We evaluate these methods in terms of computation time. All tests were performed on a workstation with a 2.6GHz AMD Opteron(tm) processor 252 using double precision arithmetic. Codes were written in Fortran 77 and compiled with g77 -$O3$. In all cases the stopping criterion was set as $\epsilon = 10^{-12}$.

4.1. Example 1. The first problem comes from the electronic structure computation of a bulk Si with 512 atoms (see [15]) which is written as follows:

$$(\sigma_\ell I - H)\boldsymbol{x}^{(\ell)} = \boldsymbol{e}_1, \quad \ell = 1, 2, \dots, m,$$

where $\sigma_\ell = 0.400 + (\ell - 1 + i)/1000$, $H \in R^{2048\times 2048}$ is a symmetric matrix with 139264 entries, $\boldsymbol{e}_1 = (1, 0, \dots, 0)^T$, and $m = 1001$. Since the shifted COCG method requires the choice of a seed system, we have chosen the optimal seed ($\ell = 714$) in this problem; otherwise some linear systems would remain unsolved.

Figure 4.1 shows the number of iterations of each method to solve the ℓth shifted linear systems. For example, in Figure 4.1, the number of iterations for the shifted COCG method at $\ell = 600$ is 150, which means the shifted COCG method required 150 iterations to obtain the (approximate) solutions of the 600th shifted linear system, i.e., $(\sigma_{600}I - H)\boldsymbol{x}^{(600)} = \boldsymbol{e}_1$.

From Figure 4.1 we make three observations: first, the three methods required almost the same number of iterations at each ℓ; second, in terms of number of iterations, the shifted QMR_SYM method often converged slightly faster than the other two methods. This phenomenon is closely related to Proposition 2.2, as it will become clearer later; third, for each method the required number of iterations depends highly on the shift parameters σ_ℓ. This result may come from the shifted eigenvalues of the coefficient matrices $\sigma_\ell I - H$, since if we choose σ_ℓ close to an eigenvalue of H, then $\sigma_\ell I - H$ is close to singular. Conversely, from the shape of the graphs in Figure 4.1 one may obtain a partial distribution of eigenvalues of H.

The history of the residual 2-norm for a particular shifted system is reported in Figure 4.2. From it we see that the relative residual 2-norm of the shifted QMR_SYM method decreases monotonically, and at every iteration step the norm is less than those of the other two methods.

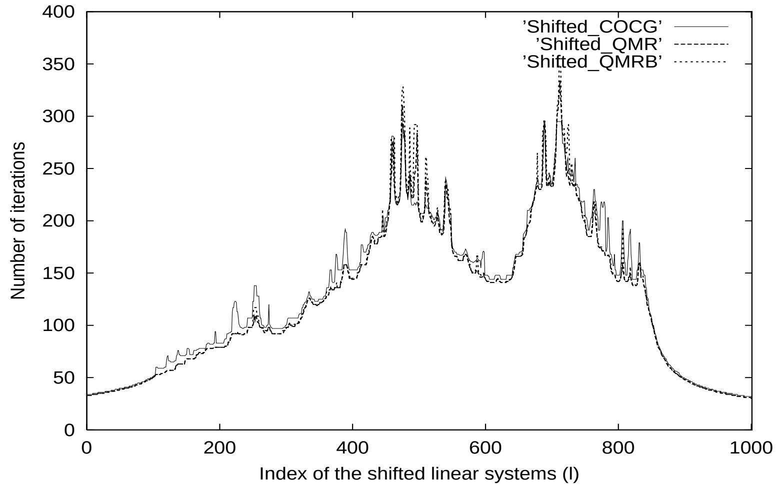

FIGURE 4.1. *Number of iterations for the shifted COCG method, the shifted QMR_SYM method, and the shifted QMR_SYM(B) method versus the index of the shifted linear systems.*

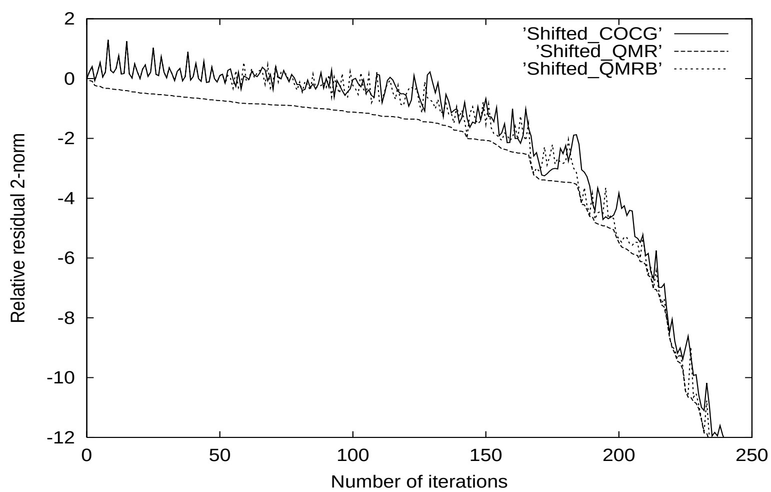

FIGURE 4.2. Log$_{10}$ *of the relative residual 2-norms versus the number of iterations of the shifted COCG method, the shifted QMR_SYM method, and the shifted QMR_SYM(B) method for the shifted linear system with* $\ell = 701, i.e., \sigma_{701} = 1.100 + 0.001i$.

Hence, we can say that the property (II) of Proposition 2.2 is experimentally supported. We also observe that, during the first fifty iterations, the shifted COCG method and the shifted QMR_SYM(B) method behave exactly in the same way. After that, their histories varies gradually. Hence, also the property (II) of Proposition 3.2 is experimentally supported.

The computation times of the three methods are given in Figure 4.3, where the horizontal axis denotes the number m of shifted linear systems that are solved. For example, in Figure 4.3, the computation time of the shifted COCG method at $m = 200$ is about 0.76 sec., which means that it requires about 0.76 sec. to solve the shifted linear systems: $((0.400 + 0.001i)I - H)\boldsymbol{x}^{(1)} = \boldsymbol{e}_1$, $((0.401 + 0.001i)I - H)\boldsymbol{x}^{(2)} = \boldsymbol{e}_1$, $\ldots$, $((0.599 + 0.001i)I - H)\boldsymbol{x}^{(200)} = \boldsymbol{e}_1$. From Figure 4.3 we see that, as the number m grows

ETNA
Kent State University
http://etna.math.kent.edu

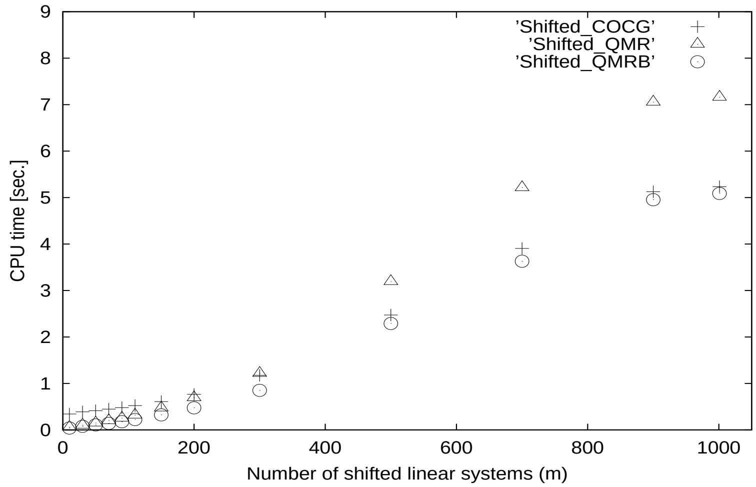

FIGURE 4.3. *CPU time, in seconds, versus the number of shifted linear systems for each iterative method.*

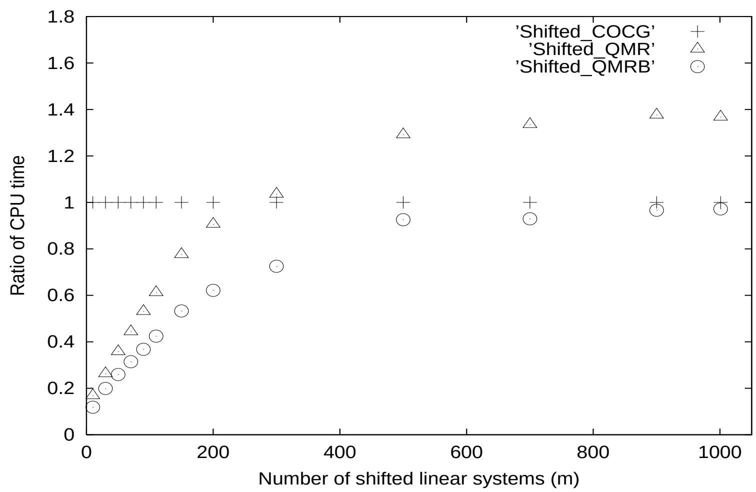

FIGURE 4.4. *The ratio between each computation time and the one of the shifted COCG method, versus the number of shifted linear systems.*

larger, the shifted QMR_SYM method requires more CPU time than the other two methods. On the other hand, the shifted QMR_SYM(B) method requires almost the same CPU time as the shifted COCG method. This phenomena can be attributed to the computational costs of updating the approximate solutions for each method and, in particular, to the following three facts: first, we know from Figure 4.1 that the three methods require almost the same number of iterations; second, the shifted QMR_SYM(B) method has almost the same computational cost than the shifted COCG method, while the shifted QMR_SYM method tends to require a larger cost per iteration than the other two methods; third, for large m, updating the approximate solutions is one of the most time-consuming parts. In the previous two sections, we already discussed the latter two facts.

In Figure 4.3 we can see little about the properties of the three methods for small ℓ. We therefore display in Figure 4.4 the ratio between each computation time and the timing

ETNA
Kent State University
http://etna.math.kent.edu

of the shifted COCG method. We see that the shifted QMR_SYM method and the shifted QMR_SYM(B) method converge much faster than the shifted COCG method when the number of shifted linear systems is small, say, $m < 200$. A possible explanation is that, for small m, updating the approximate solutions does not affect the CPU time so much. Other operations, such as matrix-vector multiplications, are now the most time-consuming parts, since the three methods require almost the same number of iterations; see Figure 4.1. From Proposition 2.2 (I) and Proposition 3.2 (I) we know that in this case the shifted QMR_SYM method and the shifted QMR_SYM(B) method require only real matrix-real vector multiplications. On the other hand, the shifted COCG method requires real matrix-complex vector multiplications. Moreover, dot products and vector additions of the complex symmetric Lanczos process used in the shifted QMR_SYM method and the shifted QMR_SYM(B) method can be done in real arithmetic. Hence, the two methods converge much faster than the shifted COCG method.

4.2. Example 2. The second problem comes from the electronic structure computation of bulk fcc Cu with 1568 atoms (see [15]):

$$(\sigma_\ell I - H)\boldsymbol{x}^{(\ell)} = \boldsymbol{e}_1, \quad \ell = 1, 2, \ldots, m,$$

where $\sigma_\ell = -0.5 + (\ell - 1 + i)/1000$, $H \in R^{14112\times 14112}$ is a symmetric matrix with 3924704 entries, $\boldsymbol{e}_1 = (1, 0, \ldots, 0)^T$, and $m = 1501$.

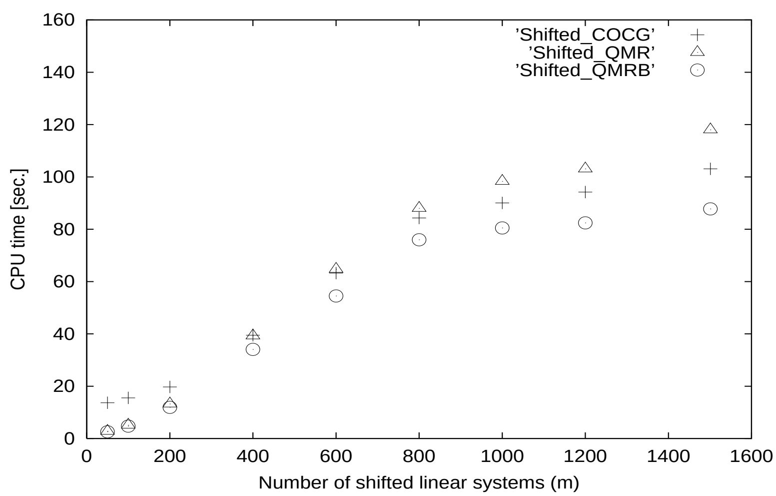

FIGURE 4.5. *CPU time, in seconds, versus the number of shifted linear systems for each iterative method.*

The computation times of the three methods for solving the m shifted linear systems is shown in Figure 4.5. The ratio between each computation time and the timing of the shifted COCG method is shown in Figure 4.6. From these figures we see that, even if the size of this matrix is about 7 times larger than before, the three methods behave similarly to the previous example.

5. Concluding remarks. In this paper, the shifted QMR_SYM method was described as a specialization of the QMR method for general non-Hermitian shifted linear systems [4]. The advantage of the method, with respect to the shifted COCG method, is that there is no need to choose a suitable seed system. On the other hand, we have found that, for a large number of shifted linear systems, the most time-consuming part of the shifted QMR_SYM

ETNA
Kent State University
http://etna.math.kent.edu

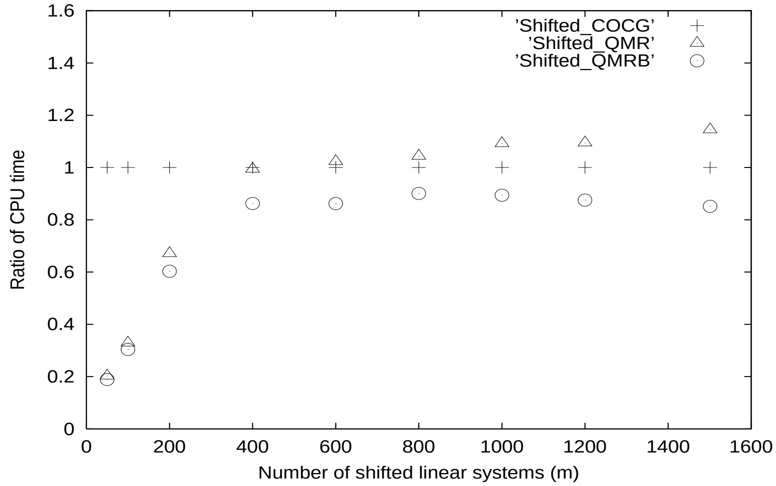

FIGURE 4.6. *The ratio between each computation time and the one of the shifted COCG method, versus the number of shifted linear systems.*

method is updating the approximate solutions, and this cost is higher than that of the shifted COCG method. We therefore have proposed the weighted quasi-minimal residual approach, with a weight particularly suited to reduce the computational cost for updating the approximate solutions. Also the resulting method, shifted QMR_SYM(B), does not require to choose a suitable seed system, which is an advantage over the shifted COCG method. From numerical experiments we have learned that shifted QMR_SYM and QMR_SYM(B) are competitive in comparison to the shifted COCG method. In particular, QMR_SYM(B) can be the method of choice for solving complex symmetric shifted linear systems with a large number of shifts, that arise from large-scale electronic structure theory. In future work, numerical tests for general complex symmetric shifted linear systems will be done to investigate the performance of the method.

Acknowledgements. We wish to express our gratitude to Roland W. Freund (UC Davis) for his fruitful comments during the conference at Harrachov in 2007. We are grateful to an anonymous referee for useful comments that substantially enhanced the quality of this paper. Finally, we would like to thank Martin H. Gutknecht (ETH Zurich) for his careful reading of the manuscript.

REFERENCES

[1] V. FABER AND T. MANTEUFFEL, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.
[2] R. W. FREUND, *On conjugate gradient type methods and polynomial preconditioners for a class of complex non-Hermitian matrices*, Numer. Math., 57 (1990), pp. 285–312.
[3] R. W. FREUND, *Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 425–448.
[4] R. W. FREUND, *Solution of shifted linear systems by quasi-minimal residual iterations*, Numerical Linear Algebra, L. Reichel, A. Ruttan, and R. S. Varga eds., W. de Gruyter, (1993), pp. 101–121.
[5] R. W. FREUND AND N. M. NACHTIGAL, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.
[6] A. FROMMER, *BiCGStab(ℓ) for families of shifted linear systems*, Computing 70 (2003), pp. 87–109.
[7] A. FROMMER AND U. GRÄSSNER, *Restarted GMRES for shifted linear systems*, SIAM J. Sci. Comput. 19 (1998), pp. 15–26.

[8] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore and London, 1996.

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

[10] V. SIMONCINI, *Restarted full orthogonalization method for shifted linear systems*, BIT, 43 (2003), pp. 459–466.

[11] V. SIMONCINI AND D.B. SZYLD, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59.

[12] T. SAKURAI AND H. SUGIURA, *A projection method for generalized eigenvalue problems using numerical integration*, J. Comput. Appl. Math., 159 (2003), pp. 119–128.

[13] T. SOGABE AND S.-L. ZHANG, *A COCR method for solving complex symmetric linear systems*, J. Comput. Appl. Math., 199 (2007), pp. 297–303.

[14] T. SOGABE, T. HOSHI, S.-L. ZHANG, AND T. FUJIWARA, *A numerical method for calculating the Green's function arising from electronic structure theory*, in: Frontiers of Computational Science, eds. Y. Kaneda, H. Kawamura and M. Sasai, Springer-Verlag, Berlin/Heidelberg, 2007, pp. 189–195.

[15] R. TAKAYAMA, T. HOSHI, T. SOGABE, S.-L. ZHANG, AND T. FUJIWARA, *Linear algebraic calculation of Green's function for large-scale electronic structure theory*, Phys. Rev. B 73, 165108 (2006), pp. 1–9.

[16] 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. Mag. 26 (1990), pp. 706–708.