

Restarted Generalized Second-Order Krylov Subspace Methods for Solving Quadratic Eigenvalue Problems

Liping Zhou, Liang Bao, Yiqin Lin, Yimin Wei and Qinghua Wu

Abstract—This article is devoted to the numerical solution of large-scale quadratic eigenvalue problems. Such problems arise in a wide variety of applications, such as the dynamic analysis of structural mechanical systems, acoustic systems, fluid mechanics, and signal processing. We first introduce a generalized second-order Krylov subspace based on a pair of square matrices and two initial vectors and present a generalized second-order Arnoldi process for constructing an orthonormal basis of the generalized second-order Krylov subspace. Then, by using the projection technique and the refined projection technique, we propose a restarted generalized second-order Arnoldi method and a restarted refined generalized second-order Arnoldi method for computing some eigenpairs of large-scale quadratic eigenvalue problems. Some theoretical results are also presented. Some numerical examples are presented to illustrate the effectiveness of the proposed methods.

Keywords—Quadratic eigenvalue problem, Generalized second-order Krylov subspace, Generalized second-order Arnoldi process, Projection technique, Refined technique, Restarting.

I. INTRODUCTION

IN this paper, we consider the quadratic eigenvalue problem (QEP) of the form

$$(\lambda^2 M + \lambda D + K)x = 0, \quad (1)$$

where $\lambda \in \mathbb{C}$, $x \in \mathbb{C}^N \setminus \{0\}$, and $M, D, K \in \mathbb{C}^{N \times N}$. The scalar λ and the nonzero vector x are called the eigenvalue and the associated eigenvector of (1), respectively. It is well known that the QEP has $2n$ eigenvalues and $2n$ eigenvectors.

The QEP arises in a wide variety of applications, such as the dynamic analysis of structural mechanical systems [6], acoustic systems [4], the elastic deformation of anisotropic materials [35], fluid mechanics [14], microelectronic mechanical systems [5], the vibration in the structural analysis for fast trains [15], total least squares problems [22], [23], and

signal processing [8]. For a comprehensive survey of applications, spectral theory, perturbation analysis and numerical approaches for the QEP, the interesting reader is referred to [38] and the references therein.

A standard approach for solving the QEP (1) is to use the linearization in which the QEP is transformed to an equivalent $2N \times 2N$ linear eigenvalue problem. By introducing a new vector $y = \lambda x$, the QEP (1) can be linearized into a generalized eigenvalue problem of the form

$$\left(\lambda \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} -D & -K \\ I & 0 \end{bmatrix} \right) \begin{bmatrix} y \\ x \end{bmatrix} = 0, \quad (2)$$

or an equivalent standard eigenvalue problem of the form

$$\begin{bmatrix} -M^{-1}D & -M^{-1}K \\ I & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \lambda \begin{bmatrix} y \\ x \end{bmatrix}, \quad (3)$$

where we assume throughout the paper that M is nonsingular. If $(\lambda, [y^T, x^T]^T)$ is an eigenpair of (2), then x is an eigenvector of (1) associated with the eigenvalue λ . After linearization, the standard Krylov subspace projection methods such as the Arnoldi algorithm [2], [11] can be applied for solving (3). However, the linearization doubles the size of the problem, i.e., the dimension of the space that the Krylov subspace projection methods work on is twice of the original one. It could significantly increase the computational cost and memory requirements for large-scale problems. So it is not preferable to work on the linearized problem directly. The second serious drawback of working on the linearized problem directly is that even if a backward stable method is used for the linear eigenvalue problem, the stability is not guaranteed for the QEP. This is because the set of admissible perturbations for (2) is larger than for (1), which maybe makes the condition number increase, see, for example, [37].

For recent years, finding a suitable numerical method for solving the QEPs with special structure is an area of active research and has attracted considerable attention. In [13], the QEP with M, D, K real symmetric and M, K positive definite was studied. A numerical algorithm was proposed for computing a few of the smallest positive eigenvalues and their associated eigenvectors. The algorithm utilizes the symmetry and positivity of the coefficient matrices, and converges under some mild conditions. Guo [12] considered the gyroscopic QEP, where $M = M^T$, $D = -D^T$, and $K = K^T$ with M being positive definite. A solvent approach was proposed for finding all eigenvalues of the gyroscopic QEP. It has been shown that the method converges quadratically when the

L. Zhou is with the Department of Mathematics and Computational Science, Hunan University of Science and Engineering, Yongzhou 425100, P. R. China. Email: lpzhou@yahoo.com.cn.

Corresponding author. L. Bao is with the Department of Mathematics, East China University of Science and Technology, Shanghai 200237, P. R. China. Email: nlbao@yahoo.cn.

Y. Lin is with the Department of Mathematics and Computational Science, Hunan University of Science and Engineering, Yongzhou 425100, P. R. China. Email: yqlin@yahoo.cn.

Y. Wei is with the School of Mathematical Sciences, Fudan University, Shanghai 200433, P. R. China and Key Laboratory of Mathematics for Nonlinear Sciences (Fudan University), Ministry of Education. Email: ymwei@fudan.edu.cn.

Q. Wu is with the Department of Mathematics and Computational Science, Hunan University of Science and Engineering, Yongzhou 425100, P. R. China. Email: qhuawu@yahoo.cn.

QEP has no eigenvalues on the imaginary axis, and converges linearly when the QEP has purely imaginary eigenvalues with even partial multiplicities. This approach requires $\mathcal{O}(N^3)$ flops and $\mathcal{O}(N^2)$ memory, and therefore is only practicable for problems of relatively small size. To overcome the slow convergence in the presence of purely imaginary eigenvalues, Qian and Lin [28] presented a new method, which has the quadratic convergence rate for any gyroscopic QEP. In their method, purely imaginary eigenvalues are firstly shifted to eigenvalues with nonzero real parts, while other eigenpairs are kept unchanged, to generate a new gyroscopic QEP. Then, the solvent approach in [12] is applied to the new gyroscopic QEP. In [19], a structure-preserving algorithm was proposed for solving the palindromic QEP, where $K = M^T$ and $D = D^T$. Moreover, a generalized T-skew-Hamiltonian implicitly restarted shift-and-invert Arnoldi algorithm was also developed for finding the eigenvalues of the large and sparse palindromic QEP in a specified region. Another class of QEPs arising from some practical applications was studied in [18]. A novel method was designed by combining a structure-preserving method and a quadratic Jacobi-Davidson method.

A number of numerical methods, which are applied to large-scale QEPs directly, have been proposed for computing a few eigenvalues and the corresponding eigenvectors for general QEPs in the literature. In these methods, they do not transform the QEP to an equivalent linear form; instead, they project the QEP onto a properly chosen low-dimensional subspace to reduce to a QEP with lower order. The reduced QEP is then solved by standard dense matrix techniques such as the QR method for linear eigenvalue problems or the QZ method for generalized linear eigenvalue problems [2], [9]. The remarkable feature in practice is that these methods are applied directly to solve the original problem, and so the essential structures of M , D , K as well as the spectral properties are preserved promisingly. In [36], a quadratic Jacobi-Davidson type method was designed and applied for the solution of quadratic eigenvalue problems arising from acoustic problems. However, the method targets at one eigenvalue at one time with only local convergence. A direct Krylov-type subspace method with a generalized Arnoldi process was briefly described in [29]. A reformulated Arnoldi algorithm for non-classically damped eigenvalue problems was presented in [30]. Meerbergen [26] proposed a quadratic residual iterative method with a locking and restarting scheme. Li and Ye [24] presented a Krylov subspace method based on an Arnoldi-type process for the QEP arising in the quadratically constrained least squares problem. In [17], an approximation method based on perturbation subspaces for block eigenvector matrices was proposed. These perturbation subspaces are contained in certain generalized Krylov subspaces. Ye [40] presented a shift-and-invert Arnoldi algorithm and discussed its inexact shift-and-invert variant. Bai and Su [3] proposed a second-order Arnoldi (SOAR for short) method by using projection technique for finding some eigenvalues and eigenvectors of the large-scale QEP. In [27], a quadratic Arnoldi algorithm was presented. In fact, it is an Arnoldi method applied to linear eigenvalue problem (3). However, by exploiting the structure of the Krylov vectors, the memory requirements are reduced

by about a half. In contrast to the quadratic Jacobi-Davidson type method, subspace iterative methods approximate a group of eigenvalues simultaneously and have global convergence. We note that while these methods use a similar projection process, they differ in the subspaces that are constructed and used for projection.

The SOAR algorithm presented in [3] is stable, but both the computational cost and the storage increase as the method proceeds. To overcome this drawback of the the SOAR algorithm, in this paper, we propose restarted SOAR methods for computing some eigenvalues and their corresponding eigenvectors of the QEP (1). That is, for given projection subspaces, if the SOAR method do not compute the desired eigenpairs with prescribed accuracy, then one chooses new better starting vectors, constructs new better subspaces, and compute approximate eigenpairs until they converge. The restarted SOAR methods in this paper are also developed under the framework of projection directly based on (1). The projection subspaces are the generalized second-order Krylov subspaces. These two restarted methods are applied to solve the large-scale quadratic eigenvalue problem directly. Hence they preserve essential structures and properties of the large-scale quadratic eigenvalue problem.

Throughout this paper, the following notation is used. I denotes the identity matrix, e_j denotes the j -th column of the identity matrix I , and 0 denotes the zero vector or zero matrix. The actual dimension of I and 0 will always be apparent from the context. The superscripts T and $*$ denote the transpose and the conjugate transpose of a vector or a matrix, respectively. We denote 1-norm and 2-norm by $\|\cdot\|_1$ and $\|\cdot\|_2$, respectively, for a vector or a matrix. The notation $\text{span}\{V\}$ denotes the space spanned by the column vectors of the matrix V and $\text{span}\{v_1, v_2, \dots, v_n\}$ denotes the space spanned by the vector sequence v_1, v_2, \dots, v_n . Finally, Matlab [25] notation is used whenever possible.

The remainder of the paper is organized as follows. In Section 2, we first introduce the generalized second-order Krylov subspace. Then, the generalized second-order Arnoldi process for constructing an orthonormal basis of the subspace is described and some implementation issues are addressed. In Section 3, we present a restarted generalized second-order Arnoldi method and a restarted refined generalized second-order Arnoldi method for solving the large-scale QEP (1). In Section 4, some numerical examples are presented to illustrate the advantage of the proposed methods. Some concluding remarks are given in last section.

II. THE GENERALIZED SECOND-ORDER KRYLOV SUBSPACE AND THE GENERALIZED SECOND-ORDER ARNOLDI PROCESS

In [3], Bai and Su gave the following definition.

Definition 2.1: Let A and B be square matrices of order N , and $u_1 \in \mathbb{C}^N$ be a nonzero vector. Then the sequence

$$r_0, r_1, r_2, \dots, r_{n-1},$$

where

$$\begin{aligned} r_0 &= u_1, \\ r_1 &= Au_1, \\ r_j &= Ar_{j-1} + Br_{j-2} \quad \text{for } j \geq 2, \end{aligned}$$

is called a second-order Krylov sequence based on A , B and u_1 . The space

$$\mathcal{G}_n(A, B; u_1) = \text{span}\{r_0, r_1, r_2, \dots, r_{n-1}\},$$

is called an n -th second-order Krylov subspace.

To generate an orthonormal basis of the second-order Krylov subspace $\mathcal{G}_n(A, B; u_1)$, Bai and Su proposed a second-order Arnoldi process, see [3] for details.

We extend their idea to obtain a generalized second-order Krylov subspace, which is defined as follows.

Definition 2.2: Let A and B be square matrices of order N , and $u_1 \neq 0$ and u_2 be two N -dimensional vectors. Then the sequence

$$r_0, r_1, r_2, \dots, r_{n-1}, \quad (4)$$

where

$$\begin{aligned} r_0 &= u_1, \\ r_1 &= Au_1 + Bu_2, \\ r_j &= Ar_{j-1} + Br_{j-2} \quad \text{for } j \geq 2, \end{aligned}$$

is called a generalized second-order Krylov sequence based on A , B and u_1, u_2 . The space

$$\mathcal{G}_n(A, B; u_1, u_2) = \text{span}\{r_0, r_1, r_2, \dots, r_{n-1}\},$$

is called an n -th generalized second-order Krylov subspace.

We note that the generalized second-order Krylov subspace $\mathcal{G}_n(A, B; u_1, u_2)$ generalizes the standard second-order Krylov subspace $\mathcal{G}_n(A, B; u_1)$ in the way when u_2 is a zero vector, the generalized second-order Krylov subspace is the standard second-order Krylov subspace, namely,

$$\mathcal{G}_n(A, B; u_1, 0) = \mathcal{G}_n(A, B; u_1).$$

By introducing the vector u_2 , we can apply the restarted strategy with the starting vector $v = [u_1^T, u_2^T]^T$, as used in the following section.

Let

$$F = \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}.$$

The Krylov subspace associated with the matrix F and the vector v is

$$\mathcal{K}_n(F; v) = \text{span}\{v, Fv, F^2v, \dots, F^{n-1}v\}. \quad (5)$$

It is easy to verify that the generalized second-order Krylov vector sequence $\{r_j\}$ and the standard Krylov vector sequence $\{F^j v\}$ are related as the following form

$$\begin{bmatrix} r_j \\ r_{j-1} \end{bmatrix} = F^j v \quad \text{for } j \geq 1. \quad (6)$$

Now, we propose an algorithm to establish an orthonormal basis q_1, q_2, \dots, q_n of the generalized second-order Krylov subspace $\mathcal{G}_n(A, B; u_1, u_2)$. We call it a GSOAR (Generalized Second-Order ARnoldi) process. The algorithm is described as follows.

Algorithm 2.1: GSOAR process

1. $q_1 = u_1 / \|u_1\|_2$;
2. $p_1 = u_2 / \|u_2\|_2$;
3. for $j = 1, 2, \dots, n$
4. $r = Aq_j + Bp_j$;
5. $s = q_j$;
6. for $i = 1, 2, \dots, j$
7. $h_{i,j} = q_i^* r$;
8. $r := r - q_i h_{i,j}$;
9. $s := s - p_i h_{i,j}$;
10. end for
11. $h_{j+1,j} = \|r\|_2$;
12. if $h_{j+1,j} = 0$, then stop
13. $q_{j+1} = r / h_{j+1,j}$;
14. $p_{j+1} = s / h_{j+1,j}$;
15. end for

About Algorithm 2.1, a few remarks are in order:

- The GSOAR process is the same as the SOAR process in [3] except that at Step 2 in the SOAR process, the vector p_1 is always set to a zero vector.
- Algorithm 2.1 is known as an implementation of orthogonalize the generalized second-order Krylov subspace $\mathcal{G}_n(A, B; u_1, u_2)$ in the modified Gram-Schmidt orthogonalization form. It is well known that in the presence of finite precision arithmetic, a loss of orthogonality can occur when the orthogonalization algorithm progresses, see [9], [11], [31]. A remedy is the so-called reorthogonalization where the current vectors have to be orthogonalized against previously created vectors. One can choose between a selective reorthogonalization or a full reorthogonalization. For another proven stable method for generating an orthonormal basis, we refer to [7].

Let \tilde{H}_n denote the $(n+1) \times n$ upper Hessenberg matrix whose nonzero entries $h_{i,j}, i = 1, 2, \dots, n+1$ and $j = 1, 2, \dots, n$ are defined by Algorithm 2.1. Let H_n be the $n \times n$ matrix obtained from \tilde{H}_n by deleting the last row. Define $Q_n = [q_1, q_2, \dots, q_n]$ and $P_n = [p_1, p_2, \dots, p_n]$. The GSOAR process can be written in matrix form

$$AQ_n + BP_n = Q_n H_n + q_{n+1} e_n^T h_{n+1,n}, \quad (7)$$

$$Q_n = P_n H_n + p_{n+1} e_n^T h_{n+1,n}. \quad (8)$$

We can also rewrite (7) and (8) in the more compact form

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \begin{bmatrix} Q_n \\ P_n \end{bmatrix} = \begin{bmatrix} Q_{n+1} \\ P_{n+1} \end{bmatrix} \tilde{H}_n. \quad (9)$$

This relation assembles the similarity between the GSOAR process and the well-known Arnoldi process [1].

It is not difficult to verify the following theorem, which shows that the vector sequence $\{q_1, q_2, \dots, q_n\}$ is an orthonormal basis of the generalized second-order Krylov subspace $\mathcal{G}_n(A, B; u_1, u_2)$.

Theorem 2.3: If $h_{j+1,j} \neq 0$ for $j \geq 1$ in Algorithm 2.1, then the vector sequence $\{q_1, q_2, \dots, q_j\}$ forms an orthonormal basis of the generalized second-order Krylov subspace $\mathcal{G}_j(A, B; u_1, u_2)$, i.e.,

$$\text{span}\{q_1, q_2, \dots, q_j\} = \mathcal{G}_j(A, B; u_1, u_2) \quad \text{for } j \geq 1, \quad (10)$$

and $q_i^T q_k = 0$ if $i \neq k$ and $q_i^T q_i = 1$ for $i, k = 1, 2, \dots, j$.

In the rest of this section, we derive a new version of the GSOAR process by using the relations presented above.

Partition P_{n+1} as $P_{n+1} = [p_1, \tilde{P}]$. It follows from (8) that

$$Q_n = P_{n+1} \tilde{H}_n = p_1 \cdot \tilde{H}_n(1, :) + \tilde{P} \cdot \tilde{H}_n(2 : n+1, :),$$

which shows that

$$p_{n+1} = \tilde{P} e_n = (Q_n - p_1 \cdot \tilde{H}_n(1, :))(\tilde{H}_n(2 : n+1, :))^{-1}.$$

So we can compute the vector p_{j+1} from Q_j and $H(1 : j+1, 1 : j)$ and avoid the explicit references and updates of the p -vectors at lines 9 and 14 of Algorithm 2.1. It makes the memory requirements reduced by about a half.

In summary, we obtain a new algorithm, which only needs about a half of the memory space and floating point operations of Algorithm 2.1. The new algorithm is outlined as follows.

Algorithm 2.2: GSOAR process with memory saving

1. $q_1 = u_1 / \|u_1\|_2$;
2. $f = u_2$;
3. for $j = 1, 2, \dots, n$
4. $r = Aq_j + Bf$;
5. for $i = 1, 2, \dots, j$
6. $h_{i,j} = q_i^T r$;
7. $r := r - q_i h_{i,j}$;
8. end for
9. $h_{j+1,j} = \|r\|_2$;
10. if $h_{j+1,j} = 0$, then terminate;
11. $q_{j+1} = r / h_{j+1,j}$;
12. $f = (Q_j - u_2 H(1, 1 : j)) H(2 : j+1, 1 : j)^{-1} e_j$.
13. end for

III. THE RESTARTED GENERALIZED SECOND-ORDER ARNOLDI METHOD AND THE RESTARTED REFINED GENERALIZED SECOND-ORDER ARNOLDI METHOD

Let $A = -M^{-1}D$ and $B = -M^{-1}K$. To obtain an approximate eigenpair (θ, z) , where $\theta \in \mathbb{C}$ and $z \in \mathcal{G}_n(A, B; u_1, u_2)$, we impose the Galerkin condition

$$(\theta^2 M + \theta D + K)z \perp \mathcal{G}_n(A, B; u_1, u_2),$$

i.e.,

$$v^*(\theta^2 M + \theta D + K)z = 0 \quad \text{for all } v \in \mathcal{G}_n(A, B; u_1, u_2). \quad (11)$$

Let the columns of the $N \times n$ matrix Q_n form an orthonormal basis of $\mathcal{G}_n(A, B; u_1, u_2)$. It follows from (11) that

$$(\theta^2 M_n + \theta D_n + K_n)g = 0, \quad (12)$$

where g is an n -dimensional vector and

$$M_n = Q_n^* M Q_n, \quad D_n = Q_n^* D Q_n, \quad K_n = Q_n^* K Q_n. \quad (13)$$

The eigenvalues of the reduced QEP (12) are called Ritz values of the original QEP (1) with respect to the generalized second-order Krylov subspace $\mathcal{G}_n(A, B; u_1, u_2)$. If g is an eigenvector of the reduced QEP (12), then $z = Q_n g$ is called a Ritz vector of the QEP (1).

In summary, we have a method for solving the QEP (1), which is similar to the SOAR method proposed in [3].

Algorithm 3.1: The GSOAR method for solving the QEP

1. Choose two starting vectors u_1 and u_2 .
2. Run GSOAR process (Algorithm 2.1) with $A = -M^{-1}D$ and $B = -M^{-1}K$ to generate an $N \times n$ orthogonal matrix Q_n whose columns span an orthonormal basis of $\mathcal{G}_n(A, B; u_1, u_2)$.
3. Compute M_n , D_n and K_n as defined in (13).
4. Solve the reduced QEP (12) for (θ, g) and obtain the Ritz pairs (θ, z) , where $z = Q_n g / \|Q_n g\|_2$.
5. Test the accuracy of Ritz pairs (θ, z) as approximate eigenvalues and eigenvectors of the QEP (1) by the relative norms of residual vectors:

$$\frac{\|(\theta^2 M + \theta D + K)z\|_2}{|\theta|^2 \|M\|_1 + |\theta| \|D\|_1 + \|K\|_1}. \quad (14)$$

About Algorithm 3.1, a few remarks are in order:

- At Step 2, The product of M^{-1} with some vector should be implemented by solving the linear systems of equations with the coefficient matrix M . To do it, the LU factorization [11] of M is employed for medium-size matrices, and the Cholesky factorization of M should be used for M symmetric definite. For large-scale matrices, a preconditioning iterative method could be employed to solve systems with M , where the preconditioner could be generated once for all. Iterative methods that are used nowadays are Krylov subspace methods such as GMRES [33], QMR [10], and BICGSTAB [39]. For a comprehensive introduction of iterative methods for linear systems of equations, the interesting reader is referred to [34]. Note that when the inexact iterative solvers are used for the linear systems of equations with the coefficient matrix M , the subspace $\text{span}\{q_1, q_2, \dots, q_n\}$ generated by Algorithm 2.1 is no longer the generalized second-order Krylov subspace $\mathcal{G}_n(A, B; u_1, u_2)$.
- At Step 4, to solve the small QEP (12), we transform it to a generalized eigenvalue problem in the form of (2), and then use a dense matrix method, such as the QZ algorithm [9], [11], to find all eigenvalues and eigenvectors (θ, g) of the small QEP.

A difficulty with the GSOAR method is that it becomes increasingly expensive as the number n increases. To remedy this difficulty, we can use the algorithm iteratively, i.e., we can restart the algorithm every n steps, where n is some fixed integer parameter. After obtaining k desired Ritz pairs $(\theta_i, z_i), i = 1, 2, \dots, k$ for QEP (1), we can view $(\theta_i, [\theta_i z_i^T, z_i^T]^T)$ as the approximate eigenpairs of the linear eigenvalue problem (3) since the QEP (1) is equivalent to the linear eigenvalue problem (3). For the linear eigenvalue problem, Saad [32] suggested that the new starting vector in restarted methods should be set to a weighted combination of the desired approximate vectors, and the desired approximate vectors are weighted by the corresponding relative residual norms. In the restarted GSOAR, we generated the new starting vector by this technique.

This restarted version of GSOAR denoted by GSOAR(n) is described as follows.

Algorithm 3.2: GSOAR(n): Restarted GSOAR method

1. Choose n , the maximum size of the subspace, and k , the desired number of approximate eigenpairs. Choose two starting vectors u_1 and u_2 .
2. Run GSOAR process (Algorithm 2.1) with $A = -M^{-1}D$ and $B = -M^{-1}K$ to generate an $N \times n$ orthogonal matrix Q_n whose columns span an orthonormal basis of $\mathcal{G}_n(A, B; u_1, u_2)$.
3. Compute M_n , D_n and K_n as defined in (13).
4. Solve the reduced QEP (12) for (θ_i, g_i) and obtain the Ritz pairs (θ_i, z_i) , where $z_i = Q_n g_i / \|Q_n g_i\|_2$ with $i = 1, 2, \dots, k$.
5. Test the accuracy of Ritz pairs (θ_i, z_i) as approximate eigenvalues and eigenvectors of the QEP (1) by the relative norms of residual vectors:

$$\alpha_i := \frac{\|(\theta_i^2 M + \theta_i D + K)z_i\|_2}{\|\theta_i\|^2 \|M\|_1 + \|\theta_i\| \|D\|_1 + \|K\|_1}. \quad (15)$$

If it is satisfied then stop, otherwise compute new starting vectors u_1 and u_2 by

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \sum_{i=1}^k \alpha_i \begin{bmatrix} \theta_i z_i \\ z_i \end{bmatrix}$$

and go to 2.

For the linear eigenvalue problem, it has been revealed that the standard projection methods may converge very slowly and even may fail to converge. In order to correct this problem, Jia [20], [21] proposed the refined technique. In the rest of this section, we extend his idea to obtain the restarted refined generalized second-order Arnoldi method for solving the QEP (1).

After obtaining k desired Ritz values $\theta_i, i = 1, 2, \dots, k$ for QEP (1), we now seek k unit length vectors $\hat{z}_i = Q_n \hat{g}_i$, which are called refined vectors [16] and satisfy

$$\begin{aligned} & \|(\theta_i^2 M + \theta_i D + K)\hat{z}_i\|_2 \\ &= \min_{z \in \mathcal{G}_n(A, B; u_1, u_2), \|z\|_2=1} \|(\theta_i^2 M + \theta_i D + K)z\|_2 \\ &= \min_{g \in \mathbb{C}^n, \|g\|_2=1} \|(\theta_i^2 M + \theta_i D + K)Q_n g\|_2 \\ &= \min_{g \in \mathbb{C}^n, \|g\|_2=1} \|(\theta_i^2 M Q_n + \theta_i D Q_n + K Q_n)g\|_2. \end{aligned}$$

It is easy to see that \hat{g}_i is nothing but the right singular vector associated with the smallest singular value of the $N \times n$ matrix

$$T_i := \theta_i^2 M Q_n + \theta_i D Q_n + K Q_n. \quad (16)$$

This approximate eigenpair (θ_i, \hat{z}_i) , called the refined Ritz pair, is better than the Ritz pair (θ_i, z_i) due to its minimal property. Thus, we can propose the restarted refined version of GSOAR for solving the QEP, which is denoted by RGSOAR(n) and described as follows.

Algorithm 3.3: RGSOAR(n): Restarted refined GSOAR method

1. Choose n , the maximum size of the subspace, and k , the desired number of approximate eigenpairs. Choose two starting vectors u_1 and u_2 .
2. Run GSOAR process (Algorithm 2.1) with $A = -M^{-1}D$ and $B = -M^{-1}K$ to generate an $N \times n$ orthogonal

matrix Q_n whose columns span an orthonormal basis of $\mathcal{G}_n(A, B; u_1, u_2)$.

3. Compute M_n , D_n and K_n as defined in (13).
4. Solve the reduced QEP (12) and obtain the desired Ritz values $\theta_i, i = 1, 2, \dots, k$. Compute the right singular vector \hat{g}_i associated with the smallest singular value of the matrix T_i defined by (16), $i = 1, 2, \dots, k$ and obtain k desired approximate eigenpairs $(\theta_i, \hat{z}_i = Q_n \hat{g}_i)$.
5. Test the accuracy of refined Ritz pairs (θ_i, \hat{z}_i) as approximate eigenvalues and eigenvectors of the QEP (1) by the relative norms of residual vectors:

$$\alpha_i := \frac{\|(\theta_i^2 M + \theta_i D + K)\hat{z}_i\|_2}{\|\theta_i\|^2 \|M\|_1 + \|\theta_i\| \|D\|_1 + \|K\|_1}.$$

If it is satisfied then terminate, otherwise compute new starting vectors u_1 and u_2 by

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \sum_{i=1}^k \alpha_i \begin{bmatrix} \theta_i \hat{z}_i \\ \hat{z}_i \end{bmatrix}$$

and go to 2.

At Step 4 in Algorithm 3.3, to generate $T_i, i = 1, 2, \dots, k$, we need to compute $M Q_n$, $D Q_n$ and $K Q_n$ only one time. However, we note that $M Q_n$, $D Q_n$ and $K Q_n$ have been already obtained when M_n , D_n and K_n are computed at Step 3. Thus, we should save $M Q_n$, $D Q_n$ and $K Q_n$ at Step 3 in order to save the expense. In order to obtain the right singular vector \hat{g}_i associated with the smallest singular value of the matrix T_i , we can compute the eigenvector associated with the smallest eigenvalue of $T_i^T T_i$.

We call the iterations between restarts a “cycle”. For operation requirements we list in Table I the major work, at each cycle, used for RGSOAR(n) and the restarted Quadratic Residual Iteration (QRI(n) for short) methods[26].

About Table I, a few remarks are in order:

- In the QRI(n) method, we need to solve a linear system of order N by the GMRES method [33]. We set the dimension of the projection subspace used by GMRES method equals m .
- The matrix-vector product operations for RGSOAR(n) method means that the product of the matrix $A = -M^{-1}D$ and $B = -M^{-1}K$ with some vectors while the matrix-vector product operations for QRI(n) means that the product of the matrix M , D and K with some vectors. In practice, the matrix M is often sparse, and we suppose that the LU factorization process for M computes a sparse lower triangular matrix L and a sparse upper triangular matrix U . Hence the flops of matrix-vector product operations for matrix A and B is almost the same as that for the matrix M , D and K .
- From Table I, we can see that at each cycle, the costs for QRI(n) method is much more than that for RGSOAR(n) method, but from the numerical tests, we will illustrate that the cycle number of QRI(n) is much less than that of RGSOAR(n). Hence the total costs for the two methods is comparable.

Let us review the basic restart Arnoldi method for solving the QEP (1) based on linearization (3). We will compare the

TABLE I
COSTS FOR RGSOAR(n) AND QRI(n) METHODS

Cost	RGSOAR(n)	QRI(n)
Matrix-vector	$5n$	$mn + \frac{3(n+1)n}{2}$
N -vector DOT	$3n^2 + \frac{(n+1)n}{2}$	$(n+1)^2n + \frac{(m^2+m)n}{2}$
N -vector SAXPY	$\frac{(n+1)n}{2}$	$\frac{(m^2+m+n+1)n}{2}$

GSOAR(n) method and RGSOAR(n) method with the following simple implementation of the restarted Arnoldi method for solving the QEP via linearization.

Algorithm 3.4: Arnoldi(n): Restarted Arnoldi method

1. Choose n , the maximum size of the subspace, and k , the desired number of approximate eigenpairs. Choose a starting vector $v = [u_1^T, u_2^T]^T$.
2. Run the well-known Arnoldi process [1] with the matrix F and the vector v to generate an orthonormal basis $\{v_1, v_2, \dots, v_n\}$ of the Krylov subspace $\mathcal{K}_n(F; v)$. Let $V_n = [v_1, v_2, \dots, v_n]$.
3. Solve the reduced eigenvalue problem

$$(V_n^* F V_n)g = \theta g,$$

and obtain the Ritz pairs (θ, y) of the eigenvalue problem of the single matrix F , where $y = V_n g$.

4. Extract the approximate eigenpairs (θ, z) of the QEP (1), and test their accuracy by the residual norms as described in (15), where $z = y(N+1 : 2N)/\|y(N+1 : 2N)\|_2$. If it is satisfied then terminate, otherwise compute a new starting vector v [32] and go to 2.

IV. NUMERICAL EXPERIMENTS

In this section, we present some numerical examples to illustrate the effectiveness of the GSOAR(n) method and the RGSOAR(n) method for solving the QEP (1).

In the following examples, the starting vector u_1 of the GSOAR(n) method and RGSOAR(n) method is chosen as a vector with all components equal to 1 and $u_2 = 0$. The starting vector v of the Arnoldi(n) method is chosen as $v = [u_1^T, u_2^T]^T$. The so-called exact eigenvalues of the QEP are computed by the dense method, namely, the QZ method for computing all eigenvalues and eigenvectors of the generalized eigenvalue problem (2).

All numerical experiments are performed on an AMD 1.4 GHz PC with main memory 512 MB. The stop criterion is

$$\max_{1 \leq i \leq k} \left(\frac{\|(\theta_i^2 M + \theta_i D + K)z_i\|_2}{|\theta_i|^2 \|M\|_1 + |\theta_i| \|D\|_1 + \|K\|_1} \right) \leq 10^{-8},$$

where k is the number of the desired eigenvalues.

Example 1. For the first test, we compare the convergence result of SOAR method and RGSOAR(n) method. Let M , D and K be 200×200 random nonsymmetric matrices. Elements of these matrices are chosen from a normal distribution with mean zero, variance one and standard deviation one. The

starting vector u of the SOAR method is chosen as a vector with all components equal to 1.

The largest magnitude eigenvalue computed by the standard dense matrix method (for all eigenvalues), and by RGSOAR(n) with $n = 10$ and SOAR methods are

$$\begin{aligned} \lambda_{\max} &= 8.21407509907045 + 12.26001479975082i, \\ \lambda_{\max}^R &= \underline{8.21407509903629} + \underline{12.26001479974864i}, \\ \lambda_{\max}^S &= \underline{8.21407520761109} + \underline{12.26001461539936i}, \end{aligned}$$

where n is the reduced dimension of the SOAR method and $i = \sqrt{-1}$.

We observe that both RGSOAR(n) and SOAR methods converge to the largest magnitude eigenvalue. The relative errors are $|\lambda_{\max}^R - \lambda_{\max}|/|\lambda_{\max}| = 2.32 \times 10^{-12}$ and $|\lambda_{\max}^S - \lambda_{\max}|/|\lambda_{\max}| = 1.45 \times 10^{-8}$, respectively. The largest magnitude eigenvalues produced by RGSOAR(n) method is more accurate than the SOAR method.

Example 2. For this example, we compare the convergence results of GSOAR(n) method and RGSOAR(n) method. M , D and K are also 200×200 random nonsymmetric matrices. Elements of these matrices are chosen from a normal distribution with mean zero, variance one and standard deviation one. $n = 10$ and $k = 2$. The plot of Figure 1 shows the relative residual norms of the first largest magnitude approximate eigenpair and the plot of Figure 2 shows the relative residual norms of the second largest magnitude approximate eigenpair.

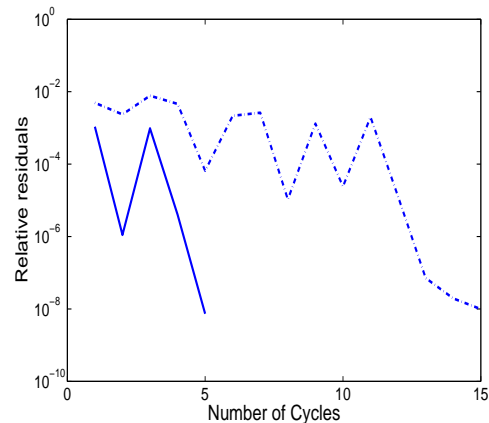


Fig. 1. Convergence plots of GSOAR(10)(dash-dot line) and RGSOAR(10) (solid line).

Figure 1 shows the cycle number of RGSOAR(10) is 5 and the cycle number of GSOAR(10) is 15. This example shows that the convergence rate of RGSOAR(10) is fairly rapid.

Example 3. For the third experiment, we use some structural engineering matrices from the Harwell-Boeing collection to compare GSOAR(n) method, RGSOAR(n) method and Arnoldi(n) method. These matrices all represent dynamic analysis in structural engineering. $M = 3 * I$. The data of matrices D and K are extracted from bcsstm06 and bcsstk06, respectively. These matrices are 420×420 . $n = 10$ and $k = 2$. The plot of Figure 3 shows the relative residual

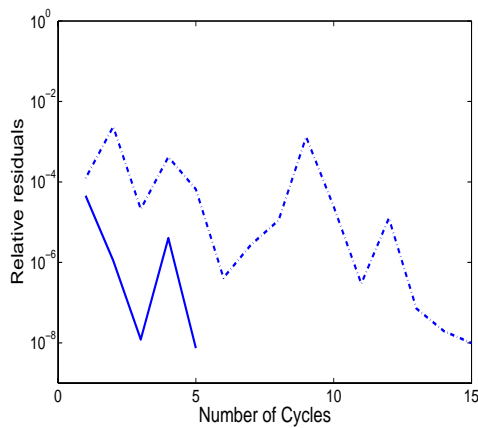


Fig. 2. Convergence plots of GSOAR(10)(dash-dot line) and RGSOAR(10) (solid line).

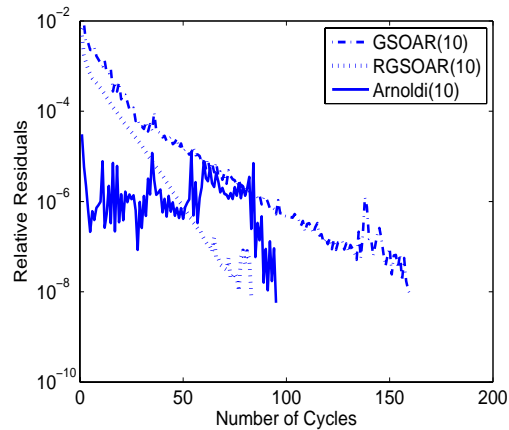


Fig. 4. Convergence plots of GSOAR(10), RGSOAR(10) and Arnoldi(10).

norms of the first largest magnitude approximate eigenpair and the plot of Figure 4 shows the relative residual norms of the second largest magnitude approximate eigenpair.

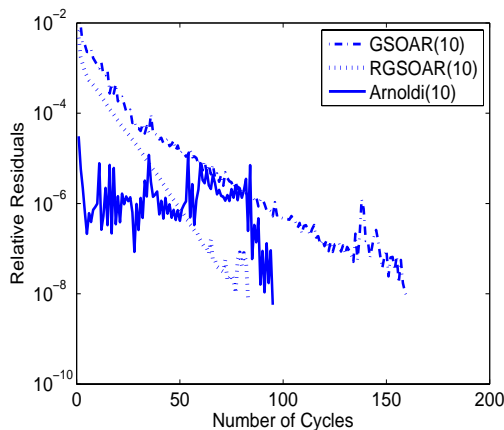


Fig. 3. Convergence plots of GSOAR(10), RGSOAR(10) and Arnoldi(10).

From Figure 3, we know the cycle number of RGSOAR(10) is 83, the cycle number of GSOAR(10) is 160 and the cycle number of Arnoldi(10) is 95, which shows that RGSOAR(10) converges faster than Arnoldi(10) for this example.

We also test this example by QRI(20) method, but we find that QRI(20) method didn't converge for this example.

Example 4. In this example, $M = 5 * I$, $D = 3 * \text{tridiag}(-1, 3, -1)$ and K comes from the Matlab test matrix *bwm200*. These matrices are 200×200 . We only compute the largest magnitude eigenvalue, i.e., $k = 1$. The plot of Figure 5 shows the relative residual norms of the largest magnitude approximate eigenpair with $n = 20$. The cycle number and cputime for RGSOAR(20) are 77 and 5.68, respectively and the cycle number and cputime for GSOAR(20) are 106 and 6.52, respectively. However, Arnoldi(20) doesn't converge after 200 cycles.

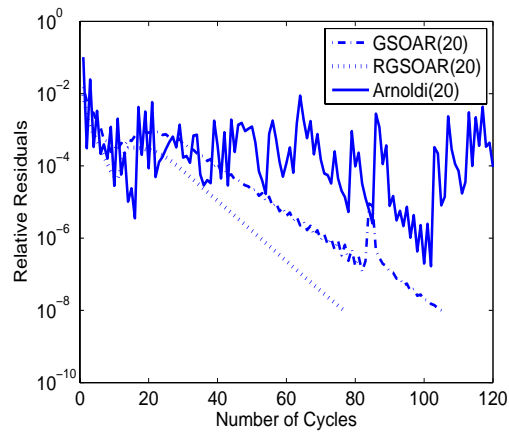


Fig. 5. Convergence plots of GSOAR(20), RGSOAR(20) and Arnoldi(20).

We also test this example by QRI(n) method and the cycle number for QRI(20) is 7. In Table II, we list the total operation requirements for RGSOAR(20) and QRI(20) methods.

$$\text{Total costs} = \text{Costs at each cycle} \times \text{Cycle number.}$$

It is clear to see from Table II that the major costs used for the two methods is comparable.

V. CONCLUSIONS AND OUTLOOK

We propose in this paper a restarted generalized second-order Arnoldi method and a restarted refined generalized second-order Arnoldi method for solving a large-scale

TABLE II
TOTAL COSTS FOR RGSOAR(20) AND QRI(20) METHODS

Cost	RGSOAR(20)	QRI(20)
Matrix-vector	7700	7210
N -vector DOT	108570	91140
N -vector SAXPY	16170	30870

quadratic eigenvalue problem. Approximations to several eigenvalues can be found at the same time. Even though some information is discarded because of restarting, the most important information is retained. The methods reduce the large-scale quadratic eigenvalue problem to a QEP of smaller size by applying the projection technique and also reduce the storage costs by restarting. Numerical tests presented in this paper show the effectiveness of the proposed methods. How to implement the implicitly restarting strategy, remains the subject of further research.

ACKNOWLEDGEMENTS

Yiqin Lin is supported by the National Natural Science Foundation of China under grant 10801048 and the Natural Science Foundation of Hunan Province under grant 09JJ6014. Yimin Wei is supported by the National Natural Science Foundation of China under grant 10871051, Shanghai Education Committee under grant 08SG01, Shanghai Science and Technology Committee under grant 08511501703, and 973 Program Project under grant 2010CB327900.

REFERENCES

- [1] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9(1951), pp. 17–29.
- [2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe and H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.
- [3] Z. Bai and Y. Su, *SOAR: a second-order Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM J. Matrix Anal. Appl., 26(2005), pp. 640–659.
- [4] A. Bermúdez, R. G. Durán, R. Rodríguez and J. Solomin, *Finite element analysis of a quadratic eigenvalue problem arising in dissipative acoustics*, SIAM J. Numer. Anal., 38(2000), pp. 267–91.
- [5] J. V. Clark, N. Zhou and K. S. J. Pister, *Modified nodal analysis for MEMS with multienergy domains*, in International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, San Diego, CA, March 27–29, 2000.
- [6] R. W. Clough and J. Penzien, *Dynamics of Structures*, McGraw-Hill, Dusseldorf, 1975.
- [7] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, *Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization*, Math. Comp., 30(1976), pp. 772–795.
- [8] C. E. Davila, *A subspace approach to estimation of autoregressive parameters from noisy measurements*, IEEE Trans. Signal Processing, 46(1998), pp. 531–534.
- [9] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [10] 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.
- [11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd edition, Johns Hopkins University Press, Baltimore, 1996.
- [12] C.-H. Guo, *Numerical solution of a quadratic eigenvalue problem*, Linear Algebra Appl., 385(2004), pp. 391–406.
- [13] J.-S. Guo, W.-W. Lin and C.-S. Wang, *Numerical solutions for large sparse quadratic eigenvalue problems*, Linear Algebra Appl., 225(1995), pp. 57–89.
- [14] R. S. Heeg and B. J. Geurts, *Spatial instabilities of the incompressible attachment-line flow using sparse matrix Jacobi-Davidson techniques*, Appl. Sci. Res., 59(1998), pp. 315–329.
- [15] A. Hilliges, C. Mehl and V. Mehrmann, *On the solution of palindromic eigenvalue problems*, in Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering (ECCO-MAS), Jyväskylä, Finland, 2004.
- [16] M. E. Hochstenbach and H. A. van der Vorst, *Alternatives to the Rayleigh quotient for the quadratic eigenvalue problem*, SIAM J. Sci. Comput., 25(2003), pp. 591–603.
- [17] U. B. Holz, G. Golub and K. H. Law, *A subspace approximation method for the quadratic eigenvalue problem*, SIAM J. Matrix Anal. Appl., 26(2005), pp. 498–521.
- [18] T.-M. Hwang, W.-W. Lin and V. Mehrmann, *Numerical solution of quadratic eigenvalue problems with structure-preserving methods*, SIAM J. Sci. Comput., 24(2003), pp. 1283–1302.
- [19] T.-M. Huang, W.-W. Lin and J. Qian, *Structure-preserving algorithms for palindromic quadratic eigenvalue problems arising from vibration of fast trains*, SIAM J. Matrix Anal. Appl., 30(2009), pp. 1566–1592.
- [20] Z. Jia, *Refined iterative algorithm based on Arnoldi's process for large unsymmetric eigenproblems*, Linear Algebra Appl., 259(1997), pp. 1–23.
- [21] Z. Jia, *A refined subspace iteration algorithm for large sparse eigenproblems*, Appl. Numer. Math., 32(2000), pp. 35–52.
- [22] J. Lampe and H. Voss, *On a quadratic eigenproblem occurring in regularized total least squares*, Comput. Stat. Data Anal., 52(2007), pp. 1090–1102.
- [23] J. Lampe and H. Voss, *Global convergence of RTLSQEP: a solver of regularized total least squares problems via quadratic eigenproblems*, Math. Modelling Anal., 13(2008), pp. 55–66.
- [24] R. Li and Q. Ye, *A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems*, SIAM J. Matrix Anal. Appl., 25(2003), pp. 405–428.
- [25] The MathWorks, Inc., *MATLAB 7*, September 2004.
- [26] K. Meerbergen, *Locking and restarting quadratic eigenvalue solvers*, SIAM J. Sci. Comput., 22(2001), pp. 1814–1839.
- [27] K. Meerbergen, *The quadratic Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM J. Matrix Anal. Appl., 30(2008), pp. 1463–1482.
- [28] J. Qian and W.-W. Lin, *A numerical method for quadratic eigenvalue problems of gyroscopic systems*, J. Sound Vibration, 306(2007), pp. 284–296.
- [29] F. A. Raeven, *A new Arnoldi approach for polynomial eigenproblems*, In Proceedings of the Copper Mountain Conference on Iterative Methods, 1996.
- [30] G. X. Ren and Z. C. Zheng, *A reformulated Arnoldi algorithm for non-classically damped eigenvalue problems*, Int. J. Numer. Methods Eng., 40(1997), pp. 3537–3555.
- [31] A. Ruhe, *Numerical aspects of Gram-Schmidt orthogonalization of vectors*, Linear Algebra Appl., 52(1983), pp. 591–601.
- [32] Y. Saad, *Variations on Arnoldi's method for computing eigenproblems of large non-Hermitian matrices*, Linear Algebra Appl., 34(1980), pp. 269–295.
- [33] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7(1986), pp. 856–869.
- [34] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd edition, SIAM, Philadelphia, 2003.
- [35] H. Schmitz, K. Volk and W. L. Wendland, *On three-dimensional singularities of elastic fields near vertices*, Numer. Methods Partial Differential Equations, 9(1993), pp. 323–337.
- [36] G. L. G. Sleijpen, G. L. Booten, D. R. Fokkema and H. A. van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, 36(1996), pp. 595–633.
- [37] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309(2000), pp. 339–361.
- [38] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Review, 43(2001), pp. 235–286.
- [39] H. A. van der Vorst, *BICGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13(1992), pp. 631–644.
- [40] Q. Ye, *An iterated shift-and-invert Arnoldi algorithm for quadratic matrix eigenvalue problems*, Appl. Math. Comput., 172(2006), pp. 818–827.