Krylov subspace methods for solving quadratic eigenvalue problems

Juan Song*

Abstract

In this paper, the Arnoldi-type process and symmetric Lanczos-type process for solving large scale quadratic eigenvalue problem $(\lambda^2 A + \lambda B + C)x = 0$ are given. One decomposition theorem about the matrices A, B and C is obtained based on the Householder transformation. The advantage of the Arnoldi-type process and symmetric Lanczos-type process is that they can preserve the matrix structure and properties of the original problems. Finally, some numerical examples are presented to show the efficiency of the proposed methods.

Keywords: Quadratic matrix polynomial, Krylov subspace, Arnoldi-type process, Symmetric Lanczos-type process.

1. Introduction

To find a scalar $\lambda \in C$ and nontrivial vectors $x \in C^n$ satisfying the following equation

$$(\lambda^2 A + \lambda B + C)x = 0 \tag{1.1}$$

is known as the quadratic eigenvalue problem (QEP). Here, $A, B, C \in C^{n\times n}$ is called as the coefficient matrices, λ is called as the eigenvalue, nonzero vectors x is the associated eigenvectors of the QEP, (λ, x) is known as eigenpair or Ritz pair, λ and x are called as Ritz value and Ritz vector, respectively. The studies on the QEP have attracted more and more attentions recently. In general, there are mainly two methods for solving the QEP, namely lineralization method and direct projection method [17].

Linearization method is to transform the QEP (1.1) into an linear form $(M - \lambda N)y = 0$ equivalently, where $M, N \in C^{2n\times 2n}$, $y \in C^{2n}$. The possibility of lineralization was proved in [5, 17], and the properties of lineralization method were studied in [3, 13, 14, 16]. But the drawback of linearization method is the double size of the problem (1.1), and make the condition number and backward error larger. Meanwhile, the essential spectral properties of the original problems may not be preserved.

Krylov subspace based on matrix $A \in C^{n\times n}$ and vector $\upsilon \in C^n$ is of the form

$$K_k(A, \upsilon) = span\{\upsilon, A\upsilon, A^2\upsilon, \cdots, A^{k-1}\upsilon\}.$$

Krylov subspace method is often used to solve matrix computational problem, such as linear system and eigenvalue problem [1, 2, 6, 7, 11, 12, 15, 18]. The

*School of Mathematical Sciences, Nanjing Normal University, Nanjing 210097, P. R. China, Email: song12368@163.com

main advantage of Krylov subspace techniques is that it can transform large scale problem into small size problem and find the desired eigenvalues.

By building one orthogonal bases V_k and projecting the original problem (1.1) into the problem $V_k^*(\lambda^2 A + \lambda B + C)V_k V_k^* x = 0$ with smaller size, the projection method can be applied to the QEP directly. The advantage of the projection methods is the spectral property of the original problem can be guaranteed by preserving the structure of the coefficient matrix, such as the symmetry or skew-symmetry or positive-definiteness or semipositive-definiteness. In order to obtain a projected lower-dimensional matrix polynomial to approximate the original one, a Krylov-type projection process was applied to the coefficient matrices B and C simultaneously [12, 7]. However, in the method, the coefficient matrix A should be the unity matrix I, or should be transformed to the unity matrix I by displacement inverse transformation, which makes the computation more complex. Therefore, in this paper, the Krylov-type projection process was applied to all three coefficient matrices A, B and C simultaneously, in which the coefficient matrix A can be projected directly, and need not be unity matrix or transformed.

This paper is organized as follows. In Section 2, an Arnoldi-type process and a symmetric Lanczos-type process for solving the large scale QEP (1.1) are presented, and the matrix A is different from the investigations in [12, 7], where A is the unity matrix I. In Section 3, the residual upper bound for approximate Ritz pairs are given out. In Section 4, combining the orthogonal basis generated in Section 2 with the refined idea, we give out the refined Arnoldi-type algorithm for solving quadratic matrix polynomial. Finally, numerical examples are given to illustrated the efficiency of given algorithm in Section 5.

Throughout this article, we use following notations. $\|\cdot\|_2$ denotes 2-norm. $\cdot^\top$ denotes the transpose and $\cdot^*$ denotes the conjugate transpose. I_n refers the $n \times n$ identity matrix, e_j denotes its jth column. MATLAB-like notations are adopted: the ith to jth entries of v consists $v_{(i:j)}$, the intersection of rows i to j and columns k to l of the matrix X consists the submatrix $X_{(i:j,k:l)}$, $X_{(:,k:l)}$ refers all rows and kth to lth columns of X, $X_{(i:j,:)}$ refers ith to jth rows and all columns of X.

2. New Arnoldi-type process and symmetric Lanczos-type process for $(\lambda^2 A + \lambda B + C)x = 0$

For solving one special kind of quadratic matrix polynomial

$$(\lambda^2 I - \lambda B - C)x = 0,$$

an Arnoldi-type process and a symmetric Lanczos-type process were presented in [12] and [7], respectively.

In this section, based on orthogonal transformations of coefficient matrices A, B and C simultaneously, we propose an new Arnoldi-type process and symmetric Lanczos-type process for solving the QEP (1.1).

2.1. Decomposition theorem for the coefficient matrices A, B and C.

By using Householder transformation, the following decomposition theorem for the coefficient matrices A, B and C were derived.

2.1. Lemma. *There exists an unitary matrix $Q \in C^{n\times n}$ with $Qe_1 = e_1$ satisfying*

$$Q^*AQ = H_a \equiv (h_{a;ij}),\ Q^*BQ = H_b \equiv (h_{b;ij}), Q^*CQ = H_c \equiv (h_{c;ij})$$

where $h_{a;ij} = 0$ for $i \geq 3j$, $h_{b;ij} = 0$ for $i \geq 3j+1$, $h_{c;ij} = 0$ for $i \geq 3j+2$.

Proof. Split

$$A = \left[\begin{array}{c|c} a_{11} & a_2^\top \\ \hline a_1 & A_{22} \end{array}\right],$$

where $a_1 = (a_{21}, a_{31}, ..., a_{n1})^\top$, $a_2 = (a_{12}, a_{13}, ..., a_{1n})^\top$.

There exists an unitary matrix $\hat{Q}_{1a} \in C^{(n-1)\times(n-1)}$ satisfying $\hat{Q}_{1a}^* a_1 = \alpha_1 e_1$. Let $Q_{1a} = diag(1, \hat{Q}_{1a})$. Then, we have

$$Q_{1a}^* A Q_{1a} = \left[\begin{array}{c|c} a_{11} & x \\ \hline \alpha_1 & x \\ 0 & X \end{array}\right], \quad Q_{1a}^* B Q_{1a} = \left[\begin{array}{c|c} b_{11} & x \\ b_{21} & x \\ \hline b_1 & X \end{array}\right].$$

Similarly, there exists an unitary matrix $\hat{Q}_{1b} \in C^{(n-2)\times(n-2)}$ satisfying $\hat{Q}_{1b}^* b_1 = \beta_1 e_1$. Let $Q_{1b} = diag(I_2, \hat{Q}_{1b})$. Then, we have

$$Q_{1b}^* Q_{1a}^* B Q_{1a} Q_{1b} = \left[\begin{array}{c|c} b_{11} & x \\ b_{21} & x \\ \hline \beta_1 & x \\ 0 & X \end{array}\right], \quad Q_{1b}^* Q_{1a}^* C Q_{1a} Q_{1b} = \left[\begin{array}{c|c} c_{11} & x \\ c_{21} & x \\ c_{31} & x \\ \hline c_1 & X \end{array}\right].$$

In the same way, we can find an unitary matrix $\hat{Q}_{1c} \in C^{(n-3)\times(n-3)}$ satisfying $\hat{Q}_{1c} c_1 = \gamma_1 e_1$. Let $Q_{1c} = diag(I_3, \hat{Q}_{1c})$ and define $Q_1 = Q_{1a} Q_{1b} Q_{1c}$. Then, we have

$$Q_1^* A Q_1 = \left[\begin{array}{c|c} a_{11} & x \\ \hline \alpha_1 & x \\ 0 & X \end{array}\right], \quad Q_1^* B Q_1 = \left[\begin{array}{c|c} b_{11} & x \\ b_{21} & x \\ \hline \beta_1 & x \\ 0 & X \end{array}\right], \quad Q_1^* C Q_1 = \left[\begin{array}{c|c} c_{11} & x \\ c_{21} & x \\ c_{31} & x \\ \hline \gamma_1 & x \\ 0 & X \end{array}\right].$$

According to the above transformation, the first columns of the matrix A, B and C have been transformed into the desired forms. Then, we transform the second columns of the matrix A, B, and C into the desired forms in the same way.

Split the matrices

$$Q_1^* A Q_1 = \left[\begin{array}{c|c|c} x & x & x \\ \alpha_1 & x & x \\ 0 & x & x \\ 0 & a_{42} & x \\ \hline 0 & a_2 & X \end{array}\right],$$

then there exists a unitary matrix $\hat{Q}_{2a} \in C^{(n-4)\times(n-4)}$ satisfying $\hat{Q}_{2a}^* a_2 = \alpha_2 e_1$.

Let $Q_{2a} = diag(I_4, \hat{Q}_{2a})$. Then we have

$$Q_{2a}^* Q_1^* A Q_1 Q_{2a} = \left[\begin{array}{c|c|c} x & x & x \\ \alpha_1 & x & x \\ 0 & x & x \\ 0 & a_{42} & x \\ \hline 0 & \alpha_2 & x \\ 0 & 0 & X \end{array}\right], \quad Q_{2a}^* Q_1^* B Q_1 Q_{2a} = \left[\begin{array}{c|c|c} b_{11} & x & x \\ b_{21} & x & x \\ \beta_1 & x & x \\ 0 & x & x \\ 0 & b_{52} & x \\ \hline 0 & b_2 & x \end{array}\right].$$

Find an unitary matrix $\hat{Q}_{2b} \in C^{(n-5)\times(n-5)}$ satisfying $\hat{Q}_{2b}^* b_2 = \beta_2 e_1$. Let $Q_{2b} = diag(I_5, \hat{Q}_{2b})$. Then, we have

$$Q_{2b}^* Q_{2a}^* Q_1^* B Q_1 Q_{2a} Q_{2b} = \left[\begin{array}{c|c|c} b_{11} & x & x \\ b_{21} & x & x \\ \beta_1 & x & x \\ 0 & x & x \\ 0 & b_{52} & x \\ \hline 0 & \beta_2 & x \\ 0 & 0 & x \end{array}\right], \quad Q_{2b}^* Q_{2a}^* Q_1^* C Q_1 Q_{2a} Q_{2b} = \left[\begin{array}{c|c|c} c_{11} & x & x \\ c_{21} & x & x \\ c_{31} & x & x \\ \gamma_1 & x & x \\ 0 & x & x \\ 0 & c_{62} & x \\ \hline 0 & c_2 & X \end{array}\right].$$

Find an unitary matrix $\hat{Q}_{2c}^* \in C^{(n-6)\times(n-6)}$ satisfying $\hat{Q}_{2c}^* c_2 = \gamma_2 e_1$. Let $Q_{2c} = diag(I_6, \hat{Q}_{2c})$ and define $Q_2 = Q_{2a} Q_{2b} Q_{2c}$. Then, we have

$$Q_2^* Q_1^* A Q_1 Q_2 = \left[\begin{array}{c|c|c} x & x & x \\ \alpha_1 & x & x \\ 0 & x & x \\ 0 & a_{42} & x \\ \hline 0 & \alpha_2 & x \\ 0 & 0 & X \end{array}\right], \; Q_2^* Q_1^* B Q_1 Q_2 = \left[\begin{array}{c|c|c} x & x & x \\ x & x & x \\ \beta_1 & x & x \\ 0 & x & x \\ 0 & b_{52} & x \\ \hline 0 & \beta_2 & x \\ 0 & 0 & x \end{array}\right],$$

$$Q_2^* Q_1^* C Q_1 Q_2 = \left[\begin{array}{c|c|c} x & x & x \\ x & x & x \\ x & x & x \\ \gamma_1 & x & x \\ 0 & x & x \\ 0 & c_{62} & x \\ \hline 0 & \gamma_2 & x \\ 0 & 0 & X \end{array}\right].$$

The following proof can be continued in a similar way. At the jth step, the jth column of matrices A, B and C has at most $3j-1$, $3j$ and $3j+1$ nonzero entries at the top respectively. The reduction can be completed by setting $Q = Q_1 Q_2 \cdots Q_k$, where $k \leq n/3$.

Obviously, we have

$$Q e_1 = Q_1 Q_2 \cdots Q_k e_1 = e_1,$$

where $Q_1 = Q_{1a} Q_{1b} Q_{1c}$, $Q_2 = Q_{2a} Q_{2b} Q_{2c}$, $\cdots$, $Q_k = Q_{ka} Q_{kb} Q_{kc}$. □

Based on the above discussions, we can get the following another description of the decomposition theorem for the coefficient matrices.

2.2. Theorem. *Given $q_1 \in C^n$ with $\|q_1\|_2 = 1$, there is an unitary matrix $Q \in C^{n\times n}$ with $Qe_1 = q_1$, such that*

$$(2.1) Q^*AQ = H_a = (h_{a;ij}),\ Q^*BQ = H_b = (h_{b;ij}),\ Q^*CQ = H_c = (h_{c;ij})$$

satisfied $h_{a;ij} = 0$ for $i \geq 3j$, $h_{b;ij} = 0$ for $i \geq 3j+1$, $h_{c;ij} = 0$ for $i \geq 3j+2$.

Proof. There exists an unitary matrix $Q_0 \in C^{n\times n}$ with $Q_0e_1 = q_1$. Then, applying Lemma 2.1 to $Q_0^*AQ_0$, $Q_0^*BQ_0$ and $Q_0^*CQ_0$ to get an unitary $\hat{Q} \in C^{n\times n}$ with $\hat{Q}e_1 = e_1$ such that

$$\hat{Q}^*(Q_0^*AQ_0)\hat{Q} = H_a,\ \hat{Q}^*(Q_0^*BQ_0)\hat{Q} = H_b,\ \hat{Q}^*(Q_0^*CQ_0)\hat{Q} = H_c$$

have the desired forms. Then, the proof is completed by letting $Q = Q_0\hat{Q}$. □

2.2. A new Arnoldi-type process for $(\lambda^2A + \lambda B + C)x = 0$. According to Lemma 2.1 and Theorem 2.2, the reduced matrices H_a, H_b, H_c of coefficient matrices A, B and C can be obtained, but they are of little use in the numerical computation when A, B and C are large and sparse. Therefore, a new Arnoldi-type process for the QEP (1.1) were presented in this section.

Rewrite (2.1) as

$$AQ = QH_a,\ BQ = QH_b,\ CQ = QH_c.$$

Inspecting the jth column, we see

$$(2.2) \qquad Aq_j = \sum_{i=1}^{3j-2} q_i h_{a;ij} + q_{3j-1}h_{a;3j-1,j},$$

$$(2.3) \qquad Bq_j = \sum_{i=1}^{3j-1} q_i h_{b;ij} + q_{3j}h_{b;3j,j},$$

$$(2.4) \qquad Cq_j = \sum_{i=1}^{3j} q_i h_{c;ij} + q_{3j+1}h_{c;3j+1,j}.$$

From (2.2) and since $q_1, q_2, \cdots, q_{3j}$ is orthogonal, we have

$$\begin{aligned} h_{a;ij} &= q_i^*Aq_j \quad for \quad i \leq 3j-2, \\ h_{a;3j-1,j} &= \left\| Aq_j - \sum_{i=1}^{3j-2} q_i h_{a;ij} \right\|_2, \\ q_{3j-1} &= (Aq_j - \sum_{i=1}^{3j-2} q_i h_{a;ij})/h_{a;3j-1,j}, \end{aligned}$$

where assume that $h_{a;3j-1,j} \neq 0$.

Similarly, assuming $h_{b;3j,j} \neq 0$, the formula (2.3) implies

$$\begin{aligned} h_{b;ij} &= q_i^* B q_j \quad for \quad i \leq 3j-1, \\ h_{b;3j,j} &= \left\| Bq_j - \sum_{i=1}^{3j-1} q_i h_{b;ij} \right\|_2, \\ q_{3j} &= (Bq_j - \sum_{i=1}^{3j-1} q_i h_{b;ij})/h_{b;3j,j}. \end{aligned}$$

In the same way, assuming $h_{c;3j+1,j} \neq 0$, the formula (2.4) implies

$$\begin{aligned} h_{c;ij} &= q_i^* C q_j \quad for \quad i \leq 3j, \\ h_{c;3j+1,j} &= \left\| Cq_j - \sum_{i=1}^{3j} q_i h_{c;ij} \right\|_2, \\ q_{3j+1} &= (Cq_j - \sum_{i=1}^{3j} q_i h_{c;ij})/h_{c;3j+1,j}. \end{aligned}$$

Above derivation leads to a process that q_{3j-1}, q_{3j}, q_{3j+1} can be constructed from q_1, q_2, $\cdots$, q_{3j-2}. After k steps of construction, we can obtain q_1, q_2, $\cdots$, q_{3k+1} such that

$$\begin{aligned} AQ_{(:,1:k)} &= Q_{(:,1:3k-1)} H_{a(1:3k-1,1:k)}, \; BQ_{(:,1:k)} = Q_{(:,1:3k)} H_{b(1:3k,1:k)}, \\ CQ_{(:,1:k)} &= Q_{(:,1:3k+1)} H_{c(1:3k+1,1:k)}. \end{aligned}$$

Figure 1 shows the computed parts of H_a, H_b and H_c when $k = 6$, and the entries marked by unfilled circles are not computed yet.

From the computed entries, the projections of A, B and C onto $span\{Q_{(:,1:k)}\}$ can be obtained. The entries marked by unfilled circles in Figure 1 are computed by

$$h_{a,ij} = q_i^* A q_j, \; h_{b,ij} = q_i^* B q_j, \; h_{c,ij} = q_i^* C q_j,$$

for $1 \leq i \leq 3k+1$ and $k+1 \leq j \leq 3k+1$, which give the projections on $span\{Q_{(:,1:3k+1)}\}$. In above analysis, it is assumed that $h_{a;3j-1,j} \neq 0$, $h_{b;3j,j} \neq 0$, $h_{c;3j+1,j} \neq 0$. When $h_{a,ij} = 0$ or $h_{b,ij} = 0$ or $h_{c,ij} = 0$, the process can be continued by continuing the next step directly, although there is no new q-vector can be generated.

According to the above mentioned analysis, the new q-vectors can be generated as the following steps. Let N be the number of q-vectors already generated, and $N = 1$ at the beginning of the process. At the first step, the matrix A is applied to q_1, and if a new q-vector is generated, $N = N+1$, otherwise, N is invariant. Then, the matrix B is applied to q_1, and if a new q-vector is generated, $N = N+1$, otherwise, N is invariant. In sequence, the matrix C is applied to q_1, and if a new q-vector is generated, $N = N+1$, otherwise, N is invariant. After the above steps, if $N = 1$, the process can be terminated since the subspace $span\{q_1\}$ is the invariant subspace about the matrices A, B and C, otherwise, the matrices A, B and C should be applied to q_2 in the same way. In general, at the jth step, let $q_1, q_2, \cdots, q_N$ be the N q-vectors have been generated, and $q_1, q_2, \cdots, q_{j-1}$ been the $j-1$ q-vectors have applied by the matrices A, B and C, and if $N =$

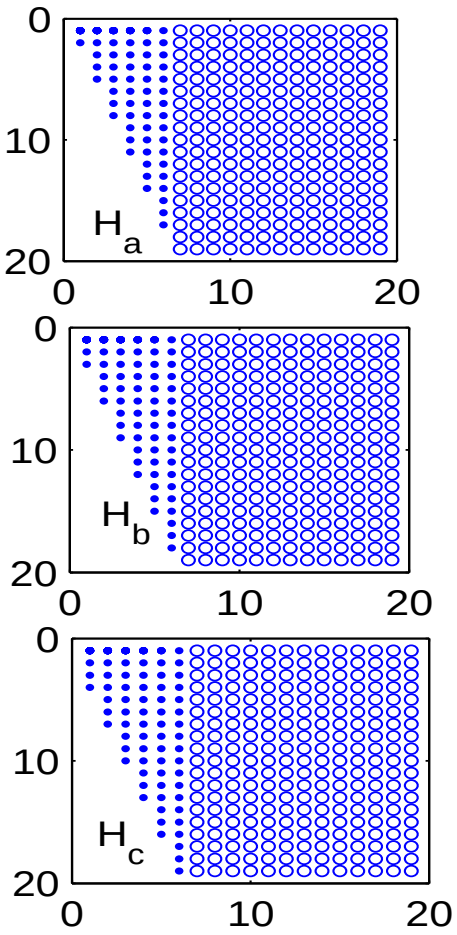

Figure 1. The sparsity of the matrix H_a, H_b and H_c

$j-1$, the process can be terminated since the subspace $span\{q_1, q_2, \cdots, q_N\}$ is the invariant subspace about the matrices A, B and C, otherwise, the matrices A, B and C should be applied to q_j in the same way. The process continues until $N = j-1$ or a preselected k number of steps is completed, in which N must satisfy $N \leq 3k+1$. In order to utilize fully the information presented by the generated subspace $span\{Q(:, 1:N)\}$, the fully projected matrices $H_{a(1:N,1:N)}$, $H_{b(1:N,1:N)}$ and $H_{c(1:N,1:N)}$ are computed in our later numerical examples. Based on the above analysis, we have the following algorithm:

Algorithm 1: New Arnoldi-type process

1. Given q_1 with $\|q_1\|_2 = 1$
2. $N = 1$
3. For $j = 1, 2, \cdots, k$ do
4. If $j > N$, break
5. $\hat{q} = Aq_j$

6. For $i = 1, 2, \cdots, N$ do
7. $h_{a;ij} = q_i^* \hat{q}$; $\hat{q} = \hat{q} - q_i h_{a;ij}$
8. End do
9. $h_{a;N+1,j} = \|\hat{q}\|_2$
10. If $h_{a;N+1,j} > 0$
11. $N = N + 1, q_N = \hat{q}/h_{a;Nj}$
12. End if
13. $\hat{q} = Bq_j$
14. For $i = 1, 2, \cdots, N$ do
15. $h_{b;ij} = q_i^* \hat{q}$; $\hat{q} = \hat{q} - q_i h_{b;ij}$
16. End do
17. $h_{b;N+1,j} = \|\hat{q}\|_2$
18. If $h_{b;N+1,j} > 0$
19. $N = N + 1$, $q_N = \hat{q}/h_{b;Nj}$
20. End if
21. $\hat{q} = Cq_j$
22. For $i = 1, 2, \cdots, N$ do
23. $h_{c;ij} = q_i^* \hat{q}$; $\hat{q} = \hat{q} - q_i h_{c;ij}$
24. End do
25. $h_{c;N+1,j} = \|\hat{q}\|_2$
26. If $h_{c;N+1,j} > 0$
27. $N = N + 1$, $q_N = \hat{q}/h_{c;Nj}$
28. End if
29. End do

In the practical numerical computation, the following statements should be made for Algorithm 1. In practical implement of line 10 to line 26, an appropriate error, e.g., $h_{a;N+1,j} > n\varepsilon\|A\|_2$, $h_{b;N+1,j} > n\varepsilon\|B\|_2$, $h_{c;N+1,j} > n\varepsilon\|C\|_2$ can be permitted, where ε is the machine roundoff unit.

Denote α_j = value of N at line 12 at step j, β_j = value of N at line 20 at step j, γ_j = value of N at line 28 at step j with $\alpha_0 = \beta_0 = \gamma_0 = 1$. Then,

$$Aq_j = \sum_{i=1}^{\alpha_j} h_{a;ij} q_i, \quad Bq_j = \sum_{i=1}^{\beta_j} h_{b;ij} q_i, \quad Cq_j = \sum_{i=1}^{\gamma_j} h_{c;ij} q_i.$$

Thus, when the above process is completed, we have

$$\begin{aligned} AQ_{(:,1:k)} &= Q_{(:,1:\alpha_k)} H_{a(1:\alpha_k,1:k)} \\ BQ_{(:,1:k)} &= Q_{(:,1:\beta_k)} H_{b(1:\beta_k,1:k)} \\ CQ_{(:,1:k)} &= Q_{(:,1:\gamma_k)} H_{c(1:\gamma_k,1:k)} \end{aligned} \tag{2.5}$$

However, if the j-loop BREAK out at line 4, an invariant subspace of A, B and C is obtained as follows

$$\begin{aligned} AQ_{(:,1:N)} &= Q_{(:,1:N)} H_{a(1:N,1:N)} \\ BQ_{(:,1:N)} &= Q_{(:,1:N)} H_{b(1:N,1:N)} \\ CQ_{(:,1:N)} &= Q_{(:,1:N)} H_{c(1:N,1:N)} \end{aligned}$$

Moreover, the nonzero entries of the jth column of H_a, H_b and H_c is contained in the first α_j, β_j and γ_j entries respectively. α_j, β_j and γ_j can increase at most by 3 at each step.

When A, B and C are Hermitian, H_a, H_b, H_c are also Hermitian. In this case, their upper triangular parts need not be fully computed. Obviously, the following simple recurrences holds:

$$\begin{aligned} h_{a;\alpha_j,j} q_{\alpha_j} &= Aq_j - \sum_{1\le i<\alpha_j, \alpha_i \ge j} h_{a;ij} q_i, \\ h_{b;\beta_j,j} q_{\beta_j} &= Bq_j - \sum_{1\le i<\beta_j, \beta_i \ge j} h_{b;ij} q_i, \\ h_{c;\gamma_j,j} q_{\gamma_j} &= Cq_j - \sum_{1\le i<\gamma_j, \gamma_i \ge j} h_{c;ij} q_i. \end{aligned}$$

Similar to Algorithm 1, we have the following Algorithm:

Algorithm 2: Symmetric Lanczos-type process

1. Given q_1 with $\|q_1\|_2 = 1$;
2. $N = 1$; $\alpha_1 = 1$; $\beta_1 = 1$; $\gamma_1 = 1$; $l_a = 1$; $l_b = 1$; $l_c = 1$;
3. For $j = 1, 2, \cdots, k$ do
4. If $j > N$, break
5. $\hat{q} = Aq_j$;
6. if $j > \alpha_{l_a}$ then $l_a = l_a + 1$;
7. For $i = l_a, \cdots, N$ do
8. $h_{a;ij} = q_i^* \hat{q}$; $\hat{q} = \hat{q} - q_i h_{a;ij}$;
9. End do
10. $h_{a;N+1,j} = \|\hat{q}\|_2$;
11. If $h_{a;N+1,j} > 0$
12. $N = N + 1$, $q_N = \hat{q}/h_{a;Nj}$, $\alpha_j = N$;
13. End if
14. $\hat{q} = Bq_j$;
15. if $j > \beta_{l_b}$ then $l_b = l_b + 1$;
16. For $i = l_b, \cdots, N$ do
17. $h_{b;ij} = q_i^* \hat{q}$; $\hat{q} = \hat{q} - q_i h_{b;ij}$;
18. End do
19. $h_{b;N+1,j} = \|\hat{q}\|_2$;
20. If $h_{b;N+1,j} > 0$
21. $N = N + 1$, $q_N = \hat{q}/h_{b;Nj}$, $\beta_j = N$;
22. End if
23. $\hat{q} = Cq_j$;
24. if $j > \gamma_{l_c}$ then $l_c = l_c + 1$;
25. For $i = l_c, \cdots, N$ do
26. $h_{c;ij} = q_i^* \hat{q}$; $\hat{q} = \hat{q} - q_i h_{c;ij}$;
27. End do
28. $h_{c;N+1,j} = \|\hat{q}\|_2$;
29. If $h_{c;N+1,j} > 0$
30. $N = N + 1$, $q_N = \hat{q}/h_{c;Nj}, \gamma_j = N$;
31. End if

32. End do

3. Analysis of residual upper bound of the algorithm for $(\lambda^2 A + \lambda B + C)x = 0$

From above discussions, we know that the solution of the QEP (1.1) can be approximated by the solution of

$$(\lambda^2 H_{a(1:N,1:N)} + \lambda H_{b(1:N,1:N)} + H_{c(1:N,1:N)})Q^\top x = 0,$$

where $Q_{(:,1:N)}, H_{a(1:N,1:N)}$, $H_{b(1:N,1:N)}$ and $H_{c(1:N,1:N)}$ be produced by Algorithms 1 or 2.

That is, if (θ_i, ν_i) is an eigenvalue and right eigenvector of

$$(\lambda^2 H_{a(1:N,1:N)} + \lambda H_{b(1:N,1:N)} + H_{c(1:N,1:N)})\nu = 0,$$

then the eigenvalue and eigenvector of the QEP (1.1) can be approximated by eigenpairs (θ_i, x_i), where $x_i = Q_{(:,1:N)}\nu_i$.

The above analysis points that an original quadratic eigenvalue problem can be approximated by a projection quadratic eigenvalue problem, and the accuracy can be calculated by the residual error. Therefore, in the following, the residual upper bound for symmetric Lanczos-type process are derived. Corresponding results for the new Arnoldi-type process can be derived similarly, the details were omitted here.

3.1. Theorem. *If the Ritz value and Ritz vector are obtained by Algorithm 2, then the following inequality*

$$\begin{aligned} &\|(\theta_i^2 A + B\theta_i + C)x_i\|_2 \\ &\le \|Q\|_2(|\theta_i|^2\|H_{a(N+1:\alpha_N,p:N)}\|_2 + |\theta_i|\|H_{b(N+1:\beta_N,p:N)}\|_2 \\ &+\|H_{c(N+1:\gamma_N,p:N)}\|_2)\|\nu_{i(p:N)}\|_2 \end{aligned} \tag{3.1}$$

holds, where p is the smallest integer such that $\gamma_p > N$ and is equal to the value of l_c at step $N+1$.

Proof. According to (2.5), we have

$$\begin{aligned} AQ_{(:,1:N)} &= Q_{(:,1:N)}H_{a(1:N,1:N)} + Q_{(:,N+1:\alpha_N)}H_{a(N+1:\alpha_N,1:N)}, \\ BQ_{(:,1:N)} &= Q_{(:,1:N)}H_{b(1:N,1:N)} + Q_{(:,N+1:\beta_N)}H_{b(N+1:\beta_N,1:N)}, \\ CQ_{(:,1:N)} &= Q_{(:,1:N)}H_{c(1:N,1:N)} + Q_{(:,N+1:\gamma_N)}H_{c(N+1:\gamma_N,1:N)}. \end{aligned}$$

Then, we have

$$\begin{aligned} (\theta_i^2 A + \theta_i B + C)x_i &= (\theta_i^2 AQ_{(:,1:N)} + \theta_i BQ_{(:,1:N)} + CQ_{(:,1:N)})\nu_i \\ &= Q_{(:,1:N)}(\theta_i^2 H_{a(1:N,1:N)} + \theta_i H_{b(1:N,1:N)} + H_{c(1:N,1:N)})\nu_i \\ &\quad +(\theta_i^2 Q_{(:,N+1:\alpha_N)}H_{a(N+1:\alpha_N,1:N)} + \theta_i Q_{(:,N+1:\beta_N)}H_{b(N+1:\beta_N,1:N)} \\ &\quad +Q_{(:,N+1:\gamma_N)}H_{c(N+1:\gamma_N,1:N)})\nu_i \\ &= (\theta_i^2 Q_{(:,N+1:\alpha_N)}H_{a(N+1:\alpha_N,p:N)} + \theta_i Q_{(:,N+1:\beta_N)}H_{b(N+1:\beta_N,p:N)} \\ &\quad +Q_{(:,N+1:\gamma_N)}H_{c(N+1:\gamma_N,p:N)})\nu_{i(p:N)}. \end{aligned}$$

Since the first $p-1$ columns of $H_{a(N+1:\alpha_N,1:N)}$, $H_{b(N+1:\beta_N,1:N)}$ and $H_{c(N+1:\gamma_N,1:N)}$ are zeros, (3.1) can be obtained by taking the norm of above formula. □

From (3.1), it is easy to see that the eigenpairs (θ_i, x_i) is good approximation to original problem when $\nu_i(p:N)$ is small.

4. Analysis of refined algorithm for $(\lambda^2 A + \lambda B + C)x = 0$

As [10] defines, for each θ, the refined process is to seek an unit vector $\tilde{\mu} \in gK_\ell(\{A, B, C\}, q_1)$ satisfies

$$\|(\theta^2 A + \theta B + C)\tilde{\mu}\|_2 = \min_{\mu \in gK_\ell(\{A,B,C\},q_1), \|\mu\|_2=1} \|(\theta^2 A + \theta B + C)\mu\|_2, \tag{4.1}$$

and $\tilde{\mu}$ is called a refined eigenvector.

Since Q_ℓ is an orthogonal basis of $gK_\ell(\{A, B, C\}, q_1)$, (4.1) is equivalent to seek an unit vector $\tilde{z} \in C^\ell$ such that $\tilde{\mu} = Q_\ell \tilde{z}$ satisfies

$$\tilde{z} = arg \min_{z \in C^\ell, \|z\|_2=1} \|(\theta^2 A + \theta B + C)Q_\ell z\|_2. \tag{4.2}$$

It is easy to see that $\tilde{z}$ is the right singular vector of $\theta^2 AQ_\ell + \theta BQ_\ell + CQ_\ell$ associated with $\sigma_{\min}(\theta^2 AQ_\ell + \theta BQ_\ell + CQ_\ell)$. Based on Algorithm 1 and refined idea in [10], the following restarted refined Arnoldi-type algorithm can be presented:

Algorithm 3: New restarted refined Arnoldi-type algorithm

1. Given m required eigenpairs, an unit initial vector q_1 and a tolerance tol.
2. Run the Arnoldi-type process to generate an orthogonal basis Q_ℓ of $gK_\ell(\{A, B, C\}, q_1)$.
3. Compute $W_1 = AQ_\ell$, $W_2 = BQ_\ell$, $W_3 = CQ_\ell$.
4. Compute $A_\ell = Q_\ell^* W_1$, $B_\ell = Q_\ell^* W_2$, $C_\ell = Q_\ell^* W_3$, and the eigenpairs of the projection problem

$$(\theta_i^2 A_\ell + \theta_i B_\ell + C_\ell)z_i = 0.$$

Then, select m Ritz values as approximations to the m desired eigenvalues $\theta_i, i = 1, 2, \cdots, m$.

5. For each θ_i, $i = 1, 2, \cdots, m$, based on SVD, $\sigma_{\min}(\theta_i^2 AQ_\ell + \theta_i BQ_\ell + CQ_\ell)$ and eigenvector $\tilde{z}_i$ associated with its smallest singular value can be obtained. Then, the refined eigenvector is $\tilde{\mu}_i = Q_\ell \tilde{z}_i$.
6. Compute the relative residual error by

$$\frac{\|(\theta_i^2 A + \theta_i B + C)\tilde{\mu}_i\|_2}{|\theta_i|^2 ||A\tilde{\mu}_i||_2 + |\theta_i| ||B\tilde{\mu}_i||_2 + ||C\tilde{\mu}_i||_2}, \ i = 1, 2, \cdots, m.$$

If they are all below tol, then stop, else continue.

7. Construct a new initial vector q_1 from $\tilde{\mu}_i, i = 1, 2, \cdots, m$, and return to step 2. Here, q_1 can be obtained as the following combinations:

$$\beta \cdot q_1 = \Sigma_{i=1}^m \|(\theta_i^2 A + \theta_i B + C)\tilde{\mu}_i\|_2 Re\tilde{\mu}_i = Q_\ell \Sigma_{i=1}^m \|(\lambda_i^2 A + \theta_i B + C)\tilde{\mu}_i\|_2 Re\tilde{z}_i.$$

5. Numerical examples

In order to show the efficiency of Algorithms 1 and 2, some numerical examples are indicated in this section, the process is realized by Matlab 7.8 on Pentium(R) Dual-Core CPU. In the following example, if [12, Algorithm 2.1] is used, it means that we transform the QEP (1.1) into $(\lambda^2 I + \lambda A^{-1}B + A^{-1}C)x = 0$, where it is

assumed that A is nonsingular. In the numerical examples, the relative residual norm for an approximate eignpairs (θ_j, x_j) are defined by

$$\gamma_j = \frac{||(\theta_j^2 A + \theta_j B + C)x_j||_2}{|\theta_j|^2||Ax_j||_2 + |\theta_j|||Bx_j||_2 + ||Cx_j||_2}.$$

5.1. Example. In this example, taking $n = 10$, $k = 2$.

$$Q = \begin{bmatrix} 0.3215 & -0.0830 & -0.1955 & 0.3294 & 0.1108 & -0.1143 & -0.8251 & -0.0615 & -0.0741 & 0.1662 \\ 0.3667 & 0.8872 & -0.0975 & -0.0782 & 0.1205 & -0.0637 & 0.0635 & 0.1858 & -0.0751 & -0.0000 \\ 0.3786 & -0.0977 & 0.5776 & -0.0956 & 0.0446 & -0.0399 & 0.1546 & -0.2211 & -0.2664 & 0.5975 \\ 0.1048 & -0.0270 & -0.3103 & -0.0251 & 0.1232 & 0.9065 & 0.0510 & -0.0729 & -0.0879 & 0.1971 \\ 0.2096 & -0.0541 & -0.1274 & 0.8489 & -0.0655 & -0.0627 & 0.4561 & 0.0000 & 0.0000 & 0.0000 \\ 0.3977 & -0.1026 & 0.5516 & 0.0701 & 0.0228 & 0.3065 & -0.1258 & 0.2486 & 0.2253 & -0.5470 \\ 0.4525 & -0.1449 & -0.2844 & -0.2552 & -0.7873 & -0.0495 & 0.0506 & 0.0369 & 0.0456 & 0.0000 \\ 0.4120 & -0.3608 & -0.3342 & -0.2781 & 0.5652 & -0.2356 & 0.2410 & 0.1756 & 0.2173 & 0.0000 \\ 0.1572 & -0.0406 & -0.0956 & -0.0867 & 0.1080 & -0.0605 & 0.0648 & -0.5558 & -0.5927 & -0.5266 \\ 0.0810 & 0.1715 & 0.0140 & -0.0062 & 0.0177 & 0.0064 & 0.0021 & -0.7104 & 0.6774 & 0.0000 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 10^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-7} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^0 \end{bmatrix},$$

$$A = Q^{-1} A_1 Q,$$

$$B = \begin{bmatrix} 5 & 6 & 7 & 8 & 9 & 0 & 1 & 2 & 3 & 4 \\ 6 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 7 & 2 & 3 & 2 & 4 & 5 & 7 & 8 & 9 & 4 \\ 8 & 3 & 2 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 9 & 4 & 4 & 1 & 3 & 3 & 2 & 1 & 5 & 4 \\ 0 & 5 & 5 & 2 & 3 & 4 & 3 & 2 & 1 & 5 \\ 1 & 6 & 7 & 3 & 2 & 3 & 9 & 9 & 4 & 2 \\ 2 & 7 & 8 & 4 & 1 & 2 & 9 & 3 & 2 & 1 \\ 3 & 8 & 9 & 5 & 5 & 1 & 4 & 2 & 1 & 3 \\ 4 & 9 & 4 & 6 & 4 & 5 & 2 & 1 & 3 & 2 \end{bmatrix},$$

$$C = \begin{bmatrix} 3 & 3 & 2 & 1 & 9 & 5 & 4 & 3 & 8 & 9 \\ 3 & 2 & 1 & 4 & 5 & 9 & 7 & 8 & 3 & 2 \\ 2 & 1 & 9 & 3 & 2 & 1 & 5 & 4 & 3 & 2 \\ 1 & 4 & 3 & 2 & 2 & 9 & 5 & 4 & 3 & 1 \\ 9 & 5 & 2 & 2 & 3 & 3 & 2 & 1 & 5 & 4 \\ 5 & 9 & 1 & 9 & 3 & 6 & 5 & 4 & 2 & 3 \\ 4 & 7 & 5 & 5 & 2 & 5 & 3 & 2 & 1 & 2 \\ 3 & 8 & 4 & 4 & 1 & 4 & 2 & 3 & 0 & 4 \\ 8 & 3 & 3 & 3 & 5 & 2 & 1 & 0 & 3 & 3 \\ 9 & 2 & 2 & 1 & 4 & 3 & 2 & 4 & 3 & 0 \end{bmatrix}.$$

Let $p = (1,1,1,1,1,1,1,1,1,1)^\top$, $k = 2$ and take initial vector $q = p/||p||_2$, the projection quadratic eigenvalue problem can be obtained by Algorithm 2. By using the polyeig function, the modulo largest eigenvalue is -6.9702×10^7, and the termination criterion is 1E-6, the residual error is 8.2930×10^{-7}. Figure 2 plots the relative residual norms for the solving method. However, by [12, Algorithm 2.1] and by polyeig function, the modulo largest eigenvalue is Inf or spill over. By polyeig(C, B, A), the modula largest eigenvalue is -6.9700×10^7.

From the example, it is indicated that when the condition number of matrix A is very large, the solution may spill over by [12, Algorithm 2.1]. In this case, we can try to utilize Algorithm 2.

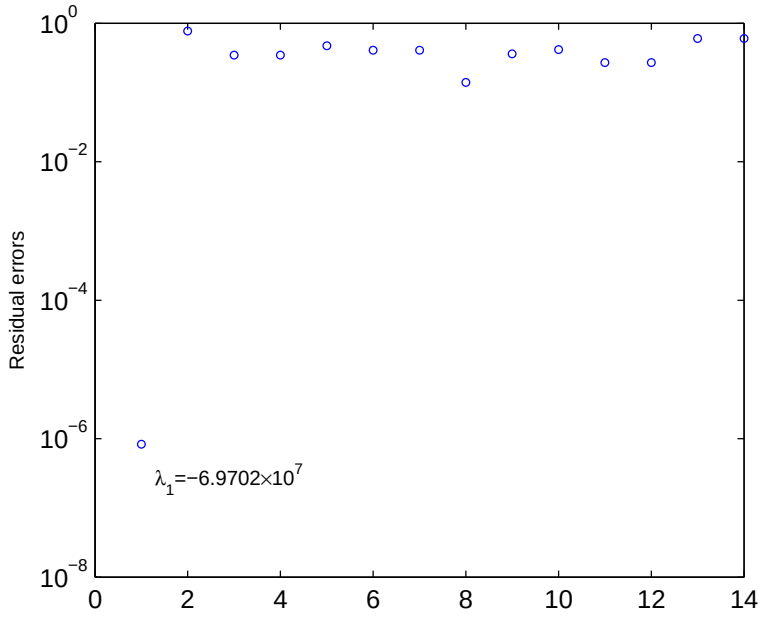

Figure 2. Residual errors of computed eigenvalues

5.2. Example. $n = 50, A_0 = rand(n)$, $\lambda_{\min}(A_0^\top A_0) = 3.3015 \times 10^{-6}$, $A = A_0^\top A_0 - 3.3014 \times 10^{-6} I_n$, $cond(A) = 5.1352 \times 10^{12}$, $B = round(80 * rand(n))$, $C = round(80 * rand(n))$.

Let $p = ones(50, 1)$, taking initial vector $q = p/||p||_2$ and $k = 3$, by utilizing Algorithm 1, the lower-dimensional quadratic eigenvalue problem can be obtained. By polyeig function, the modulo largest eigenvalue is 3.4187×10^{11}, and the termination criterion is 1E-5, the residual error is 9.1972×10^{-6}. Figure 3 plots the relative residual norms for this solving method. However, by [12, Algorithm 2.1] and or polyeig function, the modulo largest eigenvalue is Inf or spill over. Furthermore, by polyeig(C, B, A), the modula largest eigenvalue is 3.4188×10^{11}. Although the polyeig is convenient, it cannot solve large scale eigenvalue problem.

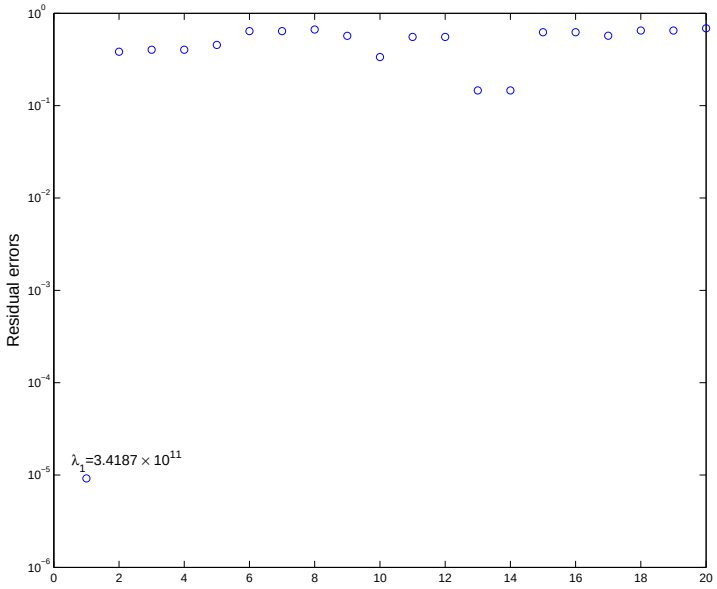

Figure 3. Residual errors of computed eigenvalues

5.3. Example. $n = 100$, $A_0 = rand(n)$, $\lambda_{\min}(A_0^\top A_0) = 2.2805 \times 10^{-5}$, $A = A_0^\top A_0 - 2.28 \times 10^{-5} I_n$, $cond(A) = 4.9641 \times 10^{11}$, $B = round(160 * rand(n))$, $C = round(160 * rand(n))$.

Let $p = ones(100, 1)$, taking initial vector $q = p/||p||_2$ and $k = 3$, by utilizing Algorithm 1, the lower-dimensional quadratic eigenvalue problem can be obtained. By polyeig function, the modulo largest eigenvalue is $2.3796 \times 10^9 - 12.764i$, and the termination criterion is 1E-5, the residual error is 9.7534×10^{-6}. Figure 4 plots the relative residual norms for this solving method. However, by [12, Algorithm 2.1] or polyeig function, the modulo largest eigenvalue is Inf or spill over. Furthermore, by polyeig(C, B, A), the modula largest eigenvalue is 2.3796×10^9.

5.4. Example. $n = 300$, $A_0 = rand(n)$, $A = A_0^\top A_0 - 9.45 \times 10^{-5} I_n$, $cond(A) = 2.4070 \times 10^{11}$, $\lambda_{\min}(A_0^\top A_0) = 9.4594 \times 10^{-5}$, $B = round(500 * rand(n))$, $C = round(500 * rand(n))$.

Let $p = ones(300, 1)$, taking initial vector $q = p/||p||_2$ and $k = 5$, by utilizing Algorithm 1, the lower-dimensional quadratic eigenvalue problem can be given. By polyeig function, the modulo largest eigenvalue is $-1.1874 \times 10^9 + 5.829i$, and the termination criterion is 1E-5, the residual error is 9.9988×10^{-6}. Figure 5 plots the relative residual norms for this solving method. However, by [12, Algorithm 2.1] or polyeig function, the modulo largest eigenvalue is Inf or spill over. Furthermore, by polyeig(C, B, A), the obtained modula largest eigenvalue is -1.1874×10^9.

5.5. Example. $n = 500$, $A_0 = rand(n)$, $\lambda_{\min}(A_0^\top A_0) = 3.9457 \times 10^{-5}$, $A = A_0^\top A_0 - 3.9 \times 10^{-5} I_n$, $cond(A) = 1.3698 \times 10^{11}$, $B = round(800 * rand(n))$, $C = round(800 * rand(n))$.

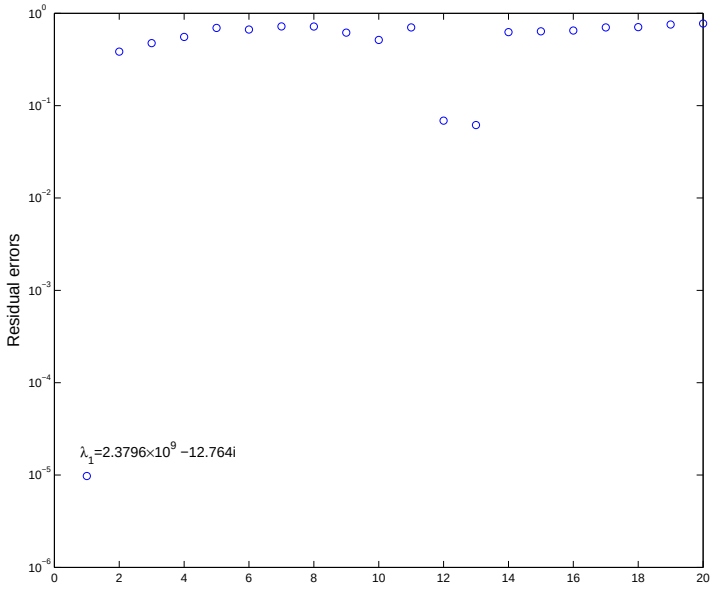

Figure 4. Residual errors of computed eigenvalues

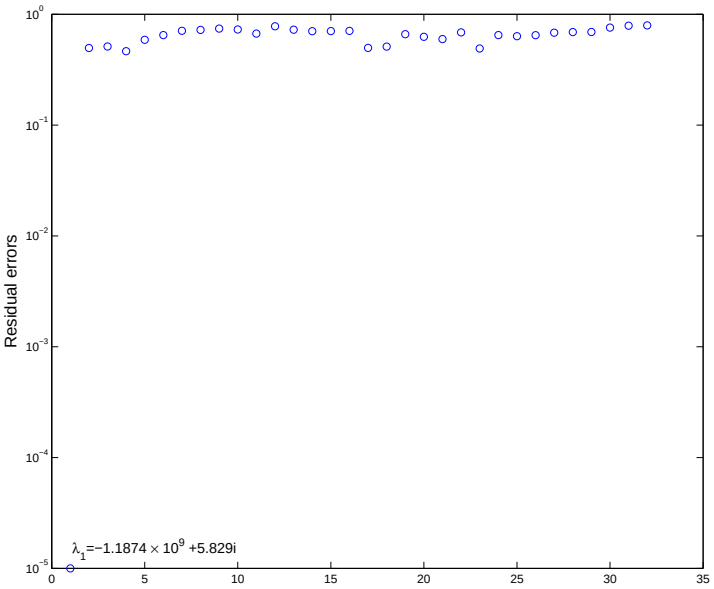

Figure 5. Residual errors of computed eigenvalues

Let $p = ones(500, 1)$, taking initial vector $q = p/||p||_2$ and $k = 8$, by utilizing Algorithm 1, the lower-dimensional quadratic eigenvalue problem can be obtained. By polyeig function, the modulo largest eigenvalue is $6.0500 \times 10^8 + 72.832i$, and the

termination criterion is 1E-5, the residual error is 9.8699×10^{-6}. Figure 6 plots the relative residual norms for this solving method. However, by [12, Algorithm 2.1] or polyeig function, the modulo largest eigenvalue is Inf or spill over. Meanwhile, by polyeig(C, B, A), the modula largest eigenvalue is 6.0500×10^8.

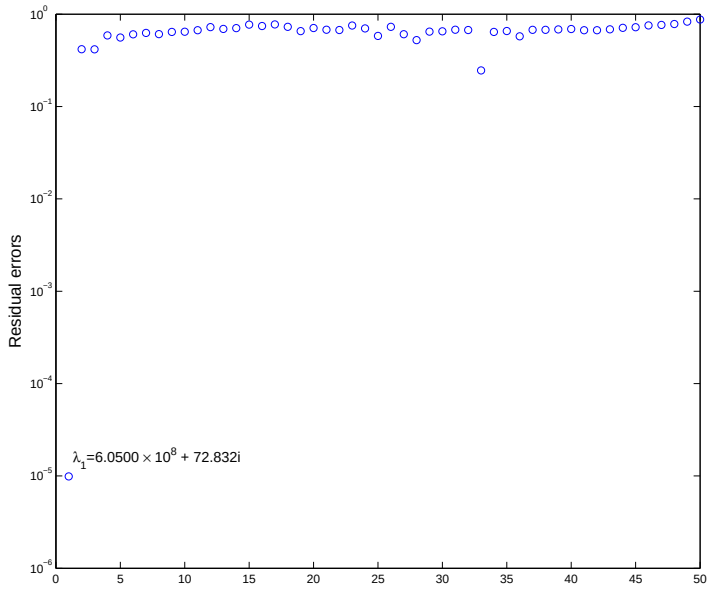

Figure 6. Residual errors of computed eigenvalues

5.6. Example. $n = 1000$, $A_0 = rand(n)$, $\lambda_{\min}(A_0^\top A_0) = 3.8766 \times 10^{-5}$, $A = A_0^\top A_0 - 3.1 \times 10^{-5} I_n$, $cond(A) = 3.2212 \times 10^{10}$, $B = round(1500 * rand(n))$, $C = round(1500 * rand(n))$.

Let $p = ones(1000, 1)$, taking initial vector $q = p/||p||_2$ and $k = 15$, by utilizing Algorithm 1, the lower-dimensional quadratic eigenvalue problem can be obtained. By polyeig function, the modulo largest eigenvalue is $-3.9828 \times 10^7 - 91.527i$, and the termination criterion is 1E-5, the residual error is 9.9989×10^{-6}. Figure 7 plots the relative residual norms for this solving method. However, by [12, Algorithm 2.1] or polyeig function, the modulo largest eigenvalue is Inf or spill over. By polyeig(C, B, A), the obtained modula largest eigenvalue is -3.9828×10^7.

References

[1] Arnoldi, W.E. *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quarterly of Applied Mathematics (9), 17-29, 1951.

[2] Z.J. Bai, Y.F. Su, *SOAR: A second-order arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM Journal on Matrix Analysis and Applications **26**(3), 640-659, 2004.

[3] M. A. Brebner, J. Grad, *Eigenvalues of $Ax = \lambda Bx$ for real symmetric matrices A and B computed by reduction to a pseudosymmetric form and the HR process*, Linear Algebra and its Applications **43** (3), 99-118, 1982.

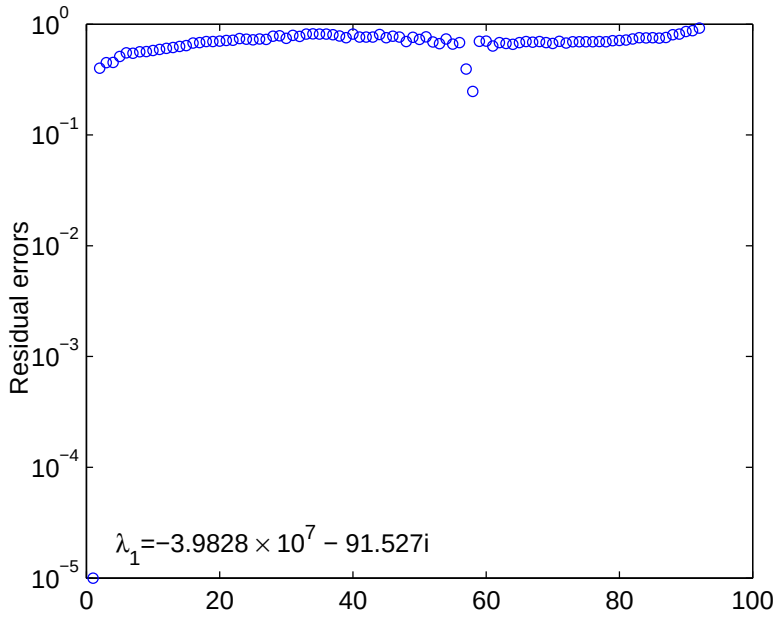

Figure 7. Residual errors of computed eigenvalues

[4] J.W. Demmel, Applied Numerical Linear Algebra, SIAM, Philadelphia, 1997.

[5] I. Gohberg, P. Lancaster, L. Rodman, Matrix Polynomials, Academic Press, New York, 1982.

[6] G.H. Golub, C.F. van Loan, Matrix Computations, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[7] L. Hoffnung, R.C. Li, Q. Ye, *Krylov type subspace methods for matrix polynomials*, Linear Algebra and its Applications **415** (1), 52-81, 2006.

[8] Z.X. Jia, *Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems*, Linear Algebra and its Applications **259** (1), 1-23, 1997.

[9] Z.X. Jia, *Using cross-product matrices to compute the SVD*, Numerical algorithms **42** (1), 31-61, 2006.

[10] Z.X. Jia, Y.Q. Sun, A refined second-order arnoldi (RSOAR) method for the quadratic eigenvalue problem and implicitly restarted algorithms, Taiwanese J. Math. Accepted, (2014)

[11] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, Journal of Research of the National Bureau Standards (45), 255-282, 1950.

[12] R.C. Li, Q. Ye, *A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems*, SIAM Journal on Matrix Analysis and Applications **25** (2), 405-428, 2003.

[13] K. Meerbergen, *The quadratic arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM Journal on Matrix Analysis and Applications **30** (4), 1463-1482, 2008.

[14] C.B. Moler, G.W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM Journal on Numerical Analysis **10** (2), 241-256, 1973.

[15] Y. Saad, Numerical Methods for Large Eigenvalue Problems, Manchester University Press, Man-chester, UK, 1992.

[16] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra and its Applications **309** (1-3), 339-361, 2000.

[17] F. Tisseur, K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Review **43** (2), 235-286, 2001.

[18] J.H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, England, 1965.