A fast numerical method for generalized shifted linear systems with complex symmetric matrices
(複素対称行列を係数に持つ一般化シフト線形方程式の高速数値解法)

愛知県立大学　曽我部知広 (Tomohiro Sogabe)
Aichi Prefectural University
鳥取大学　星健夫 (Takeo Hoshi)
Tottori University
名古屋大学　張紹良 (Shao-Liang Zhang)
Nagoya University
東京大学　藤原毅夫 (Takeo Fujiwara)
The University of Tokyo

Abstract

We consider the solution of generalized shifted linear systems with complex symmetric matrices. We present a numerical method for solving the linear systems based on the shifted COCG method with a suitable bilinear form. The method can be attractive when the *inner* linear systems can efficiently be solved.

1 Introduction

We consider the solution of generalized shifted linear systems with complex symmetric (non-Hermitian) matrices of the form:

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

where the coefficient matrices $L(\sigma_\ell):=A+\sigma_\ell B$ are nonsingular $N\times N$ complex symmetric sparse matrices, i.e., $L(\sigma_\ell)=L(\sigma_\ell)^{\mathrm{T}}\neq\overline{L}(\sigma_\ell)^{\mathrm{T}}$, with scalar shifts $\sigma_\ell\in\mathbf{C}$, $I$ is the $N\times N$ identity matrix, and $\boldsymbol{x}^{(\ell)},\boldsymbol{b}$ are complex vectors of length $N$. Matrices $A$ and $B$ are assumed to be nonsingular. The linear systems (1.1) arise in large scale electronic structure theory, and there is a strong need for fast solution of the linear systems.

When the coefficient matrices $L(\sigma_\ell)$ of the linear systems (1.1) are real symmetric matrices, the linear systems are known as parametrized (real) symmetric linear systems. An efficient numerical method for solving the linear systems has been proposed by Meerbergen [6]. For studies on other parametrized linear systems, see, e.g., an excellent survey by Simoncini & Szyld [8].

When the matrix $B$ is the identity matrix, the linear systems (1.1) reduce to standard shifted linear systems with complex symmetric matrices. Efficient numerical methods for this class of shifted linear systems have been studied based on Krylov subspace methods such as the shifted QMR method [3], the shifted WQMR method [9], and the shifted COCG method [10].

In this paper, we extend the shifted COCG method in order to solve the generalized shifted linear systems (1.1). The extension is based on (i) transforming the systems (1.1) into standard shifted linear systems and (ii) using a suitable bilinear form in order to make use of properties of the matrices $L(\sigma_\ell)$.

This paper is organized as follows. The algorithm and a property of the shifted COCG method for solving shifted linear systems with complex symmetric matrices are described in the next section. Then, in Section 3, an extension of the shifted COCG method is proposed for solving the generalized shifted linear systems. Results on some numerical experiments are reported in Section 4. Finally, some concluding remarks are made in Section 5.

2 The shifted COCG method

The shifted COCG method [10] is a powerful solver for shifted linear systems with complex symmetric matrices of the form:

$$(A+\sigma_\ell I)\boldsymbol{x}^{(\ell)} = \boldsymbol{b}, \quad \ell = 1, 2, \dots, m. \tag{2.1}$$

Let $\boldsymbol{x}_0^{(\ell)}$ be the initial approximate solutions for the linear systems (2.1) so that the corresponding initial residual vectors $\boldsymbol{r}_0^{(\ell)} := \boldsymbol{b} - (A+\sigma_\ell I)\boldsymbol{x}_0^{(\ell)}$, $\ell = 1, 2, \dots, m$ are collinear, i.e., there exist $c_2, c_3, \dots, c_m \in \mathbf{C}$ such that

$$\boldsymbol{r}_0^{(1)} = c_2 \boldsymbol{r}_0^{(2)} = \cdots = c_m \boldsymbol{r}_0^{(m)}.$$

Then, the shifted COCG method finds approximate solutions of (2.1) over the following affine space:

$$\boldsymbol{x}_n^{(\ell)} \in \boldsymbol{x}_0^{(\ell)} + K_n(A+\sigma_\ell I, \boldsymbol{r}_0^{(\ell)}) \tag{2.2}$$

so that the nth residual vector $\boldsymbol{r}_n^{(\ell)} := \boldsymbol{b} - (A+\sigma_\ell I)\boldsymbol{x}_n^{(\ell)}$ satisfies

$$\boldsymbol{r}_n^{(\ell)} \in K_{n+1}(A+\sigma_\ell I, \boldsymbol{r}_0^{(\ell)}) \perp K_n(\overline{A+\sigma_\ell I}, \overline{\boldsymbol{r}}_0^{(\ell)}). \tag{2.3}$$

Here $K_n(A, \boldsymbol{b})$ is n-dimensional Krylov subspace given by $\mathrm{span}\{\boldsymbol{b}, A\boldsymbol{b}, \dots, A^{n-1}\boldsymbol{b}\}$. Since $\boldsymbol{r}_0^{(\ell)}$'s are collinear, it is widely known that shift-invariance property of Krylov subspaces holds, i.e.,

$$\begin{aligned} K_n(A+\sigma_i I, \boldsymbol{r}_0^{(i)}) &= K_n(A+\sigma_j I, \boldsymbol{r}_0^{(j)}), \\ K_n(\overline{A+\sigma_i I}, \overline{\boldsymbol{r}}_0^{(i)}) &= K_n(\overline{A+\sigma_j I}, \overline{\boldsymbol{r}}_0^{(j)}) \end{aligned}$$

for all $i, j = 1, 2, \dots, m$. This means that it is enough to generate only one Krylov subspace, leading to cost efficient algorithm. The algorithm of the shifted COCG method, for simplicity $\boldsymbol{x}_0^{(\ell)} = \boldsymbol{0}$, is described as follows:

Algorithm 1: Shifted COCG

```
$\boldsymbol{x}_0^{(\ell)} = \boldsymbol{p}_{-1}^{(\ell)} = \boldsymbol{0},\ \boldsymbol{r}_0 = \boldsymbol{b},$
$\beta_{-1} = 0,\ \pi_0^{(s,\ell)} = \pi_{-1}^{(s,\ell)} = \alpha_{-1} = 1,$
for $n = 0, 1, \ldots$ until $\|\boldsymbol{r}_n\|_2 \le \epsilon\|\boldsymbol{b}\|_2$ do:
  $\boldsymbol{p}_n = \boldsymbol{r}_n + \beta_{n-1}\boldsymbol{p}_{n-1},$
  $\alpha_n = \frac{(\boldsymbol{r}_n, \boldsymbol{r}_n)}{(\boldsymbol{p}_n, (A + \sigma_s I)\boldsymbol{p}_n)},$
  $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha_n \boldsymbol{p}_n,$
  (begin shifted system)
   for $\ell(\neq s) = 1, \ldots, m$
     if $\|\boldsymbol{r}_n^{(\ell)}\|_2 > \epsilon\|\boldsymbol{b}\|_2$ then,
       $\pi_{n+1}^{(s,\ell)} = R_{n+1}^{(s)}(\sigma_s - \sigma_\ell) \longleftarrow (2.4)$
       $\beta_{n-1}^{(\ell)} = \left(\frac{\pi_{n-1}^{(s,\ell)}}{\pi_n^{(s,\ell)}}\right)^2 \beta_{n-1},$
       $\alpha_n^{(\ell)} = \frac{\pi_n^{(s,\ell)}}{\pi_{n+1}^{(s,\ell)}}\alpha_n,$
       $\boldsymbol{p}_n^{(\ell)} = \frac{1}{\pi_n^{(s,\ell)}}\boldsymbol{r}_n + \beta_{n-1}^{(\ell)}\boldsymbol{p}_{n-1}^{(\ell)},$
       $\boldsymbol{x}_{n+1}^{(\ell)} = \boldsymbol{x}_n^{(\ell)} + \alpha_n^{(\ell)}\boldsymbol{p}_n^{(\ell)},$
     end if
   end
  (end shifted system)
  $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n - \alpha_n(A + \sigma_s I)\boldsymbol{p}_n,$
  $\beta_n = \frac{(\boldsymbol{r}_{n+1}, \boldsymbol{r}_{n+1})}{(\boldsymbol{r}_n, \boldsymbol{r}_n)}.$
end
```

Here the symbol $(\boldsymbol{x}, \boldsymbol{y})$ be the bilinear form given by $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$, which is different from standard dot product $\boldsymbol{x}^{\mathrm{H}}\boldsymbol{y}$. In Algorithm 1, the computational formula for $\pi_{n+1}^{(s,\ell)}$ is given below.

$$R_n^{(s)}(\lambda) = (1 - \alpha_{n-1}^{(s)}\lambda)R_{n-1}^{(s)}(\lambda) + \frac{\beta_{n-2}^{(s)}}{\alpha_{n-2}^{(s)}}\alpha_{n-1}^{(s)}\left(R_{n-1}^{(s)}(\lambda) - R_{n-2}^{(s)}(\lambda)\right),\ n = 2, 3, \ldots, \quad (2.4)$$

where $R_0^{(s)}(\lambda) = 1,\ R_1^{(s)}(\lambda) = (1 - \alpha_0\lambda)R_0^{(s)}(\lambda)$.

3 An extension of the shifted COCG method to the generalized shifted linear systems

In this section, we extend the shifted COCG method for solving the generalized shifted linear systems (1.1). We first transform the generalized shifted linear systems (1.1) into the following form:

$$(B^{-1}A + \sigma_\ell I)\boldsymbol{x}^{(\ell)} = B^{-1}\boldsymbol{b}, \quad \ell = 1, 2, \dots, m. \tag{3.1}$$

The above linear systems are well-known (standard) shifted linear systems. It is therefore natural to use Krylov subspace methods for non-Hermitian shifted linear systems such as the shifted GMRES method by Datta & Saad [1] and the shifted BiCGStab(ℓ) method by Frommer [4]. For other methods to solve non-Hermitian shifted linear systems, see, e.g., [8]. These methods however does not use the property of $L(\sigma_\ell)$, i,e, complex symmetric matrices. This motivates us to find algorithm that makes use of the property, and we will show that the algorithm has nice properties: short-term recurrence relation and no requirement of restarting.

We now extend the shifted COCG method to the generalized shifted linear systems (1.1). We consider finding approximate solutions over the following affine space:

$$\boldsymbol{x}_n^{(\ell)} \in \boldsymbol{x}_0^{(\ell)} + K_n(B^{-1}A + \sigma_\ell I, B^{-1}\boldsymbol{r}_0^{(\ell)}) \tag{3.2}$$

so that the nth residual vector $\boldsymbol{r}_n^{(\ell)} := \boldsymbol{b} - (B^{-1}A + \sigma_\ell I)\boldsymbol{x}_n^{(\ell)}$ satisfies

$$\boldsymbol{r}_n^{(\ell)} \in K_{n+1}(B^{-1}A + \sigma_\ell I, B^{-1}\boldsymbol{r}_0^{(\ell)}) \perp \overline{B}K_n(\overline{B^{-1}A + \sigma_\ell I}, \overline{B}^{-1}\overline{\boldsymbol{r}}_0^{(\ell)}). \tag{3.3}$$

It is easy to see that the strategy (3.2)-(3.3) is equivalent to the strategy (2.2)-(2.3) of the shifted COCG method when the matrix $B$ is the identity matrix.

This strategy corresponds to applying the shifted COCG method with the following bilinear form:

$$(\boldsymbol{x}, \boldsymbol{y})_B := \boldsymbol{x}^{\mathrm{T}} B \boldsymbol{y}. \tag{3.4}$$

to the shifted linear systems (3.1). The resulting algorithm, referred to as the generalized shifted COCG method, is given next.

Algorithm 2: Generalized shifted COCG

```
$\boldsymbol{x}_0^{(\ell)} = \boldsymbol{p}_{-1}^{(\ell)} = \boldsymbol{0}, \ \boldsymbol{r}_0 = \boldsymbol{b},$
$\beta_{-1} = 0, \ \pi_0^{(s,\ell)} = \pi_{-1}^{(s,\ell)} = \alpha_{-1} = 1,$
for $n = 0, 1, \dots$ until $\|\boldsymbol{r}_n\|_2 \le \epsilon \|\boldsymbol{b}\|_2$ do:
  $\boldsymbol{p}_n = B^{-1}\boldsymbol{r}_n + \beta_{n-1}\boldsymbol{p}_{n-1},$
  $\alpha_n = \dfrac{\boldsymbol{r}_n^{\mathrm{T}} B^{-1} \boldsymbol{r}_n}{\boldsymbol{p}_n^{\mathrm{T}}(A + \sigma_s B)\boldsymbol{p}_n},$
```

```
    $x_{n+1} = x_n + \alpha_n p_n$,
    (begin shifted system)
      For $\ell(\neq s) = 1, \dots, m$
        if $\|r_n^{(\ell)}\|_2 > \epsilon \|b\|_2$ then,
          $\pi_{n+1}^{(s,\ell)} = R_{n+1}^{(s)}(\sigma_s - \sigma_\ell) \longleftarrow (2.4)$
          $\beta_{n-1}^{(\ell)} = \left( \frac{\pi_{n-1}^{(s,\ell)}}{\pi_n^{(s,\ell)}} \right)^2 \beta_{n-1}$,
          $\alpha_n^{(\ell)} = \frac{\pi_n^{(s,\ell)}}{\pi_{n+1}^{(s,\ell)}} \alpha_n$,
          $p_n^{(\ell)} = \frac{1}{\pi_n^{(s,\ell)}} B^{-1} r_n + \beta_{n-1}^{(\ell)} p_{n-1}^{(\ell)}$,
          $x_{n+1}^{(\ell)} = x_n^{(\ell)} + \alpha_n^{(\ell)} p_n^{(\ell)}$,
        end if
      end
    (end shifted system)
    $r_{n+1} = r_n - \alpha_n (A + \sigma_s B) p_n$,
    $\beta_n = \frac{r_{n+1}^{\mathrm{T}} B^{-1} r_{n+1}}{r_n^{\mathrm{T}} B^{-1} r_n}$.
  end
```

It is worth mentioning that Algorithm 2 generates original residual vectors, i.e., $r_n = b - (A + \sigma_s B) x_n$. In Algorithm 2, the computational formula for $\pi_{n+1}^{(s,\ell)}$ is given below.

$$R_n^{(s)}(\lambda) := (1 - \alpha_{n-1}^{(s)} \lambda) R_{n-1}^{(s)}(\lambda) + \frac{\beta_{n-2}^{(s)}}{\alpha_{n-2}^{(s)}} \alpha_{n-1}^{(s)} \left(R_{n-1}^{(s)}(\lambda) - R_{n-2}^{(s)}(\lambda) \right), \tag{3.5}$$

where $R_0^{(s)}(\lambda) = 1$, $R_1^{(s)}(\lambda) = (1 - \alpha_0 \lambda) R_0^{(s)}(\lambda)$. Here we note that in Algorithm 2, we need to compute $B^{-1} r_n$ that is equivalent to finding the solution y_n of the linear systems of the form:

$$B y_n = r_n \tag{3.6}$$

at each iteration step. Since the matrix B is real symmetric positive definite, the systems could be solved by the CG method with a suitable preconditioner.

In what follows, the linear systems (3.6) and the solutions are referred to as *inner* linear systems and *inner* solutions, respectively. The linear systems (1.1) and the solutions x_n are referred to as *outer* linear systems and *outer* solutions, respectively.

On a derivation of the generalized shifted COCG method

It is shown that the generalized shifted COCG method can be derived form the shifted

Bi-CG method [4]. Applying the shifted Bi-CG method to the linear systems (3.1) yields

$$\boldsymbol{x}_n^{(\ell)} \in \boldsymbol{x}_0^{(\ell)} + K_n(B^{-1}A + \sigma_\ell I, B^{-1}\boldsymbol{r}_0^{(\ell)}), \tag{3.7}$$

$$\boldsymbol{r}_n^{(\ell)} \in K_{n+1}(B^{-1}A + \sigma_\ell I, B^{-1}\boldsymbol{r}_0^{(\ell)}) \perp K_n((B^{-1}A + \sigma_\ell I)^{\mathrm{H}}, \boldsymbol{r}_0^{*(\ell)}), \tag{3.8}$$

where $\boldsymbol{r}_0^{*(\ell)}$ are so-called shadow vectors. If we choose $\boldsymbol{r}_0^{*(\ell)} = \overline{\boldsymbol{r}}_0^{(\ell)}$, then it follows that

$$\begin{aligned} K_n((B^{-1}A + \sigma_\ell I)^{\mathrm{H}}, \boldsymbol{r}_0^{*(\ell)}) &= K_n((B^{-1}A + \sigma_\ell I)^{\mathrm{H}}, (\overline{B})(\overline{B})^{-1}\overline{\boldsymbol{r}}_0^{(\ell)}) \\ &= K_n((\overline{AB^{-1} + \sigma_\ell I}), (\overline{B})(\overline{B})^{-1}\overline{\boldsymbol{r}}_0^{(\ell)}) \\ &= K_n(\overline{AB}^{-1}, (\overline{B})(\overline{B})^{-1}\overline{\boldsymbol{r}}_0^{(\ell)}) \\ &= \overline{B}K_n(\overline{B}^{-1}\overline{A}, \overline{B}^{-1}\overline{\boldsymbol{r}}_0^{(\ell)}) \\ &= \overline{B}K_n(\overline{B^{-1}A + \sigma_\ell I}, \overline{B}^{-1}\overline{\boldsymbol{r}}_0^{(\ell)}). \end{aligned}$$

From which, we see that the strategy (3.7)-(3.8) of the shifted Bi-CG method with the choice $\boldsymbol{r}_0^{*(\ell)} = \overline{\boldsymbol{r}}_0^{(\ell)}$ is equivalent to the strategy (3.2)-(3.3) of the generalized shifted COCG method. This result means that the generalized shifted COCG method is a simplification of the shifted Bi-CG method applied to the linear systems (3.1).

This relation essentially goes back to the relation between the Bi-CG method and the COCG method that has elegantly been shown by Freund [2].

A seed switching technique for the generalized shifted COCG method

From the property $\boldsymbol{r}_n^{(s)} = \pi_n^{(s,\ell)}\boldsymbol{r}_n^{(\ell)}$ it follows that $\|\boldsymbol{r}_n^{(\ell)}\| \le \|\boldsymbol{r}_n^{(s)}\|$ if $|\pi_n^{(s,\ell)}| = |R_n^{(s)}(\sigma_s - \sigma_\ell)| \ge 1$. Hence, if one could find a seed system such that $|R_n^{(s)}(\sigma_s - \sigma_\ell)| \ge 1$ for all $\ell$, then all shifted systems could be solved. In practice, it is generally hard to find such linear system in advance because it means to find the system with the slowest convergence behavior. For the generalized shifted COCG method, we have developed the seed switching technique in order to avoid the problem [7]. See also [11] for one of the applications. This technique can also be applied to the generalized shifted COCG method, and the resulting technique is summarized as follows:

1. Choose a seed system, and then start Algorithm 2;
2. If the seed system was solved at nth iteration, then find the new one;
3. Start Algorithm 2 from $(n+1)$th iteration using the new seed system.

In II, as one of criteria for choosing the new seed system $\tilde{s}$, we adopt $\tilde{s} = \arg\max_{i\in S}\{\|\boldsymbol{r}_n^{(\ell)}\|\}$, where $S$ denotes an index set of unsolved systems. In III, we need two steps to switch the old seed system to the new one. First, compute

$$\pi_{n+1}^{(s,\tilde{s})} = R_{n+1}^{(s)}(\sigma_s - \sigma_{\tilde{s}}), \quad \beta_n^{(\tilde{s})} = \left(\pi_n^{(s,\tilde{s})}/\pi_{n+1}^{(s,\tilde{s})}\right)^2\beta_n$$

for obtaining $\boldsymbol{r}_{n+1}^{(\tilde{s})}$ and $\beta_n^{(\tilde{s})}\boldsymbol{p}_n^{(\tilde{s})}$, and then we have $\boldsymbol{p}_{n+1}^{(\tilde{s})}(= B^{-1}\boldsymbol{r}_{n+1}^{(\tilde{s})} + \beta_n^{(\tilde{s})}\boldsymbol{p}_n^{(\tilde{s})})$. Now, we are ready to start the COCG with the blinear form (3.4) for solving the system

$(B^{-1}A + \sigma_{\tilde{s}}I)\boldsymbol{x}^{(\tilde{s})} = B^{-1}\boldsymbol{b}$ from $(n+1)$th iteration step. Second, to solve the rest of systems by using the new seed $\tilde{s}$, it requires generating $\alpha_{n+1}^{(\ell)}$, $\beta_n^{(\ell)}$ from the new seed. They can be readily generated by the following polynomial:

$$\pi_{n+1}^{(\tilde{s},\ell)} = R_{n+1}^{(\tilde{s})}(\sigma_{\tilde{s}} - \sigma_\ell) \quad \text{for all} \quad \ell \in S.$$

To obtain the above polynomial, we need to compute

$$\alpha_i^{(\tilde{s})} = \left(\pi_i^{(s,\tilde{s})}/\pi_{i+1}^{(s,\tilde{s})}\right)\alpha_\ell, \quad \beta_j^{(\tilde{s})} = \left(\pi_j^{(s,\tilde{s})}/\pi_{j+1}^{(s,\tilde{s})}\right)^2\beta_j$$

for $i = 0, \ldots, n$, $j = 0, \ldots, n-1$. Hence, the switching strategy requires only scalar operations, and moreover we see that if breakdown does not occur, iterating the process from (II) to (III) enables us to keep solving the systems without losing the dimension of the Krylov subspace that has been generated until the last switching.

The results of some numerical experiments will be reported in order to show the practical efficiency of the generalized shifted COCG method.

4 Numerical examples

In this section, we report on some numerical examples concerning the generalized shifted COCG method (Algorithm 2). 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 -*O*3. In all cases the stopping criteria were set as $\epsilon = 10^{-12}$. The inner linear systems (3.6) was solved by the CG method without preconditioner that is more efficient than IC(0) in our problems. The CG method was stopped when its relative residual 2-norm becomes less than or equal to the given value ϵ_{CG}, and in our numerical example we used $\epsilon_{\text{CG}} = 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}$.

4.1 Example 1

The first problem comes from the electronic structure computation of Au with 256 atoms, which is written as follows:

$$(\sigma_\ell S - H)\boldsymbol{x}^{(\ell)} = \boldsymbol{e}_1, \quad \ell = 1, 2, \ldots, 1001,$$

where $\sigma_\ell = 0.400 + (\ell - 1 + i)/1000$, $S, H \in R^{2304\times2304}$ are a symmetric positive definite matrix and a symmetric matrix with 1,059,584 entries, $\boldsymbol{e}_1 = (1, 0, \ldots, 0)^{\mathrm{T}}$. Since the generalized shifted COCG method requires the choice of a seed system, we have chosen the optimal seed ($\ell = 174$) in this problem; otherwise some linear systems would remain unsolved unless we use the seed switching technique.

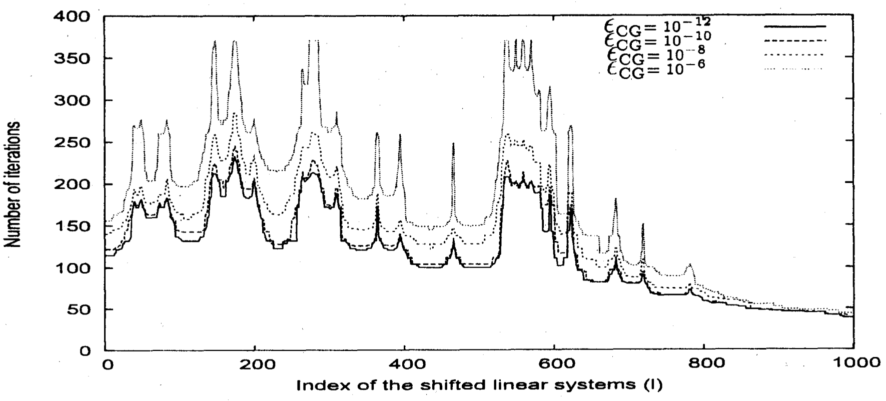

Figure 1: Number of iterations for the generalized shifted COCG method versus the index of the shifted linear systems for example 1.

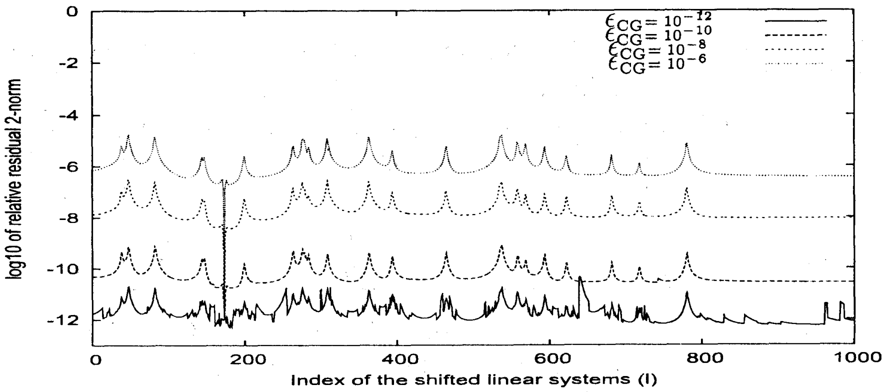

Figure 2: The dependence of the true relative residual 2-norm on the accuracy of the inner solutions for example 1.

Figure 1 shows the dependency of required number of iterations for the generalized shifted COCG method on the accuracy of inner solutions (3.6). From Fig.1, the more accurate we solve the inner linear systems, the less number of iterations were required. Since the inner linear systems are solved roughly, we need to check the loss of accuracy of the outer solutions. So, we next show the dependency of the accuracy for the outer solutions of (1.1) on the accuracy of inner solutions of (3.6) in Fig.2. We see in Fig. 2 that the accuracy is almost the same order as that of the inner solutions. This example implies how robust the present algorithm is even if we solve the inner solutions roughly.

The CPU time of the generalized shifted COCG method is given in Table 1. From Table 1, we see that roughly solving inner linear systems does not always lead to time-efficient, this is because total number of iterations for the seed system tended to increase, see Fig. 1.

The results using the seed switching technique are described in Table 2. In Table 2, 470 linear systems remained unsolved after the generalized shifted COCG method with the initial seed $0.400 + 0.001i$ was solved, which can be a practical issue. Using the seed switching technique, it automatically chose the second seed $0.573 + 0.001i$, and as a result all linear systems were solved. Next, we chose the initial seed $1.400 + 0.001i$. Then, the switch required three times. In two cases, the last seed values were same. We have checked that when we used the last seed value, no switch occured during the iterations for solving all linear systems. In our numerical experiments, the number of switching actions was, at most, three. In terms of CPU time, the generalized shifted COCG using seed switching technique required 33.70 sec. with the initial seed $0.400 + 0.001i$ and 34.30 sec. with the initial seed $1.400 + 0.001i$, where $\epsilon_{\text{CG}} = 10^{-12}$. Finally, the CPU time of the COCG method applied to all shifted linear systems was 1217.3 sec. We also used the COCG method with IC(0)-type preconditioner, but it did not improve the performance due to large number of incomplete factorizations.

Table 1. CPU time required for solving generalized shifted linear systems for example 1.

	Average number of	CPU Time [sec.]		
ϵ_{CG}	CG iterations	seed system	shifted systems	Total
10^{-12}	18.0	30.25	4.78	35.03
10^{-10}	15.7	28.32	4.95	33.27
10^{-8}	13.0	28.17	5.75	33.92
10^{-6}	10.5	31.15	7.23	38.38

Table 2. Results on the generalized shifted COCG method using seed switching technique for example 1.

	Seed value	number of unsolved systems
Initial seed	0.400+0.001i	470
1st switch	0.573+0.001i	0
Initial seed	1.400+0.001i	986
1st switch	0.486+0.001i	186
2nd switch	0.679+0.001i	8
3rd switch	0.573+0.001i	0

4.2 Example 2

The second problem is larger than the first problem, which comes from the electronic structure computation of Au with 864 atoms:

$$(\sigma_\ell S - H)\boldsymbol{x}^{(\ell)} = \boldsymbol{e}_1, \quad \ell = 1, 2, \ldots, 1001,$$

where $\sigma_\ell = 0.400 + (\ell - 1 + i)/1000$, $S, H \in R^{7776 \times 7776}$ are a symmetric positive definite matrix and a symmetric matrix with 3,619,104 entries. Other conditions are the same as in example 1. From Figs. 3 and 4, we see similar tendency to that shown in example 1.

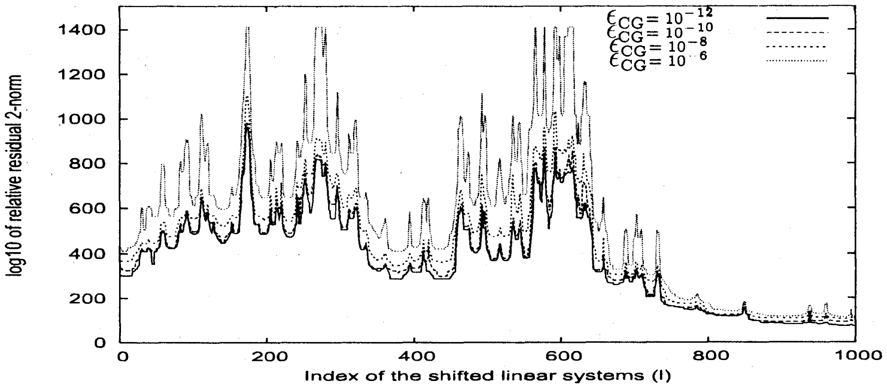

Figure 3: Number of iterations for the generalized shifted COCG method versus the index of the shifted linear systems for example 2.

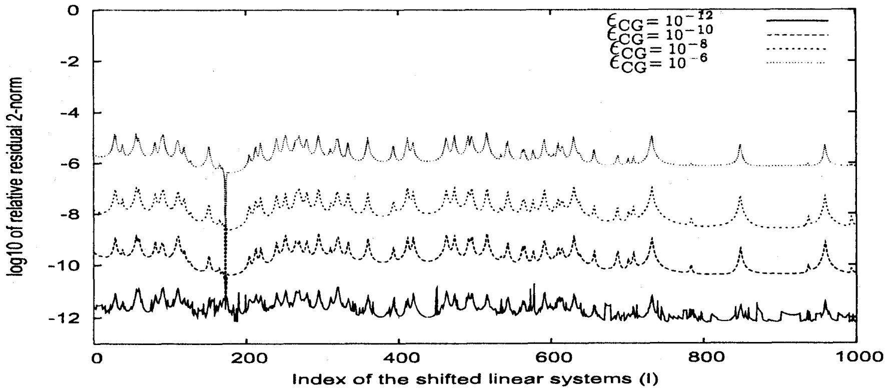

Figure 4: The dependence of the true relative residual 2-norm on the accuracy of the inner solutions for example 2.

The CPU time of the generalized shifted COCG method is given in Table 3. From Table 3, the most time-consuming part was to solve the seed system. So, it is worth reducing the CG iterations by a sophisticated preconditioner. In the case, it also found to be better to use more accurate criterion for the inner solutions. The number of switching actions was, at most, two. In terms of CPU time in Table 4, The generalized shifted COCG using seed switching technique required 522.0 sec. with the initial seed $0.400 + 0.001i$ and 525.7 sec. with the initial seed $1.400 + 0.001i$, where $\epsilon_{\text{CG}} = 10^{-12}$.

Finally, we note that the CPU time of the COCG method without preconditioning, which was better than IC(0)-type preconditioning, applied to all shifted linear systems was 16867.8 sec.

Table 3. CPU time required for solving generalized shifted linear systems for example 2.

	Average number of	CPU Time [sec.]		
ϵ_{CG}	CG iterations	seed system	shifted systems	Total
10^{-12}	20.9	484.5	50.1	534.6
10^{-10}	18.0	438.2	52.5	490.7
10^{-8}	14.9	457.4	65.6	523.0
10^{-6}	11.9	446.8	76.4	523.2

Table 4. Results on the generalized shifted COCG method using seed switching technique for example 2.

	Seed value	number of unsolved systems
Initial seed	0.400+0.001i	524
1st switch	0.572+0.001i	0
Initial seed	1.400+0.001i	998
1st switch	0.556+0.001i	275
2nd switch	0.574+0.001i	0

5 Conclusion

In this paper, the shifted COCG method was extended to solving generalized shifted linear systems with complex symmetric matrices. The resulting algorithm, the generalized shifted COCG method, was derived from two different ways: (1) the shifted COCG method with a bilinear form and (2) a simplification of the shifted Bi-CG method. We have learned that in our numerical examples, we can use iterative methods for solving the inner linear systems, and the accuracy of the solutions depends linearly on the accuracy of the solutions for the inner linear systems. From numerical examples, the algorithm found to be highly attractive when the inner linear systems can efficiently be solved. Since the present strategy can also be applied to the shifted QMR method and the shifted WQMR method, the comparison among them will be our future work.

The generalized shifted COCG method has already been implemented in the quantum mechanical nanomaterial simulation code 'ELSES' (http://www.elses.jp) and will be applied to various nanomaterials as interdisciplinary research between applied mathematics and nanomaterial science. See a recent paper [5] and references therein.

Acknowledgement

This work was partially supported by KAKENHI (Grant No. 21760058).

References

[1] B. N. Datta and Y. Saad, Arnoldi methods for large Sylvester-like observer matrix equations, and an associated algorithm for partial spectrum assignment, Linear Algebra Appl., 154-156 (1991), pp. 225-244.

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

[5] T. Fujiwara, S. Yamamoto, T. Hoshi, S. Nishino, H.Teng, T. Sogabe, S.-L. Zhang, M. Ikeda, M. Nakashima, N. Watanabe, Large-scale Electronic Structure Calculation Theory and Applications, Preprint; http://arxiv.org/abs/1002.1202

[6] K. Meerbergen, The solution of parametrized symmetric linear systems, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 1038-1059.

[7] 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.

[8] 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.

[9] T. Sogabe, T. Hoshi, S.-L. Zhang, and T. Fujiwara, On a weighted quasi-residual minimization strategy of the QMR method for solving complex symmetric shifted linear systems, Electron. Trans. Numer. Anal., 31 (2008), pp. 126-140.

[10] 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.

[11] S. Yamamoto, T. Sogabe, T. Hoshi, S.-L. Zhang, and T. Fujiwara, Shifted COCG method and its application to double orbital extended Hubbard model, J. Phys. Soc. Jpn., Vol. 77, No. 11, 114713 (2008), pp. 1-8.