数理解析研究所講究録
第1614巻 2008年 124-135

On the use of the QMR_SYM method for solving complex symmetric shifted linear systems

名古屋大学 大学院工学研究科 曽我部知広 (Tomohiro Sogabe)
Graduate school of engineering, Nagoya University
名古屋大学 大学院工学研究科 張紹良 (Shao-Liang Zhang)
Graduate school of engineering, Nagoya University

Abstract

We consider the solution of complex symmetric shifted linear systems. Such systems arise in large scale electronic structure theory and there is a strong need for the fast solution of the systems. Since the QMR_SYM method is known as a powerful solver for complex symmetric linear systems, we use the idea of the QMR_SYM method together with shift-invariant property of the Krylov subspace for solving complex symmetric shifted linear systems.

1 Introduction

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

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

where $A(\sigma_\ell):=A+\sigma_\ell I$ are nonsingular N-by-N complex symmetric sparse matrices, i.e., $A(\sigma_\ell)=A(\sigma_\ell)^{\mathrm{T}}\neq\bar{A}(\sigma_\ell)^{\mathrm{T}}$, with scalar shifts $\sigma_\ell\in\mathbf{C}$, I is the N-by-N identity matrix, and $\boldsymbol{x}^{(\ell)},\boldsymbol{b}$ are complex vectors of length N. The above systems arise in large-scale electronic structure theory [14] and there is a strong need for the fast solution of the systems.

Since the given shifted linear systems (1) are a set of sparse linear systems, 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 is applying (preconditioned) Krylov subspace methods for solving complex symmetric linear systems such as the COCG method [15], the COCR method [12], and the QMR_SYM method [2] to all of the shifted linear systems (1). On the other hand, denoting n-dimensional Krylov subspace with respect to A and $\boldsymbol{b}$ as $K_n(A,\boldsymbol{b}):=\mathrm{span}\{\boldsymbol{b},A\boldsymbol{b},\ldots,A^{n-1}\boldsymbol{b}\}$, we observe that

$$K_n(A,\boldsymbol{b})=K_n(A(\sigma_\ell),\boldsymbol{b}). \tag{2}$$

This implies that once Krylov basis vectors are generated from one of 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 computational costs involving the basis generation, e.g., matrix-vector multiplications, are saved. Here we give a concrete example: if we apply the COCG method to all of the linear systems (1), then $K_n(A(\sigma_\ell),\boldsymbol{b})$ for $\ell=1,2,\ldots,m$ are generated. On the other hand, if we apply the COCG method to one of the shifted linear systems (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 (2), the shifted conjugate orthogonal conjugate gradient (shifted COCG) method has been recently proposed [14]. The shifted COCG method works well for the problems from electronic structure theory. However, the shifted COCG method requires the choice of a seed system, the term "seed system" was mentioned in the previous paragraph, and unsuitable choice may lead to the drawback that many shifted linear systems remain unsolved. To avoid the drawback, more recently, the seed switching technique has been proposed [13]. For some problems from electronic structure theory, it has been shown that the shifted COCG method together with the seed switching technique is practical.

There is another approach to solving the shifted linear systems (1). That is the use of Krylov subspace methods for non-Hermitian shifted linear systems such as the shifted BiCGStab(ℓ) method [5], the shifted (TF)QMR method [3], the restarted shifted FOM method [9], and the restarted shifted GMRES method [6], see also, e.g., [10]. We readily see that the relation (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, these computational costs can be more expensive than that of the shifted COCG method.

The shifted COCG method is obtained from the COCG method and the COCG method is closely related to the QMR_SYM method, see [2, Prop. 3.3]. Hence, it is natural to consider algorithms using the QMR_SYM method for solving complex symmetric shifted linear systems. The main purpose of the present paper is to develop variants of the QMR_SYM method by considering the minimization of weighted quasi-residual norms for solving complex symmetric shifted linear systems. Of many possible choices of weight matrices for the norms, we will show that there exists a practical weight when the number m of shifted linear systems are large enough.

The present paper is organized as follows: in the next section, an algorithm (referred to as shifted QMR_SYM) for solving the systems (1) will be derived from two important results given by Freund [2, 3], and some properties of the shifted QMR_SYM method are given for the problem from large-scale electronic structure theory. In section 3, some results of a numerical example from electronic structure theory are shown to see the performance of the shifted QMR_SYM approach. Finally, some concluding remarks are made in section 4.

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

2 A formulation of the QMR_SYM method for solving complex symmetric shifted linear systems

The QMR method for shifted linear systems introduced in section 1 was first formulated in [3] for the case of a general non-Hermitian matrix. Therefore, by simplifying the non-Hermitian Lanczos process [8], as is known from other papers such as [2, 15], a simplified QMR method, shifted QMR_SYM, is readily obtained for the case of a complex symmetric matrix. Although the derivation of the shifted QMR_SYM method is straightforward from the viewpoint of the above simplification, in this section we precisely derive the shifted QMR_SYM method from a different viewpoint: a combination of the complex symmetric Lanczos process and the QMR_SYM method with a weighted quasi-residual approach.

First of all let us recall the complex symmetric Lanczos process (see, e.g., Algorithm 2.1 in [2]). The algorithm is given below.

Algorithm 2.1. The complex symmetric Lanczos process

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

Algorithm 2.1 can be also written in matrix form. Let $T_{n+1,n}$ and T_n be the $(n+1) \times n$ and $n \times n$ tridiagonal matrices whose entries are recurrence coefficients of the complex symmetric Lanczos process, which are given by

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

and let V_n be the $N \times n$ matrix with the complex symmetric Lanczos vectors as columns, i.e., $V_n := (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n)$. Then from Algorithm 2.1, we have

$$AV_n = V_{n+1}T_{n+1,n} = V_nT_n + \beta_n \boldsymbol{v}_{n+1}\boldsymbol{e}_n^{\mathrm{T}}, \tag{3}$$

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

Now we are ready for describing an algorithm using the QMR_SYM method for solving complex symmetric shifted linear systems. Let $\boldsymbol{x}_n^{(\ell)}$ be approximate solutions at nth iteration step for the systems (1), which are given by

$$\boldsymbol{x}_n^{(\ell)} = V_n \boldsymbol{y}_n^{(\ell)}, \quad \ell = 1, 2, \ldots, m, \tag{4}$$

where $\boldsymbol{y}_n^{(\ell)}$'s are vectors of length n. Then, from the definition of residual vectors $\boldsymbol{r}_n^{(\ell)} := \boldsymbol{b} - (A + \sigma_\ell I)\boldsymbol{x}_n^{(\ell)}$, update formulas (4), and the matrix form of the complex symmetric Lanczos process (3) we readily obtain

$$\boldsymbol{r}_n^{(\ell)} = V_{n+1}\left(g_1\boldsymbol{e}_1 - T_{n+1,n}^{(\ell)}\boldsymbol{y}_n^{(\ell)}\right), \quad \text{where } T_{n+1,n}^{(\ell)} := T_{n+1,n} + \sigma_\ell \begin{pmatrix} I_n \\ \boldsymbol{0}^{\mathrm{T}} \end{pmatrix}. \tag{5}$$

Here, $\boldsymbol{e}_1$ is the first unit vector written by $\boldsymbol{e}_1=(1,0,\ldots,0)^{\mathrm{T}}$ 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, such choices for $\boldsymbol{y}_n^{(\ell)}$ are impractical due to large amount of computational costs. Hence, an alternative approach is given, i.e., $\boldsymbol{y}_n^{(\ell)}$'s are determined by solving the following weighted least squares problems:

$$\boldsymbol{y}_n^{(\ell)} = \arg\min_{\boldsymbol{z}_n^{(\ell)}\in\mathbf{C}^n} \left\|g_1\boldsymbol{e}_1 - T_{n+1,n}^{(\ell)}\boldsymbol{z}_n^{(\ell)}\right\|_{W_{n+1}^{\mathrm{H}}W_{n+1}}, \tag{6}$$

where W_{n+1} is an $(n+1)$-by-$(n+1)$ nonsingular matrix. Thus $W_{n+1}^{\mathrm{H}}W_{n+1}$ 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

$$\boldsymbol{y}_n^{(\ell)} = \arg\min_{\boldsymbol{z}_n^{(\ell)}\in\mathbf{C}^n} \left\|g_1\boldsymbol{e}_1 - T_{n+1,n}^{(\ell)}\boldsymbol{z}_n^{(\ell)}\right\|_2. \tag{7}$$

A more slightly generalized choice 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, we have

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

Of various possible choices for ω_i, a natural choice is $\omega_i = \|\boldsymbol{v}_i\|_2$ since Ω_{n+1} has the diagonal entries of the upper triangular matrix R_{n+1} that is obtained by 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 (5) and (6) we have

$$\begin{aligned}
\min_{\boldsymbol{z}_n^{(\ell)}\in\mathbf{C}^n} \left\|g_1\boldsymbol{e}_1 - T_{n+1,n}^{(\ell)}\boldsymbol{z}_n^{(\ell)}\right\|_{R_{n+1}^{\mathrm{H}}R_{n+1}} &= \min_{\boldsymbol{z}_n^{(\ell)}\in\mathbf{C}^n} \left\|g_1R_{n+1}\boldsymbol{e}_1 - R_{n+1}T_{n+1,n}^{(\ell)}\boldsymbol{z}_n^{(\ell)}\right\|_2 \\
&= \min_{\boldsymbol{z}_n^{(\ell)}\in\mathbf{C}^n} \left\|Q_{n+1}R_{n+1}(g_1\boldsymbol{e}_1 - T_{n+1,n}^{(\ell)}\boldsymbol{z}_n^{(\ell)})\right\|_2 \\
&= \min_{\boldsymbol{z}_n^{(\ell)}\in\mathbf{C}^n} \left\|V_{n+1}(g_1\boldsymbol{e}_1 - T_{n+1,n}^{(\ell)}\boldsymbol{z}_n^{(\ell)})\right\|_2 \\
&= \min_{\boldsymbol{z}_n^{(\ell)}\in\mathbf{C}^n} \|\boldsymbol{r}_n^{(\ell)}\|_2.
\end{aligned}$$

By solving the above weighted least squares problems, all residual 2-norms are minimized. Hence $W_{n+1} = \Omega_{n+1}$ is a rational choice.

Now, for simplicity we consider the case $W_{n+1} = I_{n+1}$ and derive practical computational formulas for updating approximate solutions $\boldsymbol{x}_n^{(\ell)}$. The derivation is similar to that of the QMR_SYM method. If we find $(n+1)$-by-$(n+1)$ unitary matrices $Q_{n+1}^{(\ell)}$ such that

$$Q_{n+1}^{(\ell)}T_{n+1,n}^{(\ell)} = \begin{pmatrix} \tilde{R}_n^{(\ell)} \\ \boldsymbol{0}^{\mathrm{T}} \end{pmatrix}, \tag{8}$$

where $\tilde{R}_n^{(\ell)}$ are n-by-n banded upper triangular matrices of the form

$$\tilde{R}_n^{(\ell)} := \begin{pmatrix} t_{1,1}^{(\ell)} & t_{1,2}^{(\ell)} & t_{1,3}^{(\ell)} & & \\ & t_{2,2}^{(\ell)} & t_{2,3}^{(\ell)} & \ddots & \\ & & t_{3,3}^{(\ell)} & \ddots & t_{n-2,n}^{(\ell)} \\ & & & \ddots & t_{n-1,n}^{(\ell)} \\ & & & & t_{n,n}^{(\ell)} \end{pmatrix},$$

then it follows from (7) and (8) that we have

$$\min_{\boldsymbol{z}_n^{(\ell)} \in \mathbf{C}^n} \left\| \begin{pmatrix} \boldsymbol{g}_n^{(\ell)} \\ g_{n+1}^{(\ell)} \end{pmatrix} - \begin{pmatrix} \tilde{R}_n^{(\ell)} \\ \mathbf{0}^{\mathrm{T}} \end{pmatrix} \boldsymbol{z}_n^{(\ell)} \right\|_2, \text{ where } \begin{pmatrix} \boldsymbol{g}_n^{(\ell)} \\ g_{n+1}^{(\ell)} \end{pmatrix} := g_1 Q_{n+1}^{(\ell)} \boldsymbol{e}_1. \tag{9}$$

By solving the above least squares problems we have $\boldsymbol{y}_n^{(\ell)} = (\tilde{R}_n^{(\ell)})^{-1} \boldsymbol{g}_n^{(\ell)}$. Substituting this results into (4) and using the auxiliary vectors

$$(\tilde{\boldsymbol{p}}_1 \ \tilde{\boldsymbol{p}}_2 \ \cdots \ \tilde{\boldsymbol{p}}_n) := V_n (\tilde{R}_n^{(\ell)})^{-1},$$

we obtain the following update formulas:

$$\tilde{\boldsymbol{p}}_n^{(\ell)} = (\boldsymbol{v}_n - t_{n-2,n}^{(\ell)} \tilde{\boldsymbol{p}}_{n-2}^{(\ell)} - t_{n-1,n}^{(\ell)} \tilde{\boldsymbol{p}}_{n-1}^{(\ell)}) / t_{n,n}^{(\ell)}, \tag{10}$$

$$\boldsymbol{x}_n^{(\ell)} = \boldsymbol{x}_{n-1}^{(\ell)} + g_n^{(\ell)} \tilde{\boldsymbol{p}}_n^{(\ell)}. \tag{11}$$

Here, we note that $g_n^{(\ell)}$ is the nth entry of the vector $\boldsymbol{g}_n^{(\ell)}$. Using auxiliary vectors $\boldsymbol{p}_i^{(\ell)} = t_{i,i}^{(\ell)} \tilde{\boldsymbol{p}}_i^{(\ell)}$, the computational costs for the above recurrences are reduced by the following simple rewrite:

$$\boldsymbol{p}_n^{(\ell)} = \boldsymbol{v}_n - (t_{n-2,n}^{(\ell)} / t_{n-2,n-2}^{(\ell)}) \boldsymbol{p}_{n-2}^{(\ell)} - (t_{n-1,n}^{(\ell)} / t_{n-1,n-1}^{(\ell)}) \boldsymbol{p}_{n-1}^{(\ell)}, \tag{12}$$

$$\boldsymbol{x}_n^{(\ell)} = \boldsymbol{x}_{n-1}^{(\ell)} + (g_n^{(\ell)} / t_{n,n}^{(\ell)}) \boldsymbol{p}_n^{(\ell)}. \tag{13}$$

The new recurrences (12)-(13) require $6Nm + 3m$ operations per iteration step. Since the previous recurrences (10)-(11) require $7Nm$ operations, this rewrite is useful in practice when the number of linear systems is very large, say, $m \gg 1$.

We have obtained computational formulas for approximate solutions $\boldsymbol{x}_n^{(\ell)}$. Next, we describe how to obtain $Q_{n+1}^{(\ell)}$ of the form (8). Givens rotations, see, e.g., [7, p.215], are powerful tools to answer it, which are defined by

$$G_n^{(\ell)}(i) := \begin{pmatrix} I_{i-1} & & & \\ & c_i^{(\ell)} & s_i^{(\ell)} & \\ & -\bar{s}_i^{(\ell)} & c_i^{(\ell)} & \\ & & & I_{n-i-1} \end{pmatrix}, \ c_i^{(\ell)} \in \mathbf{R}, \ s_i^{(\ell)} \in \mathbf{C}, \ (c_i^{(\ell)})^2 + |s_i^{(\ell)}|^2 = 1.$$

By determining $c_i^{(\ell)}$ and $s_i^{(\ell)}$ such that the $(i+1, i)$ entry of a matrix $G_n^{(\ell)}(i)T$ is zero, where T is a tridiagonal matrix, we readily have the form (8) in the following way:

$$G_{n+1}^{(\ell)}(n) G_{n+1}^{(\ell)}(n-1) \cdots G_{n+1}^{(\ell)}(1) T_{n+1,n}^{(\ell)} = \begin{pmatrix} \tilde{R}_n^{(\ell)} \\ \mathbf{0}^{\mathrm{T}} \end{pmatrix}. \tag{14}$$

From the above we see that $G_{n+1}^{(\ell)}(n)G_{n+1}^{(\ell)}(n-1)\cdots G_{n+1}^{(\ell)}(1)$ is the matrix $Q_{n+1}^{(\ell)}$. Here we note that $Q_{n+1}^{(\ell)}$ and $Q_n^{(\ell)}$ are related by

$$Q_{n+1}^{(\ell)} = G_{n+1}^{(\ell)}(n)\begin{pmatrix} Q_n^{(\ell)} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{pmatrix} \text{ for } n = 2, 3, \ldots, \tag{15}$$

where $Q_2^{(\ell)} = G_2^{(\ell)}(1)$. Now we describe the complete algorithm of shifted QMR_SYM method.

Algorithm 2.2. Shifted QMR_SYM

$\boldsymbol{x}_0^{(\ell)} = \boldsymbol{p}_{-1}^{(\ell)} = \boldsymbol{p}_0^{(\ell)} = \mathbf{0}$, $\boldsymbol{v}_1 = \boldsymbol{b}/(\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b})^{1/2}$, $g_1^{(\ell)} = (\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b})^{1/2}$,
for $n = 1, 2, \ldots$ **do:**
(The complex symmetric Lanczos process)
$\alpha_n = \boldsymbol{v}_n^{\mathrm{T}} A \boldsymbol{v}_n$,
$\tilde{\boldsymbol{v}}_{n+1} = A\boldsymbol{v}_n - \alpha_n \boldsymbol{v}_n - \beta_{n-1}\boldsymbol{v}_{n-1}$,
$\beta_n = (\tilde{\boldsymbol{v}}_{n+1}^{\mathrm{T}}\tilde{\boldsymbol{v}}_{n+1})^{1/2}$,
$\boldsymbol{v}_{n+1} = \tilde{\boldsymbol{v}}_{n+1}/\beta_n$,
$t_{n-1,n}^{(\ell)} = \beta_{n-1}$, $t_{n,n}^{(\ell)} = \alpha_n + \sigma_\ell$, $t_{n+1,n}^{(\ell)} = \beta_n$,
(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:**

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

end

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

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

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

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

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

end

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

Proposition 1 (See [4]) *The nth residual 2-norms of the approximate solutions* $\boldsymbol{x}_n^{(\ell)}$ *for the shifted QMR_SYM method are given by*

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

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 1 is a result known to hold for the QMR method in [4]. 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 (See [1]) *Let* $A \in \mathbf{R}^{N\times N}$ *be real symmetric,* $\sigma_\ell \in \mathbf{C}$ *be complex shifts, and* $\boldsymbol{b} \in \mathbf{R}^N$. *Then the shifted QMR_SYM method (Algorithm 2.1) enjoys the following properties:*

I. All matrix-vector multiplications can be done in real arithmetic;

II. An approximate solution at nth iteration step for each ℓ *has minimal residual 2-norms, i.e.,* $\boldsymbol{x}_n^{(\ell)}$*'s are generated such that* $\min \|\boldsymbol{r}_n^{(\ell)}\|_2$ *over* $\boldsymbol{x}_n^{(\ell)} \in K_n(A, \boldsymbol{b})$;

III. $\|\boldsymbol{r}_n^{(\ell)}\|_2 = |g_{n+1}^{(\ell)}|$ *for* $\ell = 1, 2, \ldots, m,\ n \geq 0$.

The above properties are known results since the properties have been proved for each individual shift. See [1] for detail.

The properties of proposition 2 may be very useful for large-scale electronic structure theory [14] and a projection approach for eigenvalue problems [11] since there are complex symmetric shifted linear systems to be solved efficiently under the assumption of proposition 2.

3 A numerical example

In this section, we report some results of a numerical example for the shifted COCG method and the shifted QMR_SYM method (Algorithm 2.2). We evaluate both two methods in terms of computational 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 $\epsilon = 10^{-12}$.

We consider the solutions of the following shifted linear systems that come from electronic structure calculation of a bulk Si(001) with 512 atoms in [14]:

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

where $\sigma_\ell = 0.400 + (\ell - 1 + i)/1000$, $H \in R^{2048\times 2048}$ is a symmetric matrix with 139264 entries, $\boldsymbol{e}_1 = (1, 0, \ldots, 0)^T$, 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 will remain unsolved by another choice.

Figure 1 shows the number of iterations of each method to solve ℓth shifted linear systems. For example, in Fig. 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 600th shifted linear systems, i.e., $(\sigma_{600}I - H)\boldsymbol{x}^{(600)} = \boldsymbol{e}_1$.

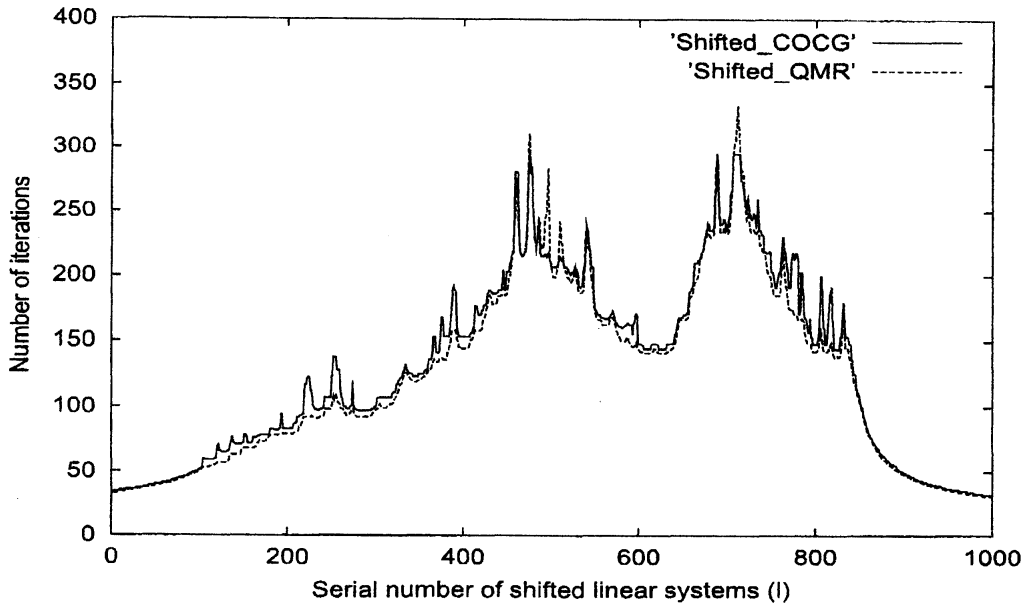

Figure 1: Number of iterations for the shifted COCG method and the shifted QMR_SYM method versus serial number of shifted linear systems.

From Fig. 1 we obtain three observations: first, the two 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 shifted COCG method. This phenomenon is closely related to proposition 2, which will be clearer later; third, for each method the required number of iterations depends highly on the shift parameters σ_ℓ. This result may come from varying eigenvalues of the coefficient matrices $\sigma_\ell I - H$ since if we choose σ_ℓ close to λ, where λ is one of the eigenvalues of H, then $\sigma_\ell I - H$ is close to

singular. Conversely, from the shape of Fig. 1 we may obtain the partial distribution of eigenvalues of H.

One of the residual 2-norm histories for the two methods is given in Fig. 2. From Fig. 2 we see that Log_{10} of the relative residual 2-norm of the shifted QMR_SYM method decreases monotonically and at every iteration step the norm is less than that of the shifted COCG method. Hence we can say that the property (II) of proposition 2 was experimentally supported by this history.

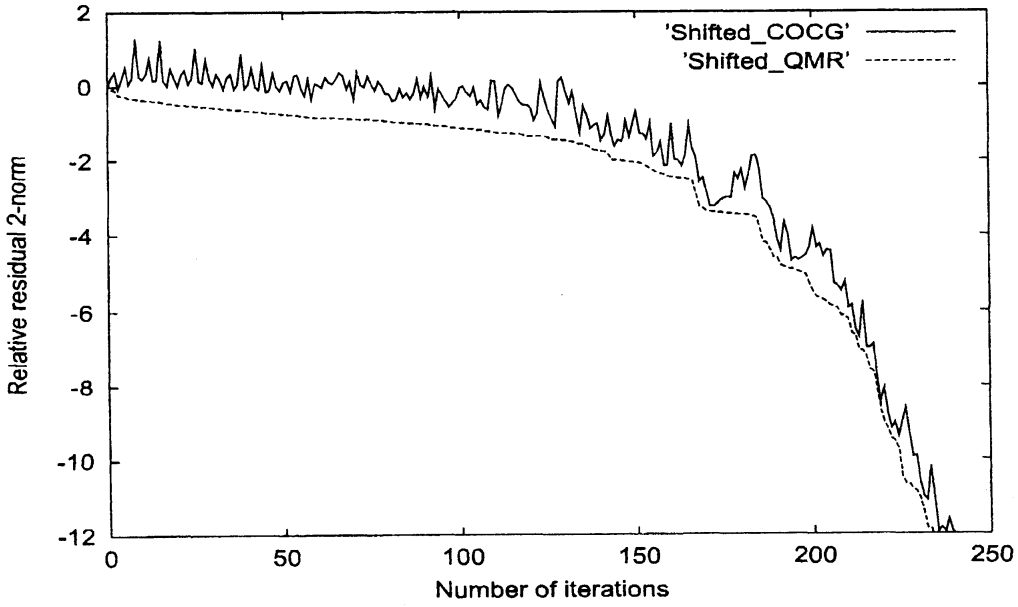

Figure 2: Log_{10} of relative residual 2-norms versus the number of iterations of the shifted COCG method and the shifted QMR_SYM method for shifted linear systems with $\ell = 701, i.e., \sigma_{701} = 1.100 + 0.001i$.

Each computational time of the two methods is given in Fig. 3, where the horizontal axis denotes the number of shifted linear systems that are solved from $\ell = 1$ to m. For example, in Fig. 3, the computational time of the shifted COCG method at $m = 200$ is about 0.76 [sec.], which means that it required 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 Fig. 3 we see that as the number m grows larger, the shifted QMR method required the CPU time more than the shifted COCG method.

In Fig. 3 we can know little about the properties of the two methods for small ℓ. Hence, we show the ratio of the CPU time of the shifted QMR_SYM method to that of the shifted COCG method in Fig. 4. We see from Fig. 4 that in terms of ratio of CPU time, the shifted QMR_SYM method converged much faster than the shifted COCG method when the number of shifted linear systems is small, say, $m < 200$. This can be explained in the following way: for small m, updating approximate solutions does not affect the CPU time so much. Other operations such as matrix-vector multiplications are now one of the most time-consuming parts since the two methods required almost

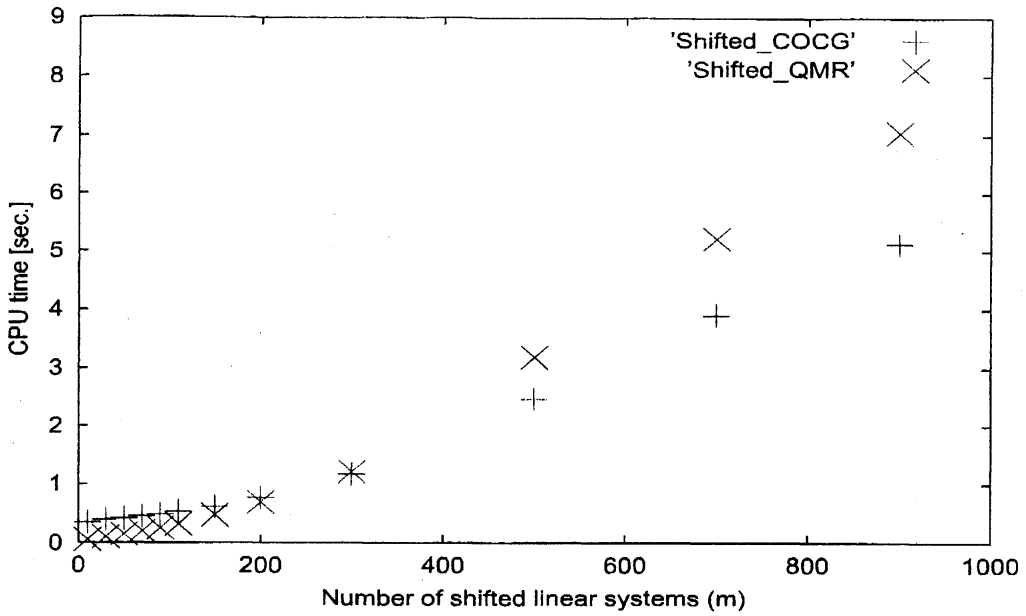

Figure 3: Required CPU time given in seconds versus the number of shifted linear systems for each iterative method.

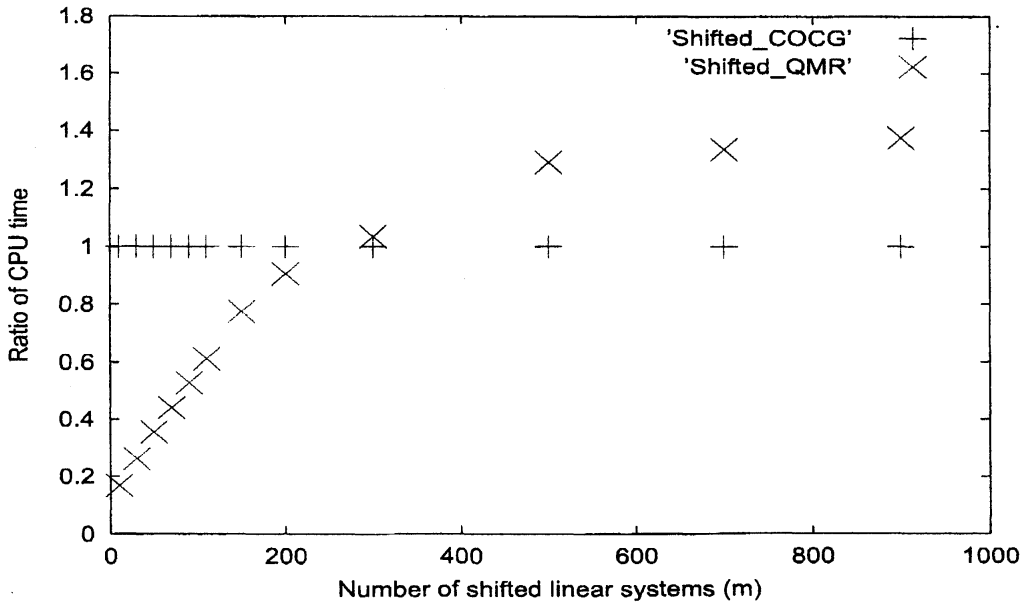

Figure 4: The ratio of the CPU time of the shifted QMR_SYM method to that of the shifted COCG method versus the number of shifted linear systems.

the same number of iterations, see Fig. 1. From proposition 2 (I) we know that in this case the cost of matrix-vector multiplication for the shifted QMR_SYM method is much cheaper than that for the shifted COCG method since the shifted QMR_SYM method require real matrix-real vector multiplications and 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 can be done in real arithmetic. Hence, the shifted QMR_SYM method converged much faster than the shifted COCG method for a small number of shifts.

4 Concluding remarks

In this paper, with the aim of solving complex symmetric shifted linear systems efficiently, we have derived the shifted QMR_SYM method from two important results given in [2, 3]. The method has an advantage over the shifted COCG method in that it has no need to choose a suitable seed system. From the results of a numerical example, we have learned that for a small number of shifts the shifted QMR_SYM method converged much faster than the shifted COCG method. In this case, the shifted QMR_SYM method is a method of choice for solving complex symmetric shifted linear systems arising from electronic structure theory.

Acknowledgments

We would like to thank Prof. Takeo Fujiwara (The University of Tokyo) and Prof. Takeo Hoshi (Tottori University) for providing us a test problem used in the numerical experiment.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] A. FROMMER, *BiCGStab(ℓ) for families of shifted linear systems*, Computing 70 (2003), pp. 87-109.

[6] A. FROMMER AND U. GRÄSSNER, *Restarted GMRES for shifted linear systems*, SIAM J. Sci. Comput. 19 (1998), pp. 15-26.

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

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

[9] V. SIMONCINI, *Restarted full orthogonalization method for shifted linear systems*, BIT Numerical Mathematics, 43 (2003), pp. 459-466.

[10] V. SIMONCINI AND D.B. SZYLD, *Recent computational developments in Krylov subspace methods for linear systems*, Num. Lin. Alg. with Appl., 14 (2007), pp. 1-59.

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

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

[13] 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.

[14] 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.

[15] H. A. VAN DER VORST AND J. B. M. MELISSEN, *A Petrov-Galerkin type method for solving $Ax = b$, where A is symmetric complex*, IEEE Trans. Mag. 26(2) (1990), pp. 706-708.