NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2010; **00**:1–22
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla

A Semiorthogonal Generalized Arnoldi Method and Its Variations for Quadratic Eigenvalue Problems

Wei-Qiang Huang[1], Tiexiang Li[2]*, Yung-Ta Li[1] and Wen-Wei Lin[3]

[1] *Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan*
[2] *Department of Mathematics, Southeast University, Nanjing 211189, China*
[3] *Department of Mathematics, National Taiwan University, Taipei 106, Taiwan*

SUMMARY

In this paper, we are concerned with the computation of a few eigenpairs with smallest eigenvalues in absolute value of quadratic eigenvalue problems. We first develop a semiorthogonal generalized Arnoldi method where the name comes from the application of a pseudo inner product in the construction of a generalized Arnoldi reduction [25] for a generalized eigenvalue problem. The method applies the Rayleigh-Ritz orthogonal projection technique on the quadratic eigenvalue problem. Consequently it preserves the spectral properties of the original quadratic eigenvalue problem. Furthermore, we propose a refinement scheme to improve the accuracy of the Ritz vectors for the quadratic eigenvalue problem. Given shifts, we also show how to restart the method by implicitly updating the starting vector and constructing better projection subspace. We combine the ideas of the refinement and the restart by selecting shifts upon the information of refined Ritz vectors. Finally an implicitly restarted refined semiorthogonal generalized Arnoldi method is developed. Numerical examples demonstrate that the implicitly restarted semiorthogonal generalized Arnoldi method with or without refinement has superior convergence behaviors than the implicitly restarted Anoldi method applied to the linearized quadratic eigenvalue problem.

Received . . .

KEY WORDS: quadratic eigenvalue problem, semiorthogonal generalized Arnoldi method, orthogonal projection, refinement, refined shifts, implicit restart.

1. INTRODUCTION

The problem of finding scalars $\lambda \in \mathbb{C}$ and nontrivial vectors $\mathbf{x} \in \mathbb{C}^n$ such that

$$(\lambda^2 M + \lambda D + K)\mathbf{x} = \mathbf{0} \tag{1}$$

where M, D and K are $n \times n$ large and sparse matrices, is known as the quadratic eigenvalue problem (QEP). The scalars λ and the associated nonzero vectors $\mathbf{x}$ are called eigenvalues and (right) eigenvectors of the QEP, respectively. Together, $(\lambda, \mathbf{x})$ is called an eigenpair of (1).

The QEP arises in a wide variety of applications, including electrical oscillation, vibro-acoustics, fluid mechanics, signal processing and the simulation of microelectronical mechanical system etc. A good survey of applications, spectral theory, perturbation analysis and numerical approaches can be found in [5, section 11.9], [28] and the references therein.

*Correspondence to: Tiexiang Li, Department of Mathematics, Southeast University, Nanjing 211189, China. E-mail: txli@seu.edu.cn

In practice, some eigenvalues of a QEP near a target τ are interested. Hence we may apply the shift transformation and consider the corresponding shifted QEP

$$(\lambda_\tau^2 M_\tau + \lambda_\tau D_\tau + K_\tau)\mathbf{x} = \mathbf{0}$$

where $\lambda_\tau = \lambda - \tau$, $M_\tau = M$, $D_\tau = 2\tau M + D$ and $K_\tau = \tau^2 M + \tau D + K$. For simplicity, we assume, without loss of generality, that $\tau = 0$. Therefore, throughout this paper, we focus on the problem of finding eigenvalues near the zero (i.e., those small ones in modulus) under the assumption that 0 is not an eigenvalue of the QEP (1) or, equivalently, that K is nonsingular.

Through the so-called "linearization" process, one may first construct a suitable matrix pair (A, B) of size $2n$ and a vector $\boldsymbol{\varphi}$ in $\mathbb{C}^{2n}$ to rewrite the QEP (1) equivalently into a generalized eigenvalue problem (GEP)

$$A\boldsymbol{\varphi} = \tfrac{1}{\lambda}B\boldsymbol{\varphi}. \tag{2}$$

If B is chosen to be nonsingular, one can further reduce (2) as a standard eigenvalue problem (SEP)

$$(B^{-1}A)\boldsymbol{\varphi} = \tfrac{1}{\lambda}\boldsymbol{\varphi} \tag{3}$$

or

$$(AB^{-1})\boldsymbol{\psi} = \tfrac{1}{\lambda}\boldsymbol{\psi} \tag{4}$$

where $\boldsymbol{\psi} = B\boldsymbol{\varphi}$. We call (3) and (4) the left-inverted SEP (ℓ-SEP) and the right-inverted SEP (r-SEP), respectively. After transforming a QEP equivalently to a SEP, the standard Krylov subspace projection methods such as the Arnoldi algorithm can be applied to solve it [28].

The way of linearization is not unique [28]. Here, we consider the second companion form of linearization [8] for the QEP (1)

$$\begin{bmatrix} -D & I_n \\ -M & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix}. \tag{5}$$

where $\widetilde{\mathbf{x}} = -\lambda M\mathbf{x}$. The computational advantage of using the second companion form will be revealed in section 3.

Since K is nonsingular, from (5) the corresponding ℓ-SEP and r-SEP of (1) are, respectively, given by

$$\begin{bmatrix} -K^{-1}D & K^{-1} \\ -M & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix} \tag{6}$$

and

$$\begin{bmatrix} -DK^{-1} & I_n \\ -MK^{-1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} K\mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} K\mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix}. \tag{7}$$

In addition to solving the QEP (1) by SEP (6) or (7), one may also work with the GEP (5) to find the desired eigenpairs of (1). The QZ algorithm [20] is the most popular algorithm for solving the dense GEP of the form (2). This procedure reduces the matrix pair (A, B) equivalently to a Hessenberg-triangular pair (H, R) via unitary transformations in a finite number of steps. This truncated QZ method proposed by Sorensen [25] is one of the approaches for solving large-scale GEPs. The method generalizes the idea of the Arnoldi algorithm to construct a generalized Arnoldi reduction which is a truncation of the QZ iteration and computes the approximated eigenpairs of the original large-scale GEP from the corresponding reduced Hessenberg-triangular pair. Furthermore, in [11], the generalized $\top$-skew-Hamiltonian implicitly restarted shift-and-invert Arnoldi (G$\top$SHIRA) algorithm is discussed for solving the palindromic QEP arising from vibration of fast trains. The generalized $\top$-isotropic Arnoldi process also produces the generalized Arnoldi reduction for a GEP whose coefficient matrices are $\top$-skew-Hamiltonian, however, a further $\top$-bi-isotropic property is required.

However, the linearization technique will double the size of the problem and, in general, matrix structures and spectral properties of the original QEP are not preserved. More importantly, a backward stable technique for linear eigenvalue problems applied to the linearized QEP is not backward stable for the original QEP [27].

To avoid these disadvantages, numerical methods are applied to the large-scale QEP directly. In these methods, the QEP is projected onto a properly chosen low-dimensional subspace to reduce a QEP directly with matrix dimension of low order and solve the reduced QEP by a standard dense matrix approach. Methods of this type include the residual iteration method [21, 12, 19], the Jacobi–Davidson method [22, 23], a Krylov-type subspace method [17], the nonlinear Arnoldi method [29], the second-order Arnoldi (SOAR) method [1, 31, 18] and an iterated shift-and-invert Arnoldi method [32]. While these methods use a similar projection process, the main difference is the selection of projection subspaces.

In this paper, we introduce a *Semiorthogonal Generalized Arnoldi* (SGA) algorithm for the particular linearized problem (5) to generate a SGA decomposition. The SGA algorithm is a variation of the generalized Arnoldi reduction [25]. We then propose an orthogonal projection approach termed as the SGA method to solve the QEP (1) where the projection subspace is defined through its orthonormal basis obtained from the SGA decomposition.

For SEPs, it has been revealed that even though the approximate eigenvalues computed by orthogonal projection methods tend to converge, the corresponding approximate eigenvectors may converge very slowly and even fail to converge. To deal with this problem, Jia [13] proposed a refined Arnoldi method to compute refined approximate eigenvectors. See also [14]. We will extend this idea and use the SGA decomposition to propose a refinement scheme for QEPs.

Due to the storage requirements and computational costs, the order of the SGA decomposition can not be large and shall be limited. Therefore it is necessary to restart the SGA method. Based on the implicitly shifted QZ iterations proposed by Sorensen in [25], we develop a restart technique for the SGA method, called the *Implicitly Restarted SGA* (IRSGA) method. Moreover, according to the information of refined approximate eigenvectors (Ritz vectors), we will propose a procedure for selecting better shifts, termed as refined shifts, for the implicitly shifted QZ algorithm to develop an *Implicitly Restarted Refined SGA* (IRRSGA) method. Compared to the implicitly restarted Arnoldi method applied on the linearized problems (6) and (7), the SGA-type methods, namely IRSGA and IRRSGA, demonstrate better convergence behaviors and require less CPU time in numerical experiments.

The paper is organized as follows. In section 2, we first introduce the SGA algorithm associated with the GEP (5). In section 3, we propose an orthogonal projection method based on the orthonormal basis generated by the SGA algorithm for solving the QEP (1). In section 4, we present a refinement scheme to get better Ritz vectors by taking advantage of the SGA decomposition. In section 5, we develop a restart technique for the SGA-type methods and discuss the selection of shifts according to the information of refinement so that the faster the methods may converge. Numerical examples are presented in section 6 and the concluding remarks are given in section 7.

Throughout this paper, we use the capital letters to denote matrices and the boldface lower case letters to denote vectors. I denotes the identity matrix, $\mathbf{e}_j$ is the jth column of the identity matrix I, and $\mathbf{0}$ denotes zero vectors and matrices. The dimensions of these vectors and matrices are conformed with dimensions used in the context. We adopt the following MATLAB notations: $\mathbf{v}(i:j)$ denotes the subvector of the vector $\mathbf{v}$ that consists of the ith to the jth entries of $\mathbf{v}$. $A(i:j,k:\ell)$ denotes the submatrix of the matrix A that consists of the intersection of the rows i to j and the columns k to ℓ, $A(i:j,:)$ and $A(:,k:\ell)$ select the rows i to j and the columns k to ℓ, respectively, of A. We use $\cdot^\top$ and $\cdot^H$ to denote the transpose and conjugate transpose. $\|\cdot\|_2$ and $\|\cdot\|_F$ denotes the 2-norm and the Frobenius norm, respectively, for a vector or a matrix.

2. THE SGA DECOMPOSITION

In this section, we first give the definition of the SGA decomposition and then discuss the existence and uniqueness of the SGA decomposition in section 2.1. In section 2.2, we will propose a SGA algorithm to generate the SGA decomposition. Subsequently, we discuss the possibility of the early termination of the SGA algorithm.

Definition 2.1 (The SGA decomposition)
Given $M, D, K \in \mathbb{C}^{n\times n}$ and $m \ll n$. We define the mth order SGA decomposition of the QEP (1) to be the relation of the form

$$\begin{bmatrix} -D & I_n \\ -M & \mathbf{0} \end{bmatrix} \begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} V_m \\ U_m \end{bmatrix} H_m + \begin{bmatrix} \mathbf{g}_m \\ \mathbf{f}_m \end{bmatrix} \mathbf{e}_m^\top, \tag{8a}$$

$$\begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix} \begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} V_m \\ U_m \end{bmatrix} R_m, \tag{8b}$$

$$Q_m^H Q_m = I_m, \quad V_m^H V_m = I_m \quad \text{and} \quad V_m^H \mathbf{g}_m = \mathbf{0}. \tag{8c}$$

where $Q_m, P_m, V_m, U_m \in \mathbb{C}^{n\times m}$, $\mathbf{g}_m, \mathbf{f}_m \in \mathbb{C}^n$, and $H_m, R_m \in \mathbb{C}^{m\times m}$ are upper Hessenberg matrix and upper triangular matrix, respectively.

Remark 2.1 (i) The orthogonality requirements in (8c) referred to as the *semiorthogonality* of the SGA decomposition, guarantee the linearly independence of columns of $\left[\begin{smallmatrix} Q_m \\ P_m \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} V_m \\ U_m \end{smallmatrix}\right]$, respectively.

(ii) If the semiorthogonality (8c) is replaced by $Q_m^H Q_m + P_m^H P_m = I_m$, $V_m^H V_m + U_m^H U_m = I_m$ and $V_m^H \mathbf{g}_m + U_m^H \mathbf{f}_m = \mathbf{0}$, we actually obtain a generalized Arnoldi reduction [25] associated with the GEP (5). Therefore, the SGA decomposition can be also viewed as a variation of the generalized Arnoldi reduction.

2.1. *Existence and uniqueness*

Given an $N \times N$ matrix C, $N = 2n$, a nonzero vector $\mathbf{b} \in \mathbb{C}^N$ and a positive integer $m \le n$, the Krylov matrix of C with respect to $\mathbf{b}$ and m is defined by

$$\mathbb{K}[C, \mathbf{b}, m] = \begin{bmatrix} \mathbf{b} & C\mathbf{b} & \cdots & C^{m-1}\mathbf{b} \end{bmatrix}.$$

In the following, for convenience, for a matrix $G \in \mathbb{C}^{N\times j}$ we usually partition G of the form $G = \left[\begin{smallmatrix} G_1 \\ G_2 \end{smallmatrix}\right]$ with $G_1 = G(1:n,:)$ and $G_2 = G(n+1:2n,:)$.

From (8), if we set

$$A = \begin{bmatrix} -D & I_n \\ -M & \mathbf{0} \end{bmatrix}, \quad B = \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix}, \tag{9}$$

and

$$Z_m = \begin{bmatrix} Q_m \\ P_m \end{bmatrix}, \quad Y_m = \begin{bmatrix} V_m \\ U_m \end{bmatrix}, \quad \boldsymbol{\eta}_m = \begin{bmatrix} \mathbf{g}_m \\ \mathbf{f}_m \end{bmatrix}, \tag{10}$$

then the SGA decomposition (8) can be compactly written as

$$AZ_m = Y_m H_m + \boldsymbol{\eta}_m \mathbf{e}_m^\top \tag{11a}$$

$$BZ_m = Y_m R_m \tag{11b}$$

$$Q_m^H Q_m = V_m^H V_m = I_m, \; V_m^H \mathbf{g}_m = \mathbf{0}. \tag{11c}$$

Using equations (8) – (10) of the SGA decomposition (11) and based on the proof technique of Theorem 3.3 in [7], we give the following theorem.

Theorem 2.2
Given $\mathbf{z}_1 \equiv \left[\begin{smallmatrix} \mathbf{q}_1 \\ \mathbf{p}_1 \end{smallmatrix}\right] \in \mathbb{C}^N$ with $\|\mathbf{q}_1\|_2 = 1$ and set $B\mathbf{z}_1 = \rho_1 \mathbf{y}_1 \equiv \rho_1 \left[\begin{smallmatrix} \mathbf{v}_1 \\ \mathbf{u}_1 \end{smallmatrix}\right]$ with $\|\mathbf{v}_1\|_2 = 1$ and $\rho_1 > 0$. Let

$$K_\ell = \mathbb{K}[B^{-1}A, \mathbf{z}_1, m] \equiv \begin{bmatrix} K_{\ell,1} \\ K_{\ell,2} \end{bmatrix} \quad \text{and} \quad K_r = \mathbb{K}[AB^{-1}, \mathbf{y}_1, m] \equiv \begin{bmatrix} K_{r,1} \\ K_{r,2} \end{bmatrix}.$$

Suppose that $K_{\ell,1}$ is of full column rank and $K_{\ell,1} = Q_m R_{\ell,m}$ is the QR-factorization with $Q_m \mathbf{e}_1 = \mathbf{q}_1$ and diagonal entries of $R_{\ell,m}$ are chosen to be positive. Here and hereafter, we use the QR_+-factorization to indicate such a QR-factorization. Then

DOI: 10.1002/nla

(i) $K_{r,1}$ is of full column rank and if $K_{r,1} = V_m R_{r,m}$ is the QR_+-factorization, then $V_m \mathbf{e}_1 = \mathbf{v}_1$.
(ii) Let $P_m = K_{\ell,2} R_{\ell,m}^{-1}$ and $U_m = K_{r,2} R_{r,m}^{-1}$. Then there exist an unreduced upper Hessenberg matrix H_m with positive subdiagonal entries and a upper triangularR_m with positive diagonal entries satisfying the SGA decomposition (11).
(iii) The SGA decomposition (11) is uniquely determined by $Z_m \mathbf{e}_1 = \left[\begin{smallmatrix}\mathbf{q}_1\\ \mathbf{p}_1\end{smallmatrix}\right]$ with $\|\mathbf{q}_1\|_2 = 1$.

Proof

(i) Since

$$\begin{aligned}\begin{bmatrix}K_{r,1}\\ K_{r,2}\end{bmatrix} &= \mathbb{K}[AB^{-1}, \mathbf{y}_1, m] = \begin{bmatrix}\mathbf{y}_1 & AB^{-1}\mathbf{y}_1 & \cdots & (AB^{-1})^{m-1}\mathbf{y}_1\end{bmatrix}\\ &= \frac{1}{\rho_1} B \begin{bmatrix}\mathbf{z}_1 & B^{-1}A\mathbf{z}_1 & \cdots & (B^{-1}A)^{m-1}\mathbf{z}_1\end{bmatrix}\\ &= \frac{1}{\rho_1} BK_\ell = \frac{1}{\rho_1}\begin{bmatrix}K & \mathbf{0}\\ \mathbf{0} & I_n\end{bmatrix}\begin{bmatrix}K_{\ell,1}\\ K_{\ell,2}\end{bmatrix} = \frac{1}{\rho_1}\begin{bmatrix}KK_{\ell,1}\\ K_{\ell,2}\end{bmatrix},\end{aligned} \tag{12}$$

the matrix $K_{r,1} = \frac{1}{\rho_1} KK_{\ell,1}$ is of full column rank and has the unique QR_+-factorization $K_{r,1} = V_m R_{r,m}$ with $V_m \mathbf{e}_1 = \mathbf{v}_1$.
(ii) By assumptions and (10), we get $K_\ell = \left[\begin{smallmatrix}Q_m\\ P_m\end{smallmatrix}\right] R_{\ell,m} = Z_m R_{\ell,m}$. From (i) and (10), we also have $K_r = \left[\begin{smallmatrix}V_m\\ U_m\end{smallmatrix}\right] R_{r,m} = Y_m R_{r,m}$. It follows from (12) that

$$BZ_m = BK_\ell R_{\ell,m}^{-1} = \rho_1 K_r R_{\ell,m}^{-1} = Y_m(\rho_1 R_{r,m} R_{\ell,m}^{-1}) \equiv Y_m R_m$$

where R_m is upper triangular with positive diagonal entries. On the other hand, it holds that

$$B^{-1}A\mathbb{K}[B^{-1}A, \mathbf{z}_1, m] = \mathbb{K}[B^{-1}A, \mathbf{z}_1, m]H_0 + (B^{-1}A)^m \mathbf{z}_1 \mathbf{e}_m^\top \tag{13}$$

where H_0 is the lower shift matrix, i.e., a matrix with ones below the main diagonal and zeros elsewhere. From (13) and (12) we have

$$\begin{aligned}AZ_m &= BZ_m R_{\ell,m} H_0 R_{\ell,m}^{-1} + B(B^{-1}A)^m \mathbf{z}_1 \mathbf{e}_m^\top R_{\ell,m}^{-1}\\ &= Y_m(\rho_1 R_{r,m} H_0 R_{\ell,m}^{-1} + \widetilde{Y}_m^H \mathbf{z}_m \mathbf{e}_m^\top) + [(I - Y_m \widetilde{Y}_m^H)\mathbf{z}_m]\mathbf{e}_m^\top\\ &\equiv Y_m H_m + \boldsymbol{\eta}_m \mathbf{e}_m^\top\end{aligned}$$

where $\mathbf{z}_m = R_{\ell,m}^{-1}(m,m)B(B^{-1}A)^m \mathbf{z}_1 \equiv \left[\begin{smallmatrix}\mathbf{z}_{m,1}\\ \mathbf{z}_{m,2}\end{smallmatrix}\right]$ and $\widetilde{Y}_m^H = [V_m^H \quad \mathbf{0}_{m,n}]$. Since H_0 is unreduced Hessenberg with subdiagonal entries “1”, $R_{\ell,m}$ and $R_{r,m}$ are upper triangular with positive diagonal entries, and V_m is orthogonal, it is easily seen that H_m is unreduced Hessenberg with positive subdiagonal entries and $V_m^H \mathbf{g}_m = V_m^H[(I_n - V_m V_m^H)\mathbf{z}_{m,1}] = \mathbf{0}$.
(iii) By (i) and (ii), we know that Y_m, $\boldsymbol{\eta}_m$, R_m and H_m are uniquely determined by Z_m so we only need to show that Z_m is unique for given $Z_m(:,1) = \left[\begin{smallmatrix}\mathbf{q}_1\\ \mathbf{p}_1\end{smallmatrix}\right]$ with $\|\mathbf{q}_1\|_2 = 1$. From (11), we have $AZ_m = BZ_m(R_m^{-1}H_m) + \boldsymbol{\eta}_m \mathbf{e}_m^\top$. Let $Z_m = \widetilde{Z}_m T_m$ be the QR_+-factorization of Z_m. Then we have the standard Arnoldi decomposition

$$B^{-1}A\widetilde{Z}_m = \widetilde{Z}_m \widetilde{H}_m + \widetilde{\boldsymbol{\eta}}_m \mathbf{e}_m^\top \tag{14}$$

where $\widetilde{H}_m = (T_m R_m^{-1} H_m + \widetilde{Z}_m^H B^{-1} \boldsymbol{\eta}_m \mathbf{e}_m^\top) T_m^{-1}$ and $\widetilde{\boldsymbol{\eta}}_m = (I_m - \widetilde{Z}_m \widetilde{Z}_m^H) B^{-1} \boldsymbol{\eta}_m \mathbf{e}_m^\top T_m^{-1}$. Note that the standard Arnoldi decomposition (14) is unreduced, it is essentially unique. It follows that Q_m and T_m^{-1} of the QR_+-factorization $\widetilde{Q}_m = Q_m T_m^{-1}$ are unique, and then $P_m = \widetilde{P}_m T_m$ is unique. This concludes the proof.

□

Theorem 2.3
If the mth order SGA decomposition (11) exists, then

$$K_\ell = \mathbb{K}[B^{-1}A, \mathbf{z}_1, m] = Z_m[\mathbf{e}_1 \; R_m^{-1}H_m\mathbf{e}_1 \; \cdots \; (R_m^{-1}H_m)^{m-1}\mathbf{e}_1], \tag{15a}$$
$$K_r = \mathbb{K}[AB^{-1}, \mathbf{y}_1, m] = Y_m[\mathbf{e}_1 \; H_mR_m^{-1}\mathbf{e}_1 \; \cdots \; (H_mR_m^{-1})^{m-1}\mathbf{e}_1]. \tag{15b}$$

Proof
It suffices to show

$$K_\ell = [\mathbf{z}_1 \; B^{-1}A\mathbf{z}_1 \; \cdots \; (B^{-1}A)^{m-1}\mathbf{z}_1] = Z_m[\mathbf{e}_1 \; R_m^{-1}H_m\mathbf{e}_1 \; \cdots \; (R_m^{-1}H_m)^{m-1}\mathbf{e}_1]. \tag{16}$$

Since $Z_m\mathbf{e}_1 = \mathbf{z}_1$, we suppose $(B^{-1}A)^{i-1}\mathbf{z}_1 = Z_m(R_m^{-1}H_m)^{i-1}\mathbf{e}_1$, for $i < m$, and prove (16) by induction. From (11) we have

$$\begin{aligned}(B^{-1}A)^i\mathbf{z}_1 &= (B^{-1}A)Z_m(R_m^{-1}H_m)^{i-1}\mathbf{e}_1\\ &= [Z_m(R_m^{-1}H_m) + B^{-1}\boldsymbol{\eta}_m\mathbf{e}_m^\top](R_m^{-1}H_m)^{i-1}\mathbf{e}_1\\ &= Z_m(R_m^{-1}H_m)^i\mathbf{e}_1 + B^{-1}\boldsymbol{\eta}_m(\mathbf{e}_m^\top(R_m^{-1}H_m)^{i-1}\mathbf{e}_1)\\ &= Z_m(R_m^{-1}H_m)^i\mathbf{e}_1\end{aligned} \tag{17}$$

because of $\mathbf{e}_m^\top(R_m^{-1}H_m)^{i-1}\mathbf{e}_1 = 0$, for $i < m$. On the other hand, from (11) follows

$$AB^{-1}Y_m = Y_m(H_mR_m^{-1}) + \widetilde{\boldsymbol{\eta}}_m\mathbf{e}_m^\top, \tag{18}$$

where $\widetilde{\boldsymbol{\eta}}_m = R_m(m,m)^{-1}\boldsymbol{\eta}_m$. Similar to (17), (15b) follows from (18) immediately. □

Remark 2.2
Theorem 2.2 shows that $K_{\ell,1}$ has the QR_+-factorization, $K_{\ell.1} = Q_mR_{\ell,m}$, then the SGA decomposition (11) exists and unique up to $Y_m\mathbf{e}_1 = \mathbf{y}_1$. Theorem 2.3 shows that if the SGA decomposition (11) exists, then K_ℓ and K_r have the QR_+-factorizations (15a) and (15b), respectively.

2.2. *The SGA algorithm*

We now derive an algorithm termed as the SGA algorithm for the computation of the SGA decomposition (9). Given $\mathbf{q}_1, \mathbf{p}_1 \in \mathbb{C}^n$ with $\|\mathbf{q}\|_1 = 1$, let

$$\begin{aligned}&R_1 = \|K\mathbf{q}_1\|_2 \neq 0, \quad \mathbf{v}_1 = K\mathbf{q}_1/R_1, \quad \mathbf{u}_1 = \mathbf{p}_1/R_1,\\ &H_1 = \mathbf{v}_1^H(-D\mathbf{q}_1 + \mathbf{p}_1), \quad \mathbf{g}_1 = -D\mathbf{q}_1 + \mathbf{p}_1 - \mathbf{v}_1H_1 \quad \text{and} \quad \mathbf{f}_1 = -M\mathbf{q}_1 - \mathbf{u}_1H_1,\end{aligned}$$

then $\mathbf{q}_1, \mathbf{p}_1, \mathbf{v}_1, \mathbf{u}_1, \mathbf{g}_1, \mathbf{f}_1, R_1$ and H_1 satisfy the SGA decomposition (8) with $m = 1$.

Suppose that we have computed the jth order $(j < m)$ SGA decomposition

$$\begin{bmatrix}-D & I_n\\ -M & \mathbf{0}\end{bmatrix}\begin{bmatrix}Q_j\\ P_j\end{bmatrix} = \begin{bmatrix}V_j\\ U_j\end{bmatrix}H_j + \begin{bmatrix}\mathbf{g}_j\\ \mathbf{f}_j\end{bmatrix}\mathbf{e}_j^\top, \tag{19a}$$
$$\begin{bmatrix}K & \mathbf{0}\\ \mathbf{0} & I_n\end{bmatrix}\begin{bmatrix}Q_j\\ P_j\end{bmatrix} = \begin{bmatrix}V_j\\ U_j\end{bmatrix}R_j, \tag{19b}$$
$$Q_j^HQ_j = I_j, \quad V_j^HV_j = I_j \quad \text{and} \quad V_j^H\mathbf{g}_j = \mathbf{0}. \tag{19c}$$

To expand the SGA decomposition to order $j+1$, we first assume that the residual vector $\mathbf{g}_j \neq \mathbf{0}$. The case $\mathbf{g}_j = \mathbf{0}$ will be discussed later. Our goal is to find suitable updating vectors and scalars satisfying the SGA decomposition of order $j+1$

$$\begin{bmatrix}-D & I_n\\ -M & \mathbf{0}\end{bmatrix}\begin{bmatrix}Q_j & \mathbf{q}\\ P_j & \mathbf{p}\end{bmatrix} = \begin{bmatrix}V_j & \mathbf{v}\\ U_j & \mathbf{u}\end{bmatrix}\begin{bmatrix}H_j & \mathbf{h}\\ \gamma\mathbf{e}_j^\top & \alpha\end{bmatrix} + \begin{bmatrix}\mathbf{g}_{j+1}\\ \mathbf{f}_{j+1}\end{bmatrix}\mathbf{e}_{j+1}^\top \tag{20a}$$
$$\begin{bmatrix}K & \mathbf{0}\\ \mathbf{0} & I_n\end{bmatrix}\begin{bmatrix}Q_j & \mathbf{q}\\ P_j & \mathbf{p}\end{bmatrix} = \begin{bmatrix}V_j & \mathbf{v}\\ U_j & \mathbf{u}\end{bmatrix}\begin{bmatrix}R_j & \mathbf{r}\\ \mathbf{0} & \rho\end{bmatrix} \tag{20b}$$
$$Q_{j+1}^HQ_{j+1} = I_{j+1}, \quad V_{j+1}^HV_{j+1} = I_{j+1} \quad \text{and} \quad V_{j+1}^H\mathbf{g}_{j+1} = \mathbf{0} \tag{20c}$$

where $Q_{j+1} = [Q_j \ \mathbf{q}]$ and $V_{j+1} = [V_j \ \mathbf{v}]$. Comparing the leading j columns of (20a) with (19a), we get

$$\gamma = \|\mathbf{g}_j\|_2 \neq 0, \quad \mathbf{v} = \mathbf{g}_j/\gamma \neq \mathbf{0} \quad \text{and} \quad \mathbf{u} = \mathbf{f}_j/\gamma. \tag{21}$$

Equating the $(j+1)$st column on both sides of (20b) and noting (20c), the vector $\mathbf{q}$ must satisfy

$$K\mathbf{q} = V_j\mathbf{r} + \mathbf{v}\rho \quad \text{and} \quad Q_j^H\mathbf{q} = \mathbf{0} \tag{22}$$

Premultiplying (22) by $Q_j^H K^{-1}$ and applying the relation $KQ_j = V_j R_j$ gives

$$\mathbf{0} = Q_j^H K^{-1} V_j \mathbf{r} + Q_j^H K^{-1}\mathbf{v}\rho = R_j^{-1}\mathbf{r} + Q_j^H K^{-1}\mathbf{v}\rho$$

and it follows that

$$\mathbf{r} = -R_j Q_j^H K^{-1}\mathbf{v}\rho. \tag{23}$$

Substituting (23) into (22), we have

$$\begin{aligned}\mathbf{q} &= K^{-1}V_j\mathbf{r} + K^{-1}\mathbf{v}\rho \\ &= (Q_j R_j^{-1})(-R_j Q_j^H K^{-1}\mathbf{v}\rho) + K^{-1}\mathbf{v}\rho = (I_j - Q_j Q_j^H)K^{-1}\mathbf{v}\rho\end{aligned}$$

where $\rho \equiv \|(I_j - Q_j Q_j^H)K^{-1}\mathbf{v}\|_2^{-1}$ so that $Q_j^H\mathbf{q} = \mathbf{0}$ and $\|\mathbf{q}\|_2 = 1$. Note that ρ is well-defined, otherwise, $\|(I_j - Q_j Q_j^H)K^{-1}\mathbf{v}\|_2 = 0$ implies $K^{-1}\mathbf{v} \in \operatorname{span}\{Q_j\}$ and hence $\mathbf{v} = KQ_j\mathbf{c} = V_j R_j\mathbf{c}$ for some constant vector $\mathbf{c}$. However, $V_j^H\mathbf{v} = \mathbf{0}$ implies $\mathbf{v} = \mathbf{0}$ which contradicts to the fact (21). After determining $\mathbf{u}$, $\mathbf{r}$ and ρ, (20b) shows that $\mathbf{p}$ can be directly computed by

$$\mathbf{p} = U_j\mathbf{r} + \mathbf{u}\rho.$$

Equating the $(j+1)$st column on both sides of (20a), we know that if we take

$$\begin{bmatrix}\mathbf{h} \\ \alpha\end{bmatrix} = \begin{bmatrix} V_j^H(-D\mathbf{q}+\mathbf{p}) \\ \mathbf{v}^H(-D\mathbf{q}+\mathbf{p})\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}\mathbf{g}_{j+1} \\ \mathbf{f}_{j+1}\end{bmatrix} = \begin{bmatrix} -D & I_n \\ -M & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{q} \\ \mathbf{p}\end{bmatrix} - \begin{bmatrix} V_j & \mathbf{v} \\ U_j & \mathbf{u}\end{bmatrix}\begin{bmatrix}\mathbf{h} \\ \alpha\end{bmatrix} \tag{24}$$

then $V_{j+1}^H\mathbf{g}_{j+1} = \mathbf{0}$ and this completes the $(j+1)$st expanding of the SGA decomposition.

Breakdown and deflation. As we encounter $\mathbf{g}_j = \mathbf{0}$, there are two possibilities, which are called breakdown and deflation. A breakdown occurs if the vector sequence $\{\left[\begin{smallmatrix}\mathbf{v}_1 \\ \mathbf{u}_1\end{smallmatrix}\right], \ldots, \left[\begin{smallmatrix}\mathbf{v}_j \\ \mathbf{u}_j\end{smallmatrix}\right], \left[\begin{smallmatrix}\mathbf{0} \\ \mathbf{f}_j\end{smallmatrix}\right]\}$ is linearly dependent. In this case, both $\mathcal{K}_j(B^{-1}A, \mathbf{q}_1)$ and $\mathcal{K}_j(AB^{-1}, \mathbf{v}_1)$ are invariant subspaces simultaneously and hence the expanding process terminates. On the other hand, it may happen that $\{\left[\begin{smallmatrix}\mathbf{v}_1 \\ \mathbf{u}_1\end{smallmatrix}\right], \ldots, \left[\begin{smallmatrix}\mathbf{v}_j \\ \mathbf{u}_j\end{smallmatrix}\right], \left[\begin{smallmatrix}\mathbf{0} \\ \mathbf{f}_j\end{smallmatrix}\right]\}$ is linearly independent. This situation is called deflation and the expanding process of the SGA decomposition should continue with modified orthogonality requirements.

When a deflation is detected at step j, we assign γ any nonzero number (say $\gamma = 1$), $\mathbf{v} = \mathbf{g}_j = \mathbf{0}$ and $\mathbf{u} = \mathbf{f}_j/\gamma \neq \mathbf{0}$ to start the $(j+1)$st expanding process of the SGA decomposition. Without repeating the discussions above, it is easy to see that $\mathbf{v}$, $\mathbf{u}$ and γ satisfy the jth column of (20a) but $V_{j+1}^H V_{j+1} = \left[\begin{smallmatrix} I_j & \\ & 0\end{smallmatrix}\right]$.

Equating the $(j+1)$st column on both sides of (20b) shows that $\mathbf{q} = K^{-1}V_j\mathbf{r} = Q_j(R_j^{-1}\mathbf{r})$ (since $KQ_j = V_j R_j$) and the orthogonality requirement $\{\mathbf{q}_1, \ldots, \mathbf{q}_j, \mathbf{q}\}$ in (20c) enforces $\mathbf{r} = \mathbf{0}$ and $\mathbf{q} = \mathbf{0}$. Again, by taking ρ any nonzero number (say $\rho = 1$) and then setting $\mathbf{p} = \mathbf{u}\rho = \mathbf{f}_j\gamma^{-1}\rho$, the updating $\mathbf{q}$, $\mathbf{p}$, $\mathbf{r}$ and ρ satisfy the $(j+1)$st column of (20b) but $Q_{j+1}^H Q_{j+1} = \left[\begin{smallmatrix} I_j & \\ & 0\end{smallmatrix}\right]$. This indicates that if the expanding process of the SGA decomposition encounters deflation at a certain step then the updating $\mathbf{v}$-vector and $\mathbf{q}$-vector will be zero simultaneous in the next expanding process. Therefore, the zero vectors of the V-matrix and the Q-matrix in a deflated SGA decomposition appear in the same columns.

To accomplish the $(j+1)$st expanding process of the SGA decomposition, the equations in (24) are given by

$$\begin{bmatrix}\mathbf{h} \\ \alpha\end{bmatrix} = \begin{bmatrix} V_j^H\mathbf{p} \\ 0\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}\mathbf{g}_{j+1} \\ \mathbf{f}_{j+1}\end{bmatrix} = \begin{bmatrix} -D & I_n \\ -M & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{0} \\ \mathbf{p}\end{bmatrix} - \begin{bmatrix} V_j & \mathbf{0} \\ U_j & \mathbf{u}\end{bmatrix}\begin{bmatrix}\mathbf{h} \\ 0\end{bmatrix}.$$

DOI: 10.1002/nla

In summary, if deflations occur at step $1 < j_1, \ldots, j_d \leq m$, then we have the mth order deflated SGA decomposition

$$\begin{bmatrix}-D & I_n\\ -M & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathring{Q}_m\\ \mathring{P}_m\end{bmatrix} = \begin{bmatrix}\mathring{V}_m\\ \mathring{U}_m\end{bmatrix}\mathring{H}_m + \begin{bmatrix}\mathbf{g}_m\\ \mathbf{f}_m\end{bmatrix}\mathbf{e}_m^\top \tag{25a}$$

$$\begin{bmatrix}K & \mathbf{0}\\ \mathbf{0} & I_n\end{bmatrix}\begin{bmatrix}\mathring{Q}_m\\ \mathring{P}_m\end{bmatrix} = \begin{bmatrix}\mathring{V}_m\\ \mathring{U}_m\end{bmatrix}\mathring{R}_m \tag{25b}$$

$$\mathring{Q}_m^H\mathring{Q}_m = J_m, \quad \mathring{V}_m^H\mathring{V}_m = J_m \quad \text{and} \quad \mathring{V}_m^H\mathbf{g}_m = \mathbf{0} \tag{25c}$$

where $\mathring{Q}_m(:, j_i) = \mathring{V}_m(:, j_i) = \mathbf{0}$, $\mathring{R}_m(1 : j_i - 1, j_i) = \mathbf{0}$, $\mathring{H}_m(j_i, j_i) = 0$, $\mathring{R}_m(j_i, j_i)$, $\mathring{H}_m(j_i, j_i - 1)$ are nonzero numbers and

$$J_m(s,t) = \begin{cases} 1 & \text{if } s = t \neq j_i, \\ 0 & \text{otherwise,} \end{cases} \qquad i = 1, \ldots, d.$$

The following theorem distinguishes the deflation and breakdown.

Theorem 2.4 ([1], Lemma 3.2)
For a sequence of linearly independent vectors $\{\mathbf{y}_1, \ldots, \mathbf{y}_m\}$ with partition $\mathbf{y}_i = \left[\begin{smallmatrix}\mathbf{v}_i\\ \mathbf{u}_i\end{smallmatrix}\right]$, if there exists a subsequence $\{\mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_j}\}$ of the $\mathbf{v}$ vectors that are linearly independent and the remaining vectors are zeros, $\mathbf{v}_{i_{j+1}} = \cdots = \mathbf{v}_{i_m} = \mathbf{0}$, then a vector $\mathbf{y} = \left[\begin{smallmatrix}\mathbf{0}\\ \mathbf{u}\end{smallmatrix}\right] \in \text{span}\{\mathbf{y}_1, \ldots, \mathbf{y}_m\}$ if and only if $\mathbf{u} \in \text{span}\{\mathbf{u}_{i_{j+1}}, \ldots, \mathbf{u}_{i_m}\}$.

The pseudocode for the SGA algorithm that iteratively generates an mth order (deflated) SGA decomposition is listed in Algorithm 2.1.

Algorithm 2.1 SGA Algorithm

Input: $M, D, K \in \mathbb{C}^{n\times n}$, $\mathbf{q}_1, \mathbf{p}_1 \in \mathbb{C}^n$ with $\|\mathbf{q}_1\|_2 = 1$ and $m \geq 1$.
Output: Q_m, V_m, U_m, $\mathbf{g}_m := \mathbf{g}$, $\mathbf{f}_m := \mathbf{f}$, $\mathbb{M}_m$, $\mathbb{D}_m$, upper Hessenberg $H_m \in \mathbb{C}^{m\times m}$ and upper triangular $R_m \in \mathbb{C}^{m\times m}$ satisfy the SGA decomposition (8) of order m.

```
$Q_1 := \mathbf{q}_1$;  $R_1 := \|K\mathbf{q}_1\|_2$;  $V_1 := K\mathbf{q}_1/R_1$;  $U_1 := \mathbf{p}_1/R_1$;  $\mathbb{M}_1 := M\mathbf{q}_1$;  $\mathbb{D}_1 := D\mathbf{q}_1$;
$\mathbf{g} := -\mathbb{D}_1 + \mathbf{p}_1$;  $H_1 := V_1^H\mathbf{g}$;  $\mathbf{g} := \mathbf{g} - V_1H_1$;  $\mathbf{f} := -\mathbb{M}_1 - U_1H_1$;
for $j = 1, 2, \ldots, m-1$ do
  if $\mathbf{g} \neq \mathbf{0}$ then
    $\gamma := \|\mathbf{g}\|_2$;  $\mathbf{v} := \mathbf{g}/\gamma$;  $\mathbf{u} := \mathbf{f}/\gamma$;  $V_{j+1} := [V_j\ \mathbf{v}]$;  $U_{j+1} := [U_j\ \mathbf{u}]$;  $H_j := \begin{bmatrix}H_j\\ \gamma\mathbf{e}_j^\top\end{bmatrix}$;
    Solve $K\mathbf{q} = \mathbf{v}_{j+1}$ for $\mathbf{q}$
    $\mathbf{r} := Q_j^H\mathbf{q}$;  $\mathbf{q} := \mathbf{q} - Q_j\mathbf{r}$;  $\rho := \|\mathbf{q}\|_2^{-1}$;  $\mathbf{q} := \mathbf{q}\rho$;  $\boldsymbol{\mu} := M\mathbf{q}$;  $\boldsymbol{\delta} := D\mathbf{q}$;
    $Q_{j+1} := [Q_j\ \mathbf{q}]$;  $\mathbb{M}_{j+1} := [\mathbb{M}_j\ \boldsymbol{\mu}]$;  $\mathbb{D}_{j+1} := [\mathbb{D}_j\ \boldsymbol{\delta}]$;  $R_{j+1} := \begin{bmatrix}R_j & \mathbf{r}\\ \mathbf{0} & \rho\end{bmatrix}$;
    $\mathbf{g} := -\boldsymbol{\delta} + U_{j+1}R_{j+1}(:, j+1)$;  $\mathbf{h} := V_{j+1}^H\mathbf{g}$;  $H_{j+1} := [H_j\ \mathbf{h}]$;
    $\mathbf{g} := \mathbf{g} - V_{j+1}\mathbf{h}$;  $\mathbf{f} := -\boldsymbol{\mu} - U_{j+1}\mathbf{h}$;
  else
    if $\mathbf{f} \in \text{span}\{\mathbf{u}_i \mid i : \mathbf{v}_i = \mathbf{0},\ 1 \leq i \leq j\}$ then
      breakdown
    else
      $V_{j+1} := [V_j\ \mathbf{0}]$;  $U_{j+1} := [U_j\ \mathbf{f}]$;  $Q_{j+1} := [Q_j\ \mathbf{0}]$;  $\mathbb{M}_{j+1} := [\mathbb{M}_j\ \mathbf{0}]$;  $\mathbb{D}_{j+1} := [\mathbb{D}_j\ \mathbf{0}]$;
      $\mathbf{h} := V_j^H\mathbf{f}$;  $H_{j+1} := \begin{bmatrix}H_j & \mathbf{h}\\ \mathbf{e}^\top & 0\end{bmatrix}$;  $R_{j+1} := \begin{bmatrix}R_j & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}$;  $\mathbf{g} := \mathbf{f} - V_j\mathbf{h}$;  $\mathbf{f} := -U_j\mathbf{h}$;
    end if
  end if
end for
```

Remark 2.3
The following remarks give some detailed explanations of the SGA algorithm.

(i) At each expanding process of the SGA decomposition, we need to solve a linear system (see line 6 of the SGA algorithm). To make the computation more efficient, a factorization of K, such as the LU factorization, should be made available outside of the first **for**-loop of the SGA algorithm.

(ii) At lines 8 and 15 of the SGA algorithm, we additionally store the vectors $D\mathbf{q}_j$ and $M\mathbf{q}_j$ at each expanding step and output two $n \times m$ matrices

$$\mathbb{D}_m := [D\mathbf{q}_1 \ \cdots \ D\mathbf{q}_m] = DQ_m \quad \text{and} \quad \mathbb{M}_m := [M\mathbf{q}_1 \ \cdots \ M\mathbf{q}_m] = MQ_m.$$

The pre-stored matrices $\mathbb{D}_m$ and $\mathbb{M}_m$ save computational costs in the subsequent projection process for solving the QEP.

(iii) From (8b), we know that P_m can be completely determined by U_m, that is, for $j = 1, \ldots, m$, $\mathbf{p}_j$ can be replaced by the relation $\mathbf{p}_j = U_m(:, 1:j)R_m(1:j, j)$. See line 9 of the SGA algorithm. Hence we only need to evaluate and store Q_m, V_m, $\mathbf{g}_m$, U_m, $\mathbf{f}_m$, H_m and R_m as we implement the SGA algorithm.

(iv) At line 12 of the SGA algorithm, we decide whether the expanding process encounters a deflation or a breakdown. In practice, we use the modified Gram–Schmidt procedure to check it as suggested in [1].

3. THE SGA METHOD FOR SOLVING QEPS

In this section, we use the unitary matrix Q_m produced by the SGA algorithm to develop an orthogonal projection technique to solve the QEP. For simplicity, we assume that the deflation does not occur and hence $Q_m^H Q_m = I_m$. When the deflation occurs, the same orthogonal projection technique is applied with the modification of replacing Q_m with the nonzero columns of $\mathring{Q}_m$ shown in (25).

3.1. *The SGA method*

The SGA method applies the Rayleigh-Ritz subspace projection technique on the subspace $\mathcal{Q}_m \equiv \text{span}\{Q_m\}$ with the Galerkin condition:

$$(\theta^2 M + \theta D + K)\boldsymbol{v} \perp \mathcal{Q}_m,$$

that is, we seek an approximate eigenpair $(\theta, \boldsymbol{v})$ with $\theta \in \mathbb{C}$, $\boldsymbol{v} \in \mathcal{Q}_m$ such that

$$\boldsymbol{\omega}^*(\theta^2 M + \theta D + K)\boldsymbol{v} = 0 \quad \text{for all } \boldsymbol{\omega} \in \mathcal{Q}_m \tag{26}$$

where $\cdot^*$ denotes the transpose $\cdot^\top$ when M, D, K are real or complex symmetric, otherwise, $\cdot^*$ denotes the conjugate transpose $\cdot^H$ of matrices. Since $\boldsymbol{v} \in \mathcal{Q}_m$, it can be written as $\boldsymbol{v} = Q_m\boldsymbol{\xi}$ and (26) implies that θ and $\boldsymbol{\xi}$ must satisfy the reduced QEP:

$$(\theta^2 M_m + \theta D_m + K_m)\boldsymbol{\xi} = \mathbf{0} \tag{27}$$

where

$$M_m = Q_m^* M Q_m, \quad D_m = Q_m^* D Q_m, \quad K_m = Q_m^* K Q_m. \tag{28}$$

The eigenpair $(\theta, \boldsymbol{\xi})$ of the small-scale QEP (27) defines a Ritz pair $(\theta, Q_m\boldsymbol{\xi})$ of the QEP (1) whose accuracy is measured by the norm of the residual vector $\mathbf{r}_{\theta,\boldsymbol{\xi}} = (\theta^2 M + \theta D + K)Q_m\boldsymbol{\xi}$.

Note that by explicitly formulating the matrices M_m, D_m, and K_m, essential structures of M, D, and K are preserved. For example, if M is symmetric positive definite, so is M_m. As a result, essential spectral properties of the QEP will be preserved. For example, if the QEP is a gyroscopic dynamical system in which M and K are symmetric, one of them is positive definite, and D is skew-symmetric, then the reduced QEP is also a gyroscopic system. It is known that in this case, the eigenvalues are symmetrically placed with respect to both the real and imaginary axes [16]. Such a spectral property will be preserved in the reduced QEP.

Before we present the SGA method for solving the QEPs, we discuss how to take advantage of the SGA algorithm to efficiently generate the coefficient matrices (M_m, D_m, K_m) of the projected QEP (28). As we describe in Remark 2.3(ii), the resultant matrices $\mathbb{M}_m := MQ_m$ and $\mathbb{D}_m := DQ_m$ produced from the SGA algorithm provide us the necessary multiplications of M, D with Q_m. For the projected matrix K_m, even if the SGA algorithm does not exactly perform the matrix-vector product of K and $\mathbf{q}_j$ at each step, $j = 1, \ldots, m$, we can use the equality $KQ_m = V_m R_m$ in (8b) to reduce the computational costs. The product of $V_m R_m$ needs about $2nm^2$ flops, but the product of KQ_m needs about $2n^2 m$ flops. Therefore, the small-scale matrices M_m and D_m can be respectively generated by

$$M_m = Q_m^* \mathbb{M}_m, \quad D_m = Q_m^* \mathbb{D}_m \quad \text{and} \quad K_m = Q_m^* V_m R_m. \tag{29}$$

Totally, (28) needs about $6n^2 m + 6nm^2$ flops to generate the coefficient matrices of the projected QEP (27), however, the matrix products (29) only need $8nm^2$ flops. Also note that if we consider the first companion form linearization of the QEP (1), there is no such an advantage. That is, (28) is the only way to generate the coefficient matrices of the reduced QEP (27).

Algorithm 3.1 The SGA method

```
Input: $M, D, K \in \mathbb{C}^{n\times n}$, $\mathbf{q}_1, \mathbf{p}_1 \in \mathbb{C}^n$ with $\|\mathbf{q}_1\|_2 = 1$ and $m \geq k \geq 1$.
Output: $m$ Ritz pairs and their relative residuals.
1: Run the SGA algorithm (Algorithm 2.1) to generate an $m$th order SGA decomposition (8).
2: Compute $M_m$, $D_m$ and $K_m$ via (29).
3: Solve the reduced QEP (27) for $(\theta_i, \boldsymbol{\xi}_i)$ with $\|\boldsymbol{\xi}_i\|_2 = 1$, $i = 1, \ldots, 2m$ and sorting Ritz values
   so that $\{(\theta_1, Q_m\boldsymbol{\xi}_1), \ldots, (\theta_k, Q_m\boldsymbol{\xi}_k)\}$ are wanted Ritz pairs.
4: Test the accuracy of Ritz pairs $(\theta_i, \boldsymbol{v}_i)$, $\boldsymbol{v}_i = Q_m\boldsymbol{\xi}_i$, $i = 1, \ldots, k$ as approximate eigenvalues
   and eigenvectors of the QEP (1) by the relative norms of residual vectors:
```

$$\frac{\|(\theta_i^2 M + \theta_i D + K)\boldsymbol{v}_i\|_2}{|\theta_i|^2\|M\|_F + |\theta_i|\|D\|_F + \|K\|_F}, \qquad i = 1, \ldots, k. \tag{30}$$

3.2. *Projection subspace*

In this subsection we explain the motivation of choosing the projection subspace $\mathcal{Q}_m \equiv \mathrm{span}\{Q_m\}$ where Q_m is generated from the SGA algorithm. We first recall a lemma in the SOAR method [1].

Lemma 3.1 ([1], Lemma 2.2)
Let C be an arbitrary $N \times N$ matrix. Let $W_{m+1} = [W_m \; \mathbf{w}_{m+1}]$ be an $N \times (m+1)$ rectangular matrix that satisfies $CW_m = W_{m+1}\underline{H}_m$ for an $(m+1) \times m$ upper Hessenberg matrix $\underline{H}_m$. Then there is an upper triangular matrix T_m such that

$$W_m T_m = \begin{bmatrix} \mathbf{w}_1 & C\mathbf{w}_1 & \cdots & C^{m-1}\mathbf{w}_1 \end{bmatrix}.$$

Furthermore, if the first $m-1$ subdiagonal elements of $\underline{H}_m$ are nonzero, then T_m is nonsingular and

$$\mathrm{span}\{W_m\} = \mathcal{K}_m(C, \mathbf{w}_1).$$

Next, we consider a Krylov subspace associated with the linearized eigenvalue problem (3) and show that it is embedded into a larger subspace spanned by some column vectors in the SGA decomposition (8).

Theorem 3.1
Consider the SGA decomposition (8) of order m. Let

$$\widehat{Q}_{\widehat{m}} = \begin{array}{cc} & \begin{array}{ccc} m & m & 1 \end{array} \\ \begin{array}{c} n \\ n \end{array} & \left[\begin{array}{ccc} Q_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -MQ_m & \mathbf{p}_1 \end{array}\right] \end{array} \in \mathbb{C}^{2n\times(2m+1)}. \tag{31}$$

Then, for A and B defined in (9), we have $\mathcal{K}_m(B^{-1}A, [\begin{smallmatrix}\mathbf{q}_1\\ \mathbf{p}_1\end{smallmatrix}]) \subseteq \operatorname{span}\{\widehat{Q}_{\widehat{m}}\}$.

Proof
From (11b), we have $[\begin{smallmatrix}V_m\\ U_m\end{smallmatrix}] = [\begin{smallmatrix}K & \mathbf{0}\\ \mathbf{0} & I_n\end{smallmatrix}][\begin{smallmatrix}Q_m\\ P_m\end{smallmatrix}]R_m^{-1}$. Substituting it into the equation (11a) and then premultiplying it by $[\begin{smallmatrix}K^{-1} & \mathbf{0}\\ \mathbf{0} & I_n\end{smallmatrix}]$, we get

$$\begin{bmatrix} -K^{-1}D & K^{-1} \\ -M & \mathbf{0} \end{bmatrix} \begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} Q_m & \mathbf{q}_m^{\ell} \\ P_m & \mathbf{p}_m^{\ell} \end{bmatrix} \begin{bmatrix} H_m^{\ell} \\ \mathbf{e}_m^{\top} \end{bmatrix} \tag{32}$$

where $H_m^{\ell} = R_m^{-1}H_m$ is an unreduced upper Hessenberg matrix, $\mathbf{q}_m^{\ell} = K^{-1}\mathbf{g}_m$ and $\mathbf{p}_m^{\ell} = \mathbf{f}_m$. By (32) and Lemma 3.1, we know that

$$\mathcal{K}_m\left(B^{-1}A, \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{p}_1 \end{bmatrix}\right) \equiv \mathcal{K}_m\left(\begin{bmatrix} -K^{-1}D & K^{-1} \\ -M & \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{p}_1 \end{bmatrix}\right) = \operatorname{span}\left\{\begin{bmatrix} Q_m \\ P_m \end{bmatrix}\right\} \tag{33}$$

and the set $\{[\begin{smallmatrix}\mathbf{q}_1\\ \mathbf{p}_1\end{smallmatrix}], \ldots, [\begin{smallmatrix}\mathbf{q}_m\\ \mathbf{p}_m\end{smallmatrix}]\}$ is a non-orthonormal basis of the above Krylov subspace (33). Next, we show that

$$\begin{bmatrix} \mathbf{q}_i \\ \mathbf{p}_i \end{bmatrix} \in \operatorname{span}\{\widehat{Q}_{\widehat{m}}\} \quad \text{for } \ i = 1, \ldots, m, \tag{34}$$

and the conclusion of Theorem 3.1 follows directly from (33) and (34).

To prove (34), it suffices to show that $\mathbf{p}_i \in \operatorname{span}\{-MQ_m, \mathbf{p}_1\}$, $1 \le i \le m$. We prove this by induction. Clearly, $\mathbf{p}_1 \in \operatorname{span}\{-MQ_m, \mathbf{p}_1\}$. Suppose that $\mathbf{p}_1, \ldots, \mathbf{p}_i \in \operatorname{span}\{-MQ_m, \mathbf{p}_1\}$ for $1 < i \le m-1$. From the equality (32), we have $-MQ_m = P_mH_m^{\ell} + \mathbf{p}_m^{\ell}\mathbf{e}_m^{\top}$. Thus,

$$-M\mathbf{q}_i = P_mH_m^{\ell}(:, i) = P_iH_m^{\ell}(1:i, i) + \mathbf{p}_{i+1}H_m^{\ell}(i+1, i)$$

and it follows that

$$\mathbf{p}_{i+1} = H_m^{\ell}(i+1, i)^{-1}\left(-M\mathbf{q}_i - P_iH_m^{\ell}(1:i, i)\right) \in \operatorname{span}\{-MQ_m, \mathbf{p}_1\}.$$

We complete the proof. □

Instead of using the Krylov subspace $\mathcal{K}_m(B^{-1}A, [\begin{smallmatrix}\mathbf{q}_1\\ \mathbf{p}_1\end{smallmatrix}])$, we choose the larger subspace $\mathcal{K}_m(B^{-1}A, [\begin{smallmatrix}\mathbf{q}_1\\ \mathbf{p}_1\end{smallmatrix}])$ to extract approximations of eigenpairs. To project the coefficient matrices of the GEP (5) onto the subspace $\operatorname{span}\{\widehat{Q}_{\widehat{m}}\}$, we get

$$\widehat{Q}_{\widehat{m}}^{*} \begin{bmatrix} -D & I_n \\ -M & \mathbf{0} \end{bmatrix} \widehat{Q}_{\widehat{m}} = \begin{array}{cc} & \begin{array}{ccc} m & m & 1 \end{array} \\ \begin{array}{c} m \\ m \\ 1 \end{array} & \left[\begin{array}{cc|c} -D_m & -M_m & Q_m^*\mathbf{p}_1 \\ N_m & \mathbf{0} & \mathbf{0} \\ \hline -\mathbf{p}_1^*MQ_m & \mathbf{0} & 0 \end{array}\right] \end{array} \equiv \widehat{A} \tag{35a}$$

$$\widehat{Q}_{\widehat{m}}^{*} \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix} \widehat{Q}_{\widehat{m}} = \begin{array}{cc} & \begin{array}{ccc} m & m & 1 \end{array} \\ \begin{array}{c} m \\ m \\ 1 \end{array} & \left[\begin{array}{cc|c} K_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & N_m & -Q_m^*M^*\mathbf{p}_1 \\ \hline \mathbf{0} & -\mathbf{p}_1^*MQ_m & \mathbf{p}_1^*\mathbf{p}_1 \end{array}\right] \end{array} \equiv \widehat{B} \tag{35b}$$

where M_m, D_m, K_m are defined in (28) and $N_m = Q_m^*M^*MQ_m$. Therefore, the GEP (5) is reduced to the problem

$$\widehat{A}\mathbf{s} = \nu\widehat{B}\mathbf{s} \tag{36}$$

DOI: 10.1002/nla

with $\widehat{A}$ and $\widehat{B}$ defined in (35). Observe that if we premultiply (36) by the nonsingular matrix

$$L \equiv \begin{bmatrix} I_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_m & \mathbf{0} \\ \mathbf{0} & \mathbf{p}_1^* M Q_m N_m^{-1} & 1 \end{bmatrix}$$

then the coefficient matrices of the resultant GEP $(L\widehat{A})\mathbf{s} = \mu(L\widehat{B})\mathbf{s}$ are respectively of the forms

$$L\widehat{A} \equiv \left[\begin{array}{cc|c} -D_m & -M_m & Q_m^* \mathbf{p}_1 \\ N_m & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & 0 \end{array}\right], \quad L\widehat{B} \equiv \left[\begin{array}{cc|c} K_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & N_m & -Q_m^* M^* \mathbf{p}_1 \\ \hline \mathbf{0} & \mathbf{0} & c \end{array}\right] \tag{37}$$

where $c = \mathbf{p}_1^*(I_n - MQ_m N_m^{-1} Q_m^* M^*)\mathbf{p}_1$. The pencil obtained from the last component of both matrices in (37) either provides the zero eigenvalue or be a singular pencil. In both cases, the eigenvalues computed from this pencil are not wanted. Therefore, we can simply drop the last column and row of both matrices in (37) to consider the leading $2m \times 2m$ submatrices, which is just the first companion form linearization [8] of the reduced QEP (27), for solving QEPs.

4. REFINED SGA METHOD

As we obtain a Ritz pair $(\theta, \boldsymbol{v}_\theta)$ by the SGA method, a refinement strategy for QEP is to seek a unit vector $\boldsymbol{v}_\theta^+ \in \mathcal{Q}_m = \operatorname{span}\{Q_m\}$ satisfying

$$\boldsymbol{v}_\theta^+ \equiv \underset{\boldsymbol{v} \in \mathcal{Q}_m,\ \|\boldsymbol{v}\|_2 = 1}{\arg\min} \|(\theta^2 M + \theta D + K)\boldsymbol{v}\|_2. \tag{38}$$

Here we call $\boldsymbol{v}_\theta^+$ the refined Ritz vector corresponding to the Ritz value θ. We next turn to propose a novel refinement scheme by taking advantage of the SGA decomposition for computing refined Ritz vectors. An other refinement scheme for QEPs we refer to [15].

Let $(\theta, \boldsymbol{\xi}_\theta)$ be an eigenpair obtained from the small-scale QEP (27), then $(\theta, \boldsymbol{v}_\theta) = (\theta, Q_m \boldsymbol{\xi}_\theta)$ is a Ritz pair the QEP (1). To solve the optimization problem (38), we find that

$$\begin{aligned} &(\theta^2 M + \theta D + K)Q_m \\ &= \theta^2(-U_m H_m - \mathbf{f}_m \mathbf{e}_m^\top) + \theta(P_m - V_m H_m - \mathbf{g}_m \mathbf{e}_m^\top) + V_m R_m \\ &= V_m(-\theta H_m + R_m) + \mathbf{g}_m(-\theta \mathbf{e}_m^\top) + U_m(-\theta^2 H_m + \theta R_m) + \mathbf{f}_m(-\theta^2 \mathbf{e}_m^\top) \\ &= \begin{bmatrix} V_m & \mathbf{g}_m & U_m & \mathbf{f}_m \end{bmatrix} \begin{bmatrix} -\theta H_m + R_m \\ -\theta \mathbf{e}_m^\top \\ -\theta^2 H_m + \theta R_m \\ -\theta^2 \mathbf{e}_m^\top \end{bmatrix}, \end{aligned} \tag{39}$$

where we use the SGA decomposition (8) in the first two equalities. Since V_m is a column orthonormal matrix, the QR factorization of $\begin{bmatrix} V_m & \mathbf{g}_m & U_m & \mathbf{f}_m \end{bmatrix}$ is of the form

$$\begin{bmatrix} V_m & \mathbf{g}_m & U_m & \mathbf{f}_m \end{bmatrix} = \begin{bmatrix} V_m & \widetilde{\mathbf{g}}_m & \widetilde{U}_m & \widetilde{\mathbf{f}}_m \end{bmatrix} \begin{bmatrix} I_m & \mathbf{t}_{12} & T_{13} & \mathbf{t}_{14} \\ & t_{22} & \mathbf{t}_{23} & t_{24} \\ & & T_{33} & \mathbf{t}_{34} \\ & & & t_{44} \end{bmatrix} \tag{40}$$

where $\begin{bmatrix} V_m & \widetilde{\mathbf{g}}_m & \widetilde{U}_m & \widetilde{\mathbf{f}}_m \end{bmatrix}$ is unitary. Since the vector 2-norm is invariant under unitary transformations, (39) and (40) imply

$$\min_{\boldsymbol{v} \in \mathcal{Q}_m,\ \|\boldsymbol{v}\|_2 = 1} \|(\theta^2 M + \theta D + K)\boldsymbol{v}\|_2 = \min_{\|\boldsymbol{\xi}\|_2 = 1} \|(\theta^2 M + \theta D + K)Q_m \boldsymbol{\xi}\|_2 = \min_{\|\boldsymbol{\xi}\|_2 = 1} \|S(m, \theta)\boldsymbol{\xi}\|_2$$

where

$$S(m,\theta) \equiv \begin{bmatrix} I_m & \mathbf{t}_{12} & T_{13} & \mathbf{t}_{14} \\ & t_{22} & \mathbf{t}_{23} & t_{24} \\ & & T_{33} & \mathbf{t}_{34} \\ & & & t_{44} \end{bmatrix} \begin{bmatrix} -\theta H_m + R_m \\ -\theta \mathbf{e}_m^\top \\ -\theta^2 H_m + \theta R_m \\ -\theta^2 \mathbf{e}_m^\top \end{bmatrix} \in \mathbb{C}^{(2m+2)\times m}. \tag{41}$$

Since the right singular vector $V_\theta \mathbf{e}_m$ of $S(m,\theta)$ corresponding to the smallest singular value $s_{\theta,\min}$ yields the minimum $\|S(m,\theta)V_\theta \mathbf{e}_m\|_2 = s_{\theta,\min}$, as a consequence, the unit vector $\boldsymbol{v}_\theta^+ \equiv Q_m V_\theta \mathbf{e}_m$ is the solution to the minimization problem (38) with minimum $s_{\theta,\min}$. In summary, we have the following theorem.

Theorem 4.1
Let $(\theta, Q_m\boldsymbol{\xi}_\theta)$ be a Ritz pair the QEP (1) computed from the SGA method. Let $S(m,\theta) = U_\theta \Sigma_\theta (V_\theta)^H$ be a singular value decomposition of $S(m,\theta)$ defined in (41) and $s_{\theta,\min}$ be its smallest singular value. Then the vector $\boldsymbol{v}^+ \equiv Q_m V_\theta \mathbf{e}_m$ is the solution to the optimization problem (38) with minimum $s_{\theta,\min}$.

When applying the refinement strategy for several Ritz pairs, we compute the QR factorization (40) only once and subsequently use the factorization for refining each Ritz pair. Combining the SGA method with the refinement strategy, we propose the refined SGA (RSGA) method in Algorithm 4.1.

Algorithm 4.1 RSGA method

```
Input: $M, D, K \in \mathbb{C}^{n\times n}$, $\mathbf{q}_1, \mathbf{p}_1 \in \mathbb{C}^n$ with $\|\mathbf{q}_1\|_2 = 1$ and $m \geq k \geq 1$.
Output: $m$ refined Ritz pairs and their relative residuals.
Run steps 1–3 of the SGA method to obtain $k$ wanted Ritz pairs $(\theta_i, Q_m\boldsymbol{\xi}_i)$, $i = 1, \dots, k$.
Calculate a $QR$ factorization of $[V_m\ \mathbf{g}_m\ U_m\ \mathbf{f}_m]$ where the $Q$-factor and $R$-factor are denoted
  as in (40).
for $i = 1, \dots, k$ do
    Calculate the matrix $S(m,\theta_i)$ as defined in (41).
    Calculate a compact singular value decomposition of $S(m,\theta_i) = U_{\theta_i}\Sigma_{\theta_i}(V_{\theta_i})^H$.
    Let $s_{\theta_i,\min}$ be the smallest singular value of $S(m,\theta_i)$. Then the refined Ritz vector is given
    by $\boldsymbol{v}_i^+ = Q_m V_{\theta_i}\mathbf{e}_m$ and the corresponding relative residual is given by

        $$\frac{\|(\theta_i^2 M + \theta_i D + K)\boldsymbol{v}_i^+\|_2}{|\theta_i|^2\|M\|_F + |\theta_i|\|D\|_F + \|K\|_F} = \frac{s_{\theta_i,\min}}{|\theta_i|^2\|M\|_F + |\theta_i|\|D\|_F + \|K\|_F}. \tag{42}$$

end for
```

5. IMPLICIT RESTART OF THE SGA METHOD

Similar to the standard implicitly restarted Arnoldi (IRA) method [24] for SEPs, the SGA/RSGA method also needs restarting to control storage and orthogonalization expense. In this section, we will apply the implicitly shifted QZ iteration [25] to implicitly restart the SGA/RSGA method, namely IRSGA/IRRSGA.

5.1. *The IRSGA method and the IRRSGA method*

In this subsection, we first briefly discuss the implicitly restarted step of the SGA algorithm based on the implicitly shifted QZ iteration [25]. For details, see [25, 26].

Suppose we have computed the mth order SGA decomposition (11). For given shifts $\vartheta_1, \dots, \vartheta_p$, $p = m - k$, which are in general the unwanted approximate eigenvalues, let E_i and F_i be unitary matrices computed by the implicitly shifted QZ iteration with the single shift ϑ_i, $i = 1, \dots, p$. Write $E^+ = E_1 \cdots E_p$ and $F^+ = F_1 \cdots F_p$. Note that F_i is upper Hessenberg, $i = 1, \dots, p$.

Let $H_m^+ \equiv (E^+)^H H_m F^+$, $R_m^+ \equiv (E^+)^H R_m F^+$, $Z_m^+ \equiv Z_m F^+$ and $Y_m^+ \equiv Y_m E^+$. Then H_m^+ and R_m^+ are again upper Hessenberg and upper triangular, respectively. Let $Q_m^+ \equiv Q_m F^+$ and $V_m^+ \equiv V_m E^+$ then $(Q_m^+)^H Q_m^+ = (V_m^+)^H V_m^+ = I_m$. Postmultiplying (11a) and (11b) by F^+, we get

$$AZ_m^+ = Y_m^+ H_m^+ + \boldsymbol{\eta}_m \mathbf{e}_m^\top F^+, \tag{43a}$$

$$BZ_m^+ = Y_m^+ R_k^+. \tag{43b}$$

Since $\mathbf{e}_m^\top F_1 = [0 \ \cdots \ 0 \ \alpha_1 \ \beta_1]$, by induction, we see that the first $k-1$ entries of $\mathbf{e}_m^\top F^+$ are zeros.

Let $\boldsymbol{\eta} \equiv h_{k+1,k}^+ \mathbf{y}_{k+1}^+ + F^+(m,k)\boldsymbol{\eta}_m$. Drop the last $m-k$ columns of (43a) and (43b), and then set $\boldsymbol{\eta}_k^+ \equiv \boldsymbol{\eta}$. Then, by writing $Z_k^+ = \begin{bmatrix} Q_k^+ \\ P_k^+ \end{bmatrix}$, $Y_k^+ = \begin{bmatrix} V_k^+ \\ U_k^+ \end{bmatrix}$ and $\boldsymbol{\eta}_k^+ = \begin{bmatrix} \mathbf{g}_k^+ \\ \mathbf{f}_k^+ \end{bmatrix}$, we get the k step SGA decomposition:

$$AZ_k^+ = Y_k^+ H_k^+ + \boldsymbol{\eta}_k^+ \mathbf{e}_k^\top, \tag{44a}$$

$$BZ_k^+ = Y_k^+ R_k^+. \tag{44b}$$

$$(Q_k^+)^H Q_k^+ = (V_k^+)^H V_k^+ = I_k, \ (V_k^+)^H \mathbf{g}_k^+ = \mathbf{0}. \tag{44c}$$

Now, we present the IRSGA method and the IRRSGA method in the following algorithm.

Algorithm 5.1 The IRSGA/IRRSGA method

```
Input: $M, D, K \in \mathbb{C}^{n\times n}$ and $m \geq k \geq 1$.
Output: $k$ desired eigenpairs.
for $i = 1, 2, \ldots$ do
    Run the SGA algorithm (Algorithm 2.1) to generate an $m$th order SGA decomposition.
    Run the SGA method (Algorithm 3.1) or the RSGA method (Algorithm 4.1) to compute $k$
    candidates of Ritz pairs and check their convergence by (30) or (42).
    if #(convergent Ritz pairs) $\geq k$ then
        break
    else
        Select $p := m - k$ shifts $\vartheta_1, \ldots, \vartheta_p$.
        Let $\boldsymbol{\varepsilon} := \mathbf{e}_m^\top$ and $\boldsymbol{\eta} := \boldsymbol{\eta}_m$
        for $i = 1, \ldots, p$ do
            Compute unitary matrices $E_i$ and $F_i$ by the implicit-$QZ$ step with a single shift $\vartheta_i$ so
            that $E_i^H H_m F_i$ and $E_i^H R_m F_i$ are upper Hessenberg and upper triangular, respectively.
            Update $H_m := E_i^H H_m F_i$, $R_m := E_i^H R_m F_i$, $Z_m := Z_m F_i$, $Y_m := Y_m E_i$ and $\boldsymbol{\varepsilon} := \boldsymbol{\varepsilon} F_i$
        end for
        Set $\boldsymbol{\eta}_k := H_m(k+1,k) Y_m(:,k+1) + \boldsymbol{\varepsilon}(k+1)\boldsymbol{\eta}$
        Set $Z_k := Z_m(:,1:k)$, $Y_k := Y_m(:,1:k)$, $H_k := H_m(1:k,1:k)$, $R_k := R_m(1:k,1:k)$
    end if
end for
```

Remark 5.1

Note that applying an implicitly restarted process on a deflated SGA decomposition (25) may not yield a deflated SGA decomposition. We know that the Q-matrix and V-matrix of the SGA decomposition must adhere to one of the two orthogonality requirements: (1) all column vectors form an orthonormal set and (2) when deflation occurs, all column vectors form an orthonormal set except zero columns. In the first case, the resultant Q^+-matrix and V^+-matrix maintain the same orthogonality requirement as in the Q-matrix and V-matrix of the SGA decomposition. In the second case, both Q-matrix and V-matrix contain some zero column(s). Then the nonzero columns of the updated Q^+-matrix will be linearly dependent and the resultant decomposition is not a SGA decomposition. The same phenomenon occurs on the updated V^+-matrix.

To overcome this problem, we only need to perform column compression to make the updated Q^+-matrix and V^+-matrix of the forms $[\widehat{Q}^+ \ \mathbf{0}]$ and $[\widehat{V}^+ \ \mathbf{0}]$, simultaneously. On the other hand, it requires to update H^+-matrix and R^+-matrix by postmultiplying an upper triangular matrix as we perform the column compression. The resultant H^+-matrix and R^+-matrix are still upper Hessenberg form and upper triangular, respectively. Consequently, the column compression transforms a decomposition to a deflated SGA decomposition.

5.2. *The selection of shifts*

The above scheme involves selection of shifts $\vartheta_1, \ldots, \vartheta_{m-k}$. A good selection of shift is a key for success of the implicit restart technique. A popular choice of the shift values for IRA method [24] is to choose unwanted Ritz values, and are called exact shifts in [24]. When we solve the reduced QEP (27) to get $2m$ eigenvalues and select k Ritz values as approximations to the desired eigenvalues, we may directly use the reciprocal values of the remaining unwanted Ritz values as shifts which we also call them exact shifts. Among the selection of $2m - k$ shift candidates, we always take the reciprocal values of the $m - k$ unwanted Ritz values which are farthest from the target as shifts. Applying implicitly shifted QZ iteration with exact shifts to the SGA method, we have an implicitly restarted SGA (IRSGA) method.

For the RSGA method, we can also choose exact shifts. However, the refinement strategy can not only improve the accuracy of the Ritz pairs but also provide more accurate approximations to some of the unwanted eigenvalues. Suppose that $(\vartheta, \boldsymbol{\omega}) = (\vartheta, Q_m\boldsymbol{\zeta})$ is a Ritz pair of the QEP (1) which we are not interested in and the reciprocal of ϑ is one possible candidate of the shifts for the restarting process. Let $\boldsymbol{\omega}^+ = Q_m\boldsymbol{\zeta}^+$ be the refined Ritz vector corresponding the Ritz value ϑ as we discussed in section 4. Now, we illustrate how to find better shifts based on the unwanted refined Ritz vector $\boldsymbol{\omega}^+$. For an approximate eigenvector $\boldsymbol{\omega}$ of the QEP (1), the usual approach to deriving an approximate eigenvalue θ from $\boldsymbol{\omega}$ is to impose the Galerkin condition

$$(\theta^2 M + \theta D + K)\boldsymbol{\omega} \perp \boldsymbol{\omega}$$

and this follows that $\theta = \theta(\boldsymbol{\omega})$ must be one of the two solutions to the quadratic equation [10]

$$a_2\theta^2 + a_1\theta + a_0 = 0 \tag{45}$$

where $a_2 = \boldsymbol{\omega}^* M\boldsymbol{\omega}$, $a_1 = \boldsymbol{\omega}^* D\boldsymbol{\omega}$ and $a_0 = \boldsymbol{\omega}^* K\boldsymbol{\omega}$. Therefore, as we obtain the unwanted refined Ritz vector $\boldsymbol{\omega}^+$, (45) provides us one way to compute more accurate Ritz value beyond our interests and should be filtered in the restarting process. Since $\boldsymbol{\omega}^+ = Q_m\boldsymbol{\zeta}^+$, the coefficients corresponding to the quadratic equation (45) would be reduced as follows

$$a_2 = (\boldsymbol{\zeta}^+)^* M_m \boldsymbol{\zeta}^+, \quad a_1 = (\boldsymbol{\zeta}^+)^* D_m \boldsymbol{\zeta}^+ \quad \text{and} \quad a_0 = (\boldsymbol{\zeta}^+)^* K_m \boldsymbol{\zeta}^+ \tag{46}$$

where M_m, D_m and K_m is the projections of M, D and K onto the subspace $\operatorname{span}\{Q_m\}$ respectively as described in (28).

Hence, if ϑ_1^+ and ϑ_2^+ are roots of the quadratic equation (45) with coefficients defined in (46) then their reciprocal values would be better candidates for the restarting process. Consequently, if $(\vartheta_1, Q_m\zeta_1), \ldots, (\vartheta_p, Q_m\zeta_p)$ are p Ritz pairs that are farthest from our target and if $\vartheta_{i,1}^+, \vartheta_{i,2}^+$ are the roots of the quadratic equation (45) with respect to the unwanted refined Ritz vector $\boldsymbol{\omega}_i^+ = Q_m\boldsymbol{\zeta}_i^+$, $i = 1, \ldots, p$, then we choose the p values from $\vartheta_{1,1}^+, \vartheta_{1,2}^+, \ldots, \vartheta_{p,1}^+, \vartheta_{p,2}^+$ that are farthest from our target and take their reciprocal values as the shifts for the restarting process and call them the refined shifts. In our numerical examples, an implicitly restarted refined SGA (IRRSGA) method is a restart version of the RSGA method with refined shifts.

6. NUMERICAL EXAMPLES

The purpose of this section is to present a few numerical experiments to validate that the IRRSGA method is viable for solving the QEP (1). In addition, the examples demonstrate the superior

DOI: 10.1002/nla

properties of the IRSGA method and the IRRSGA method than the two versions of the IRA method [24] for solving the QEP where one IRA method is applied to the ℓ-SEP (6) and the other is applied to the r-SEP (7), respectively. The abbreviations ℓ-IRA and r-IRA are used to indicate that the IRA method is applied to ℓ-SEP and r-SEP, respectively.

In our examples, the number m denotes the order of the SGA/Arnoldi decomposition, k denotes the number of desired eigenpairs. The starting vector of the SGA method and the standard Arnoldi method are chosen as a vector with all components equal to 1 and the stopping tolerance for relative residuals is chosen to be $\mathrm{tol} = 10^{-14}$. The maximum number $r_{\max}$ of restarting process is set to be $r_{\max} = 30$.

Example 6.1
This example is obtained from "NLEVP: a collection of nonlinear eigenvalue problem" [3], namely "damped beam" arising from the vibration analysis of a beam simply supported at both ends and damped in the middle. In our MATLAB implementation, the command `nlevp('damped_beam',2000)` is used to construct real symmetric coefficient matrices M, D, K with $M = M^\top > 0$, $D = D^\top \geq 0$ and $K = K^\top > 0$. The matrix size is $n = 4,000$. Ten eigenvalues nearest the origin (i.e., $k = 10$) are computed by by four methods with $m = 20$. Figure 1(a) shows the maximum relative residuals of the ten desired eigenpairs computed by ℓ-IRA, r-IRA, IRSGA and IRRSGA with respect to iterations $1, 2, \ldots, 30$. We find that the maximum relative residuals computed by ℓ-IRA and r-IRA stagnate and those computed by the IRSGA method oscillate between 10^{-12} and 10^{-13}. All relative residuals of the desired eigenpairs computed by the IRRSGA method meet the stopping tolerance in 1 iteration. To investigate the convergence behaviors of the ten eigenpairs computed by ℓ-IRA, r-IRA, IRSGA and IRRSGA, we depict relative residual norms of the 1 step iteration in Figure 1(b). Compared to the IRSGA method, the refinement strategy of the IRRSGA method significantly improves the accuracy of computed eigenpairs even up to 5 digits for the eight computed eigenpairs that do not meet the convergence criterion. We report the number of iterations and CPU times in Table I. In summary, among the four methods the IRRSGA method is the only viable approach that accurately finds desired eigenpairs within 30 iterations.

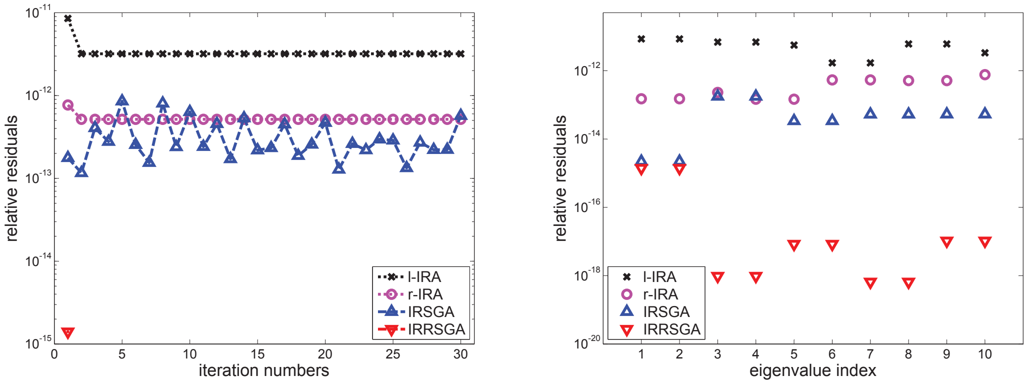
	#Its	CPU time
ℓ-IRA	30	32.8563
r-IRA	30	55.4423
IRSGA	30	38.3231
IRRSGA	1	7.3048

Table I. Iteration numbers and CPU times in Example 6.1.

Example 6.2
In Example 6.1, we see an amazing effect of the refinement strategy, i.e., ten wanted eigenpairs converge in one iteration before the restarting process with refined shifts in IRRSGA. This example illustrates that the refinement strategy with refined shifts introduced in section 5.2 for the IRRSGA method accelerate the convergence.

We consider the damped vibration mode of an acoustic fluid confined in a cavity with absorbing walls capable of dissipating acoustic energy [2]. The fluid domain $\Omega \subseteq \mathbb{R}^2$ is assumed to be polyhedral, and the boundary $\partial\Omega = \Gamma_A \cup \Gamma_R$, where the absorbing boundary Γ_A is the union of all the different faces of Ω and is covered by damping material. The rigid boundary Γ_R is the remaining part of $\partial\Omega$. Figure 2(a) gives an example of such a setup, where the top boundary is absorbing and

DOI: 10.1002/nla



(a) Evolution of relative residuals

(b) Relative residuals of the 1 step iteration

Figure 1. Example 6.1: Convergence histories for ℓ-IRA, r-IRA, IRSGA and IRRSGA.

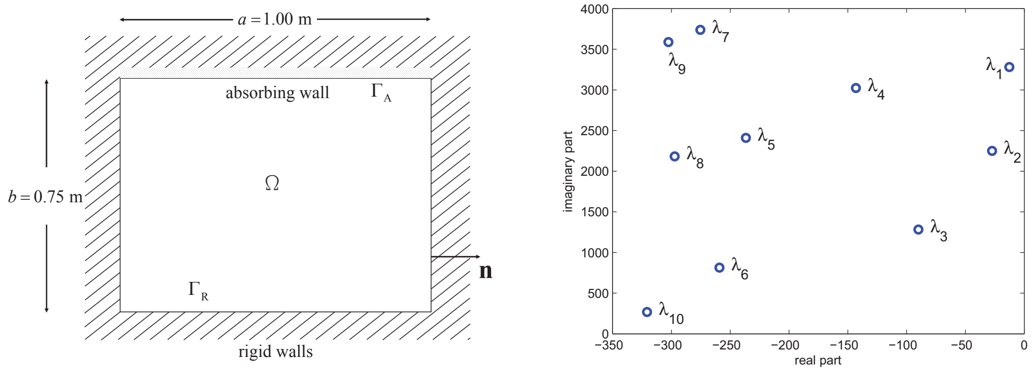
the remaining boundary is rigid. The equations characterizing the wave motion in Ω are

$$\begin{cases} \rho\frac{\partial^2 U}{\partial t^2} + \nabla P = \mathbf{0}, \quad P = -\rho c^2 \mathrm{div} U \\ P = \left(\alpha U \cdot \mathbf{n} + \beta \frac{\partial U}{\partial t} \cdot \mathbf{n}\right) \quad \text{on } \Gamma_A \\ U \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_R, \end{cases}$$

where the acoustic pressure P and the fluid displacement U depend on space $\mathbf{x}$ and time t, ρ is the fluid density, c is the speed of sound in air, $\mathbf{n}$ is the unit outer normal vector along $\partial\Omega$, and α, β are coefficients related to the normal acoustic impedance. The absorbing boundary on Γ_A indicates that the pressure is balanced by the effects of the viscous damping (the β term) and the elastic behavior (the α term). The model induces the following QEP

$$(\lambda^2 M_u + (\alpha + \lambda\beta)A_u + K_u)\mathbf{u} = \mathbf{0},$$

where M_u and K_u are mass and stiffness matrices, respectively, and A_u is used to describe the effect of the absorbing wall.



(a) Fluid in a cavity with one absorbing wall.

(b) The distribution of the ten desired eigenvalues.

Figure 2. Fluid in a cavity with one absorbing wall and the distribution of the ten desired eigenvalues.

In this example, we adopt the geometry illustrated in Figure 2(a) and use the following physical data: $\rho = 1\mathrm{kg/m}^3$, $c = 340\mathrm{m/s}$, $\alpha = 5 \times 10^4\mathrm{N/m}^3$, and $\beta = 200\mathrm{Ns/m}^3$. The same values are used in [2]. The rectangular domain is uniformly partitioned into 384 by 288 rectangles and each rectangle

DOI: 10.1002/nla

is further refined into two triangles. The dimension of coefficient matrices in this problem is $n = (3 \times 384 - 1) \times 288 = 331,488$. We compute ten analytic solutions of the desired eigenvalues $\lambda_1, \ldots, \lambda_{10}$ plotted in Figure 2(b) with the lowest positive vibration frequencies satisfying $0 < \frac{\mathrm{Im}(\lambda_i)}{2\pi} < 600$Hz. The order m is set to be $m = 20$. The shift target is taken by $\tau = -25 + 600\pi\mathtt{i}$, $\mathtt{i} = \sqrt{-1}$.

Table II shows that compared to the IRSGA method, the refinement strategy used in the IRRSGA method reduces the number of iterations and CPU time. Moreover, the IRRSGA method calculates ten desired eigenpairs in the smallest number of iterations and the shortest CPU time among four competitive methods.

	#Its	CPU time
ℓ-IRA	18	806.69
r-IRA	18	836.14
IRSGA	9	777.74
IRRSGA	7	735.38

Table II. Iteration numbers and CPU time in Example 6.2.

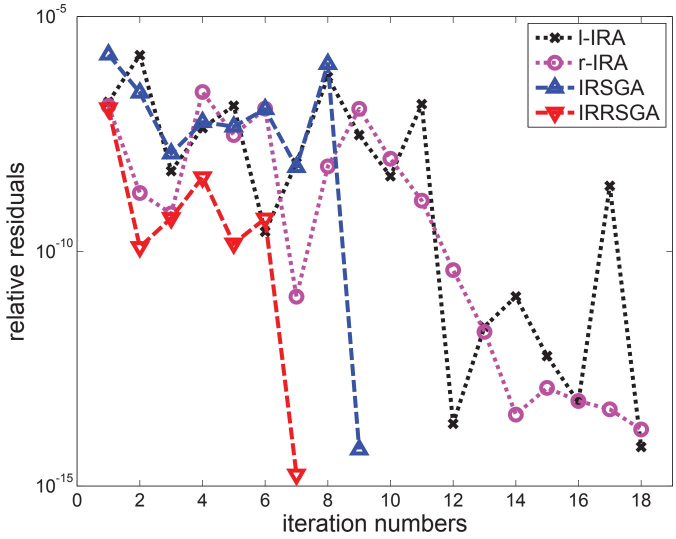


Figure 3. Example 6.2: Convergence histories for ℓ-IRA, r-IRA, IRSGA and IRRSGA.

Example 6.3

This experiment consists of six benchmark examples from the NLEVP [3]. In the following, we describe each example and the choice of parameters for generating the coefficient matrices of corresponding QEPs. All numerical results show that regardless of iteration numbers or CPU time, both IRSGA and IRRSGA appear to be more efficient and more competitive than the traditional Arnoldi methods ℓ-IRA and r-IRA. The standard Arnoldi methods can not calculate all desired eigenpairs in 30 iterations but our IRSGA and IRRSGA methods can effectively find all desired eigenpairs with high accuracy in less or around 10 iterations. The IRSGA and the IRRSGA methods have similar convergence behavior and the latter consumes a slightly more time than the former. This might be due to the fact that the IRSGA method converges in very few iterations. Figure 4 depicts the maximum of the k residual norms versus restarts and show the convergence processes of each

DOI: 10.1002/nla

example. Correspondingly, Table III lists the iteration numbers and the CPU time of each method for each example.

(a) **Acoustic 1D**. This example arises from the finite element discretization of the time harmonic wave equation $-\Delta p - (2\pi f/c)^2 p = 0$ [4]. Here p denotes the pressure, f is the frequency, c is the speed of sound in the medium, and ζ is the (possibly complex) impedance. On the domain $[0,1]$ with $c = 1$, the $n \times n$ matrices M, D and K are defined by

$$M = -4\pi^2 \tfrac{1}{n}(I_n - \tfrac{1}{2}\mathbf{e}_n\mathbf{e}_n^\top), \quad D = 2\pi\mathtt{i}\tfrac{1}{\zeta}\mathbf{e}_n\mathbf{e}_n^\top, \quad K = n\left(\mathrm{tridiag}(-1,2,-1) - \mathbf{e}_n\mathbf{e}_n^\top\right).$$

Observe that matrices M, K are real symmetric and D is complex symmetric. We use `nlevp(’acoustic_wave_1d’,5000,1)` to generate M, D, K with size $n = 5{,}000$ and compute the six eigenvalues nearest origin (i.e., $k = 6$) with $m = 12$.

(b) **Acoustic 2D**. This example is a two-dimensional acoustic wave equation [4] on $[0,1] \times [0,1]$. The coefficient matrices (M, D, K) are given by

$$\begin{aligned} M &= -4\pi^2 h^2 I_{q-1} \otimes (I_q - \tfrac{1}{2}\mathbf{e}_q\mathbf{e}_q^\top), \quad D = 2\pi\mathtt{i}\tfrac{h}{\zeta} I_{q-1} \otimes (\mathbf{e}_q\mathbf{e}_q^\top), \\ K &= I_{q-1} \otimes D_q + T_{q-1} \otimes (-I_q + \tfrac{1}{2}\mathbf{e}_q\mathbf{e}_q^\top), \end{aligned}$$

where h denotes the mesh size, $q = 1/h$, $\otimes$ denotes the Kronecker product, ζ is the (possibly complex) impedance, $D_q = \mathrm{tridiag}(-1,4,-1) - 2\mathbf{e}_q\mathbf{e}_q^\top \in \mathbb{R}^{q \times q}$ and $T_{q-1} = \mathrm{tridiag}(1,0,1) \in \mathbb{R}^{(q-1)\times(q-1)}$. We use `nlevp(’acoustic_wave_2d’,90,0.1*1i)` to get the real symmetric matrices (M, D, K). The matrix size is given by $n = 8{,}010$ and we compute six eigenvalues nearest origin (i.e., $k = 6$) with $m = 12$.

(c) **Concrete**. This problem arises from a model of a concrete structure supporting a machine assembly [6] and induces the QEP, $(\lambda^2 M + \lambda D + (1 + \mu\mathtt{i})K)\mathbf{x} = \mathbf{0}$, where M is real diagonal and low rank. D, the viscous damping matrix, is pure imaginary and diagonal, K is complex symmetric, and the factor $1 + \mu\mathtt{i}$ adds uniform hysteretic damping. We use `nlevp(’concrete’,0.04)` to generate the complex symmetric coefficient matrices. The matrix size $n = 2{,}472$ and we compute ten eigenvalues nearest the origin (i.e., $k = 10$) with $m = 20$.

(d) **Spring dashpot**. The QEP arises from a finite element model of a linear spring in parallel with Maxwell elements [9]. The mass matrix M is rank deficient and symmetric, the damping matrix D is rank deficient and block diagonal, and the stiffness matrix K is symmetric and has arrowhead structure. Matrices M, D, K are generated from `nlevp(’spring_dashpot’,7850,5000,0)` with size $n = 10{,}002$. We compute 50 eigenvalues nearest the origin (i.e., $k = 50$) with $m = 100$.

(e) **Wiresaw1**. We use `nlevp(’wiresaw1’,10000,0.01)` to generate the coefficient matrices of the gyroscopic QEP arising in the vibration analysis of a wiresaw [30]. Here M, D, K are $n \times n$ matrices defined by

$$M = \frac{1}{2} I_n, \quad D = -D^\top = [d_{ij}] \quad \text{and} \quad K = \operatorname*{diag}_{1 \le i \le n}\left(\frac{i^2\pi^2(1 - v^2)}{2}\right)$$

where $d_{ij} = \frac{4ij}{i^2 - j^2} v$ if $i + j$ is odd and, otherwise, $d_{ij} = 0$. The matrix size for this problem is $n = 10{,}000$ and we compute 10 eigenvalues nearest the origin (i.e., $k = 10$) with $m = 20$.

(f) **Wiresaw2**. When the effect of viscous damping is added to the problem in Wiresaw1, the corresponding QEP has the form [30]

$$(\lambda^2 M + \lambda(D + \eta I_n) + K + \eta D)\mathbf{x} = \mathbf{0}$$

where M, D and K are the same as in Wiresaw1 and η is a real nonnegative damping parameter. We take $\eta = 0.5$ and use `nlevp(’wiresaw2’,10000,0.01,0.5)` to generate the coefficient matrices. The matrix size is $n = 10{,}000$ and we compute 10 eigenvalues near the target -0.5 (i.e., $k = 10$ and $\tau = -0.5$) with $m = 20$.

(a) Acoustic 1D

	#Its	CPU time
ℓ-IRA	30	34.41
r-IRA	30	56.60
IRSGA	3	7.31
IRRSGA	3	7.47

(b) Acoustic 2D

	#Its	CPU time
ℓ-IRA	30	88.27
r-IRA	30	127.83
IRSGA	12	31.89
IRRSGA	11	27.45

(c) Concrete

	#Its	CPU time
ℓ-IRA	30	7.20
r-IRA	30	7.30
IRSGA	4	3.84
IRRSGA	4	4.06

(d) Spring dashpot

	#Its	CPU time
ℓ-IRA	30	907.47
r-IRA	30	1595.34
IRSGA	3	106.08
IRRSGA	3	114.98

(e) Wiresaw1

	#Its	CPU time
ℓ-IRA	8	69.80
r-IRA	4	75.24
IRSGA	2	35.83
IRRSGA	2	37.09

(f) Wiresaw2

	#Its	CPU time
ℓ-IRA	7	65.00
r-IRA	4	78.16
IRSGA	2	37.23
IRRSGA	2	39.39

Table III. Iteration numbers and CPU time in Example 6.3.

7. CONCLUSIONS

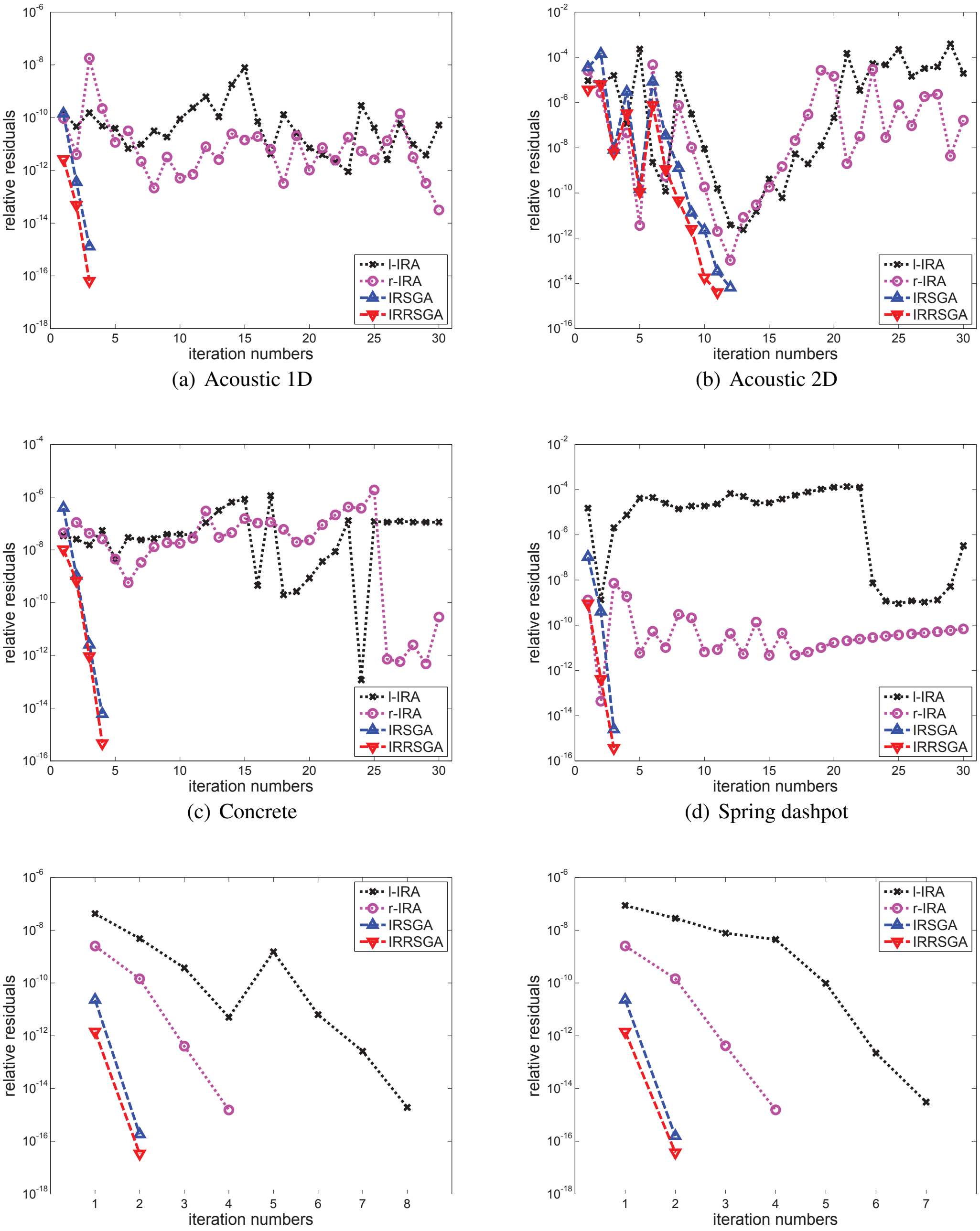
We have presented the SGA method, an orthogonal projection method, for solving QEPs based on a SGA decomposition. We have developed a practical algorithm to compute the SGA decomposition. The application of the SGA decomposition is three fold. First, we compute an orthonormal basis of the projection subspace in the SGA decomposition. Second, the SGA decomposition (8) has computational advantage for generating the coefficient matrices of reduced QEP (28). Third, we take advantage of the SGA decomposition to save some computational costs in the refinement process resulting a refined version of the SGA method abbreviated as the RSGA method for solving QEPs. After applying an implicit restart technique to SGA/RSGA methods, we have restart versions of SGA and RSGA, namely, the IRSGA/IRRSGA method. We have reported the numerical results on computation of the approximate eigenpairs with small eigenvalues in modulus. Compared to the standard IRA method, both the IRSGA method and IRRSGA method are superior in accuracy and convergence rate. We also see that the IRRSGA method may significantly improve the accuracy for obtaining the desired eigenpairs when the standard IRA method and the IRSGA method can not converge in a certain number of iterations.

ACKNOWLEDGEMENTS

This work is partially supported by the National Science Council, the Taida Institute for Mathematical Sciences, the National Center for Theoretical Sciences and the Center of Mathematical Modeling and Scientific Computing. The second author's work was supported by the National Science Foundation of China (No.11101080) and the Specialized Research Fund for the Doctoral Program of Higher Education (No.20110092120023).

REFERENCES

DOI: 10.1002/nla



(a) Acoustic 1D

(b) Acoustic 2D

(c) Concrete

(d) Spring dashpot

(e) Wiresaw1

(f) Wiresaw2

Figure 4. Example 6.3: Convergence histories for ℓ-IRA, r-IRA, IRSGA and IRRSGA.

1. Bai Z, Su Y. SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM Journal on Matrix Analysis and Applications* 2005; **26**(3):640–659.
2. Bermúdez A, Durán RG, Rodríguez R, Solomin J. Finite element analysis of a quadratic eigenvalue problem arising in dissipative acoustics. *SIAM Journal on Numerical Analysis* 2000; **38**(1):267–91.
3. Betcke T, Higham NJ, Mehrmann V, Schroder C, Tisseur F. NLEVP: a collection of nonlinear eigenvalue problems. Available from: http://www.mims.manchester.ac.uk/research/numerical-analysis/nlevp.html.
4. Chaitin-Chatelin F, van Gijzen MB. Analysis of parameterized quadratic eigenvalue problems in computational acoustics with homotopic deviation theory. *Numerical Linear Algebra with Applications* 2006; **13**:487–512.
5. Datta BN. *Numerical Linear Algebra and Applications* (2nd edn). SIAM: Philadelphia, PA (2010).

6. Feriani A, Perotti F, Simoncini V. Iterative system solvers for the frequency analysis of linear mechanical systems. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**:1719–1739.
7. Flaschka U, Lin WW, Wu JL. A KQZ algorithm for solving linear-response eigenvalue equations. *Linear Algebra and its Applications* 1992; **165**:93–123.
8. Gohberg I, Lancaster P, Rodman L. Matrix Polynomials. Academic Press: New York, 1982.
9. Gotts A. Report regarding model reduction, model compaction research project. Manuscript, University of Nottingham, February 2005.
10. Hochstenbach ME, van der Vorst HA. Alternatives to the Rayleigh quotient for the quadratic eigenvalue problem. *SIAM Journal on Scientific Computing* 2003; **25**(2):591–603.
11. Huang TM, Lin WW, Qian J. Structure-preserving algorithms for palindromic quadratic eigenvalue problems arising from vibration of fast trains. *SIAM Journal on Matrix Analysis and Applications* 2009; **30**(4):1566–1592.
12. Huitfeldt J, Ruhe A. A new algorithm for numerical path following applied to an example from hydrodynamical flow. *SIAM Journal on Scientific Computing* 1990; **11**(6):1181–1192.
13. Jia Z. Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems. *Linear Algebra and its Applications* 1997; **259**:1–23.
14. Jia Z, Stewart GW. An analysis of the Rayleigh-Ritz method for approximating eigenspaces. *Mathematics of Computation* 2001; **70**(234):637–647.
15. Jia Z, Sun Y. A Refined Second-order Arnoldi (RSOAR) Method for the Quadratic Eigenvalue Problem and Implicit Restarting. Available from: http://arxiv.org/abs/1005.3947, 2010.
16. Lancaster P. Lambda-Matrices and Vibrating Systems. Pergamon Press: Oxford, 1966.
17. Li RC, Ye Q. A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems. *SIAM Journal on Matrix Analysis and Applications* 2003; **25**(2):405–428.
18. Lin Y, Bao L. Block second-order Krylov subspace methods for large-scale quadratic eigenvalue problems. *Applied Mathematics and Computation* 2006; **181**(1):413–422.
19. Meerbergen K. Locking and restarting quadratic eigenvalue solvers. *SIAM Journal on Scientific Computing* 2001; **22**(5):1814–1839.
20. Moler CB, Stewart GW. An algorithm for generalized matrix eigenvalue problems. *SIAM Journal on Numerical Analysis* 1973; **10**(2):241–256.
21. Neumaier A. Residual inverse iteration for the nonlinear eigenvalue problem. *SIAM Journal on Numerical Analysis* 1985; **22**(5):914–923.
22. Sleijpen GLG, Booten AGL, Fokkema DR, van der Vorst HA. Jacobi–Davidson type methods for generalized eigenproblems and polynomial eigenproblems. *BIT Numerical Mathematics* 1996; **36**(3):595–633.
23. Sleijpen GLG, van der Vorst HA, van Gijzen M. Quadratic eigenproblems are no problem. *SIAM News* 1996; **29**(7):8–9.
24. Sorensen DC. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM Journal on Matrix Analysis and Applications* 1992; **13**(1):357–385.
25. Sorensen DC. Truncated QZ methods for large scale generalized eigenvalue problems. *Electronic Transactions on Numerical Analysis* 1998; **7**:141–162.
26. Stewart GW. Matrix Algorithms, Volume II: Eigenvalues. SIAM: Philadelphia, PA, 2001.
27. Tisseur F. Backward Error and Condition of Polynomial Eigenvalue Problems. *Linear Algebra and its Applications* 2000; **309**:339–361.
28. Tisseur F, Meerbergen K. The Quadratic Eigenvalue Problem. *SIAM Review* 2001; **43**(2):235–286.
29. Voss H. An Arnoldi method for nonlinear eigenvalue problems. *BIT Numerical Mathematics* 2004; **44**:387–401.
30. Wei S, Kao I. Vibration analysis of wire and frequency response in the modern wiresaw manufacturing process. *Journal of Sound Vibration* 2000; **231**(5):1383–1395.
31. Wang B, Su Y, Bai Z. The second-order biorthogonalization procedure and its application to quadratic eigenvalue problems. *Applied Mathematics and Computation* 2006; **172**(2):788–796.
32. Ye Q, An iterated shift-and-invert Arnoldi algorithm for quadratic matrix eigenvalue problems. *Applied Mathematics and Computation* 2006; **172**(2):818–827.