國立交通大學

應用數學系

博士論文

二次暨有理特徵值問題中高效能 Arnoldi 型態演算法

Efficient Arnoldi-Type Algorithms for Quadratic and Rational Eigenvalue Problems

研 究 生：黃韋強

指導教授：林文偉 教授

中華民國一〇一年八月

二次暨有理特徵值問題中高效能 Arnoldi 型態演算法

Efficient Arnoldi-Type Algorithms for Quadratic and Rational Eigenvalue Problems

研究生：黃韋強　　Student：Wei-Qiang Huang

指導教授：林文偉　　Advisor：Wen-Wei Lin

國立交通大學應用數學系

博士論文

A Thesis

Submitted to Department of Applied Mathematics

College of Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics

August 2012

Hsinchu, Taiwan, Republic of China

中華民國一O一年八月

二次暨有理特徵值問題中高效能 Arnoldi 型態演算法

學生：黃韋強　　　　　　　　　　　　　　　　　　指導教授：林文偉　教授

國立交通大學應用數學系博士班

摘　　要

本論文探討求解二次特徵值問題及有理特徵值之高效能 Arnoldi 型態演算法。其研究主題可分為兩部分：（一）流固系統中非線性特徵值問題之 Arnoldi 型態演算法之比較；（二）求解二次特徵值問題中的半正交廣義 Arnoldi 法。

我們探討並分析一個具有耗散聲能吸音牆密閉空間中聲場的阻尼振動模態。利用有限元素法，我們可由位移場的棱邊離散化將問題轉變為一個求解二次特徵值問題。另一方面，若考慮壓力節點的離散則會獲得一個有理特徵值問題。透過線性化的技巧，我們可將這兩個非線性特徵值問題分別改寫成型態為 $\mathcal{A}\mathbf{x}=\lambda\mathcal{B}\mathbf{x}$ 的廣義特徵值問題。該問題可以用 Arnoldi 演算法處理兩種不同型態係數矩陣，$\mathcal{B}^{-1}\mathcal{A}$ 及 $\mathcal{A}\mathcal{B}^{-1}$，的標準特徵值問題。數值結果顯示利用 Arnoldi 法求解 $\mathcal{A}\mathcal{B}^{-1}$ 具有較高的精準度。

對於求解二次特徵值問題中絕對值較靠近零之特徵值所對應的特徵對，我們發展了一個正交投影法－半正交廣義 Arnoldi 法。此外，我們更進一步提出可精化、可重啟動的半正交廣義 Arnoldi 法。相較於將二次特徵值問題線性化後再利用傳統隱式重啟動 Arnoldi 法求解，數值實驗顯示隱式重啟動半正交廣義 Arnoldi 法(不論是否有精化過程）具有極佳的收斂行為。

Efficient Arnoldi-Type Algorithms for Quadratic and Rational Eigenvalue Problems

student：Wei-Qiang Huang Advisors：Dr. Wen-Wei Lin

Department of Applied Mathematics
National Chiao Tung University

ABSTRACT

In this dissertation, we consider two themes related to Arnoldi-type algorithms for solving nonlinear eigenvalue problems.

We develop and analyze efficient methods for computing damped vibration modes of an acoustic fluid confined in a cavity, with absorbing walls capable of dissipating acoustic energy. The edge-based finite elements for the displacement field results in a quadratic eigenvalue problem. On the other hand, the discretization in terms of pressure nodal finite elements results in a rational eigenvalue problem. We use the linearization technique to transform these nonlinear eigenvalue problems, respectively, into generalized eigenvalue problems $\mathcal{A}\mathbf{x} = \lambda\mathcal{B}\mathbf{x}$ and apply Arnoldi algorithm to two different types of single matrices $\mathcal{B}^{-1}\mathcal{A}$ and $\mathcal{A}\mathcal{B}^{-1}$. Numerical accuracy shows that the application of Arnoldi on $\mathcal{A}\mathcal{B}^{-1}$ is better than that on $\mathcal{B}^{-1}\mathcal{A}$.

For computing a few eigenpairs with smallest eigenvalues in absolute value of quadratic eigenvalue problems, we develop the semiorthogonal generalized Arnoldi method, an orthogonal projection technique. Furthermore, we propose refinable and restartable variations of this method to improve the accuracy and efficiency. 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.

誌謝

本篇論文得以完成，首先得感謝我的指導教授，林文偉教授。謝謝老師您近四年來對我的栽培、鼓勵、鞭策、體諒與照顧。您對學術研究的熱誠、堅持、洞察力與創造力始終是我敬佩及效仿的對象。儘管我並無太多數值計算的背景知識，您仍然在旁一步步引領著我進入這豐富的研究領域。當面臨學術或人生所遭遇的瓶頸與問題，您所分享的研究心得與人生歷練提供我更多不同的思考角度。您對我生活上的幫助更是讓我無顧之憂的學習。能在您的教導下完成學位論文，是一件榮耀的事。接著，我要感謝口試委員林松山教授、朱景華教授、王振男教授和王偉仲教授，於學位論文口試時的提問及建議。特別感謝林松山教授協助我申請研發替代役讓我得以逐步邁向下個階段中的學習規劃。

謝謝泓勳、育豪、麻將、青松、偉碩、德軒、柏任、函恩、芷瑄、昌翰、劭芃、安怡、美亨、淑如、明誠學長、Apostol 學長、恭儉學長、文貴學長、明杰學長、光暉學長、建綸學長、瑜堯、郁傑、建智學長、其棟學長、宏椽（清章）學長等在我五年博士求學階段中，一同奮鬥過的學長姐、同學夥伴以及學弟妹。遇見你們並一起在交大奮鬥是一件令人開心與值得回味的事。

感謝王辰樹教授、黃聰明教授、李勇達教授、李宗錂教授、張書銘教授、郭岳承教授、蔣俊岳教授、黃建智博士、林敏雄教授、吳金典教授、王偉仲教授、黃印良教授等 Lin Group 的前輩在我求學這段期間帶著我了解、剖析問題，並與我分享心情和各種大小趣事。此外我得感謝於交大求學這段時間內，每個教導過我或一同參與研究過程的各位老師，包括莊重教授、賴明治教授、王夏聲教授、李明佳教授、吳慶堂教授、盧鴻興教授、周所向教授、張企教授、陳泰賓教授等。同時我也得感謝各位系助理：陳盈吟小姐、張麗君小姐、小張姐、崔妮臻小姐、宋雅鈴小姐、Billing、李珍韶學姐等在這段時間內，於公於私對我的幫助。

感謝父親黃丁福先生，母親許錦雀女士以及可愛的妹妹春萍、于娗。謝謝父母養我、育我，並讓我能任性的完成學業。請允許我目前以一張文憑回報你們對我最無私的關心、鼓勵與支持。

接著，我要謝謝我的女友蕙甄。感謝妳一直以來的陪伴，並體諒我將大多數的時間投入於研究與學習的無底洞中。謝謝妳陪著我經歷每個無助、徬徨、失望、開心與平凡的日子。家人與妳是支持我一路來最大的力量也是我最後的避風港。

誠如林文偉教授常告誡我們的一句話：「拿到博士學位沒什麼了不起，學術研究的一切才剛要開始」。也許迎接我的未來充滿太多的未知數，但我將滿懷感恩，不怕苦、不怕難，心甘情願地往前走下去。由衷地感謝每個關心我的大家，謝謝你們。

Qiang 2012/08/31 @ NCTU

Contents

List of Figures

List of Tables

List of Algorithms

1

Introduction and Preliminaries

Contents

The theme explored in this thesis is to develop and exploit efficient Arnoldi-type methods to solve the quadratic and rational eigenvalue problems. This chapter will briefly introduce some basic notions, mathematical notations and conventional methods of the so-called “eigenvalue problems”. We then, in Chapter 2, develop and analyze efficient methods for quadratic and rational eigenvalues arising from computing damped vibration modes of an acoustic fluid confined in a cavity with absorbing walls capable of dissipating acoustic energy. In Chapter 3, we will propose an orthogonal projection method for solving quadratic eigenvalue problems. Finally, conclusions and the future work of this thesis will be discussed in Chapter 4.

1.1 Notations

The following notations are frequently used in this thesis. Other notations will be clearly defined whenever they are used.

- $\mathtt{i} = \sqrt{-1}$.
- We use the symbol $\forall$ to mean ‘for all’ throughout the thesis.
- $\mathbb{R}$ denotes the set of real numbers and $\mathbb{C}$ denotes the set of complex numbers.
- $\mathrm{Re}(\lambda)$ and $\mathrm{Im}(\lambda)$, respectively, denote the real part and the complex part of the scalar $\lambda \in \mathbb{C}$.
- $\mathbf{0}$ denotes zero vectors and matrices with appropriate size.
- I_n denotes the $n \times n$ identity matrix.
- $\mathbf{e}_j$ denotes the jth column of the identity matrix I_n with specified n.
- We use $\cdot^\top$ and $\cdot^H$ to denote the transpose and conjugate transpose for vectors or matrices.

- $\otimes$ denotes the Kronecker product.

- $\|\cdot\|_2$, $\|\cdot\|_F$ and $\|\cdot\|_\infty$ respectively denote the 2-norm, Frobenius norm and infinity norm for vectors or matrices.

- 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;

  $A(i:j,:)$ denotes the rows of A from i to j and $A(:,k:\ell)$ denotes the columns of A from k to ℓ.

1.2 The Arnoldi Method for Standard Eigenvalue Problems

Given a large sparse matrix $A \in \mathbb{C}^{n\times n}$, the Arnoldi method [1] is a well known and very prevalent algorithm for solving the so-called standard eigenvalue problem (SEP)

$$A\mathbf{x} = \lambda\mathbf{x}. \tag{1.1}$$

That is, to find a scalar λ (real or complex) and a nonzero n-vector $\mathbf{x}$ satisfying the equations (1.1). In this case, we say that λ is an eigenvalue of A and $\mathbf{x}$ is called an eigenvector of A with respect to λ. Moreover, the pair $(\lambda, \mathbf{x})$ is said to be an eigenpair of A.

Starting with a unit vector $\mathbf{v}_1$, the Arnoldi method successively constructs a sequence of unitary vectors $\mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_m$ which forms a unitary basis of the Krylov subspace $\mathcal{K}_m(A, \mathbf{v}_1) \equiv \mathrm{span}\{\mathbf{v}_1, A\mathbf{v}_1, \ldots, A^{m-1}\mathbf{v}_1\}$ with $m \ll n$ such that

$$\begin{cases} h_{j+1,j}\mathbf{v}_{j+1} = A\mathbf{v}_j - \sum\limits_{i=1}^{j} h_{ij}\mathbf{v}_i, \quad j = 1, 2, \ldots, m, \\ \mathbf{v}_s^H \mathbf{v}_t = 0, \ \forall s \neq t \quad \text{and} \quad \mathbf{v}_s^H \mathbf{v}_s = 1, \ \forall s, \end{cases}$$

or equivalently,

$$\begin{cases} AV_m = V_m H_m + h_{m,m+1}\mathbf{v}_{m+1}\mathbf{e}_m^\top, \\ \begin{bmatrix} V_m^H \\ \mathbf{v}_{m+1}^H \end{bmatrix} [V_m \ \mathbf{v}_{m+1}] = \begin{bmatrix} I_m & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \end{cases} \tag{1.2}$$

where V_m is an $n \times m$ matrix with column vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$, H_m is an $m \times m$ upper Hessenberg matrix. After building the factorization (1.2), called the Arnoldi decomposition, we then reduce A into the upper Hessenberg H_m through the unitary transformation $V_m^H A V_m = H_m$. The eigenvalues and corresponding eigenvectors of the reduced SEP $H_m\mathbf{z} = \mu\mathbf{z}$ can be solved by the classical eigenvalue techniques, such as the QR algorithm (also named the Francis algorithm [18, 19]). Moreover, we see that if $(\theta, \mathbf{y})$ is an eigenpair of H_m then $(\theta, V_m\mathbf{y})$ is called a Ritz pair of A – an approximate eigenpair of A with the residual norm

$$\|(A - \theta I_n)V_m\mathbf{y}\| = |h_{m+1,m}||\mathbf{e}_m^\top\mathbf{y}|.$$

For more details on the practical realization and theoretical analysis of the Arnoldi method, we refer to [2, 14, 22, 49, 54, 67, 73].

There are some variations of the Arnoldi method. In practice, a small number of eigenvalues that are nearest to a target σ or located in a prescribed region of the complex plane and the corresponding eigenvectors are often of interest. Under the assumption that σ is not an eigenvalue of the SEP (1.1) but not to far away from the wanted eigenvalues, the shift-and-invert Arnoldi method [48, 54] tends to solve the transformed eigenproblem

$$(A - \sigma I_n)^{-1}\mathbf{x} = \nu\mathbf{x}, \tag{1.3}$$

where the scalar value σ is called a shift. It is easy to verify that (1.3) and (1.1) are mathematically equivalent since $(\nu, \mathbf{x})$ is an eigenpair of (1.3) if and only if $(\sigma + \frac{1}{\nu}, \mathbf{x})$ is an eigenpair of (1.1).

The restarted Arnoldi method aims to overcome the increasing storage as well as the computational cost of the Arnoldi decomposition (1.2) as m is increasing. In [53], Saad coped with these difficulties by developing the explicitly restarted Arnoldi iteration. The idea of this strategy is to compute another mth order Arnoldi decomposition with a "better" initial vector which is a linear combination of some wanted Ritz vectors. The implicitly restarted Arnoldi method [59] and Krylov-Schur algorithm [28, 61, 63], on the other hand, are two remarkable implicitly restarting schemes. These schemes are called *implicit* due to the fact that the initial vector is sequentially constructed by using the implicitly shifted QR algorithm [18, 19] on the Hessenberg matrix H_m in (1.2). We will review the implementation of the Krylov-Schur restarting in Section 2.4.

Another possible problem is that even though some desirable eigenvalues computed by the Arnoldi method already attempt to converge, the corresponding approximate eigenvectors may converge very slowly and even fail to converge. The refined Arnoldi method [32] proposed by Jia gave an alternative approach to remedy this problem by computing refined approximate eigenvectors. See also [33]. We will mimic this idea and design a refinement strategy for our Arnoldi-type method in Section 3.4. Other variations of the Arnoldi method include the block-Arnoldi method [55], the inexact Arnoldi method [56], the residual Arnoldi method [37, 38], and so on.

1.3 The Generalized Arnoldi Method for Generalized Eigenvalue Problems

The generalized eigenvalue problem (GEP) for the matrix pencil $A - \lambda B$ of two square matrices A and B with size n is to determine scalars λ and n-vectors $\mathbf{x} \neq \mathbf{0}$ such that

$$A\mathbf{x} = \lambda B\mathbf{x}. \tag{1.4}$$

If B is nonsingular, the GEP (1.4) can be transformed into SEPs

$$(B^{-1}A)\mathbf{x} = \lambda\mathbf{x} \tag{1.5}$$

or

$$(AB^{-1})\mathbf{y} = \lambda\mathbf{y}, \quad \mathbf{y} = B\mathbf{x} \tag{1.6}$$

and subsequently solved by the standard Arnoldi method. Alternatively, the QZ algorithm [45], an analog of the QR algorithm for the GEP, is the method of choice for dealing with the GEP (1.4) with small dense coefficient matrices.

The truncated QZ method proposed by Sorensen [60] is one of the approaches for solving large-scale GEPs. For $m \ll n$, this method constructs a generalization of the standard Arnoldi decomposition (1.2),

$$\begin{cases} AZ_m = Y_m H_m + h_{m+1,m}\mathbf{y}_{m+1}\mathbf{e}_m^\top, \\ BZ_m = Y_m R_m, \\ Z_m^H Z_m = I_m, \ Y_m^H Y_m = I_m, \ Y_m^H \mathbf{y}_{m+1} = \mathbf{0}, \end{cases} \tag{1.7}$$

which is called the generalized Arnoldi reduction in [60], and deals with the small-sized GEP $H_m\mathbf{v} = \mu R_m\mathbf{v}$ of the $m \times m$ upper Hessenberg-triangular pair (H_m, R_m) to approximate eigenpairs of the original large-scale GEP (1.4).

1.4 Quadratic Eigenvalue Problems and Linearizations

In this section, we consider the quadratic eigenvalue problem (QEP) of the form

$$Q(\lambda)\mathbf{x} \equiv (\lambda^2 M + \lambda D + K)\mathbf{x} = \mathbf{0}, \tag{1.8}$$

where M, D and K are $n \times n$ large and sparse matrices. The QEP is a special case of the polynomial eigenvalue problem (PEP)

$$P(\lambda)\mathbf{x} \equiv \left(\sum_{i=0}^{d} \lambda^i P_i\right)\mathbf{x} = \mathbf{0}, \tag{1.9}$$

where P_i are constant matrices of size n for $1 \leq i \leq d$. $P(\lambda) \equiv \sum\limits_{i=0}^{d} \lambda^i P_i$ is called a matrix polynomial (in λ) of degree d. Obviously, for $d = 0, 1$ and 2, the PEP (1.9) is, respectively, indeed the case of SEP (1.1), GEP (1.4) and QEP (1.8).

The "linearization" is a typical and most widely used technique to solve the QEP in which the problem is reformulated into a linear one which doubles the order of the system. By selecting suitable matrices $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{2n\times 2n}$ and the vector $\boldsymbol{\varphi} \in \mathbb{C}^{2n}$, we can convert (1.8) into the GEP

$$(\mathcal{A} - \lambda\mathcal{B})\boldsymbol{\varphi} = \mathbf{0} \tag{1.10}$$

satisfying the relation

$$\mathcal{E}(\lambda)(\mathcal{A} - \lambda\mathcal{B})\mathcal{F}(\lambda) = \begin{bmatrix} Q(\lambda) & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix},$$

where $\mathcal{E}(\lambda)$ and $\mathcal{F}(\lambda)$ are $2n \times 2n$ matrix polynomials in λ with constant nonzero determinants. In this case,

$$\det(\mathcal{A} - \lambda\mathcal{B}) = \det(\lambda^2 M + \lambda D + K)$$

indicates the eigenvalues of the original QEP (1.8) coincide with the eigenvalues of the enlarged GEP (1.10). As a result, the linearization technique of QEPs makes classical methods for GEPs as well as SEPs can be used.

There are many choices of $(\mathcal{A}, \mathcal{B})$'s, but probably the most popular ones in practice are the so-called companion forms [21]: the first companion form

$$\mathcal{A} = \begin{bmatrix} -D & -K \\ I_n & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathcal{B} = \begin{bmatrix} M & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix}$$

as well as the second companion form

$$\mathcal{A} = \begin{bmatrix} -D & I_n \\ -K & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathcal{B} = \begin{bmatrix} M & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix}. \tag{1.11}$$

There are some drawbacks, however, of the linearization technique to solve QEPs. For instance, the doubling size of the problem increases the computational cost and the original structures of the coefficient matrices (M, D, K) such as symmetry and positive definiteness may be lost. To circumvent these drawbacks, one may expect to solve the QEP (1.8) directly. The QEP is projected onto a properly chosen low-dimensional subspace in order to lower the matrix sizes of the coefficient matrices in (1.8). The reduced QEP can then be solved by a standard approach for dense matrices. Methods of this type include the residual iteration method [27, 43, 47], the Jacobi-Davidson method [57, 58], a Krylov-type subspace method [40], the nonlinear Arnoldi method [68], the second-order Arnoldi method [3, 41, 72] and an iterated shift-and-invert Arnoldi method [75]. While these methods use a similar projection process, the main difference between them is the selection of projection subspaces. Convergence analysis of projection methods to approximate eigenpairs of the QEP (1.8) has recently been invented in [29].

In Chapter 2, we consider a QEP arising from a finite element model and convert it into SEPs in (1.5) and (1.6) through an equivalent second companion form (1.11). We will report theoretical and numerical comparisons of these two SEPs (1.5) there. In Chapter 3, we combine the generalized Arnoldi reduction (1.7) and the second companion form linearization (1.11) of the QEP (1.8) to develop a projection method to solve the QEP (1.8) directly.

1.5 Rational Eigenvalue Problems and the Trimmed Linearization

The rational eigenvalue problem (REP) concerns the problem of finding $(\lambda, \mathbf{x})$ with $\mathbf{x} \neq \mathbf{0}$ satsifying the equation

$$R(\lambda)\mathbf{x} \equiv \left(P(\lambda) - \sum_{j=1}^{r} \frac{s_j(\lambda)}{t_j(\lambda)} C_j \right) \mathbf{x} = \mathbf{0}, \tag{1.12}$$

where $P(\lambda)$ is an $n \times n$ matrix polynomial in λ, $s_j(\lambda)$ and $t_j(\lambda)$ are scalar polynomials in λ, and C_j are $n \times n$ constant matrices. The simulation of the three-dimensional pyramid quantum dot heterostructure [31] produces a REP. For more examples related to the REPs, see [9, 44, 64].

To solve the REP (1.12), one may immediately multiply (1.12) by the scalar polynomial $\prod_{j=1}^{r} t_j(\lambda)$ to convert it into a PEP. Subsequently, the PEP can be linearized to a GEP. The nonlinear eigensolver, on the other hand, is another way to solve this problem. The nonlinear Jacobi-Davidson method [70] and the nonlinear Arnoldi method [68] fall into this category. Yet, these approaches also have the problem that restricts advantages of the underlying matrix structures and properties of REPs.

Trimmed linearization [64] is a recent linearization-based approach to solve the REP (1.12), especially when matrices C_j in (1.12) have the low-rank property. This method utilizes and preserves the structure and property of the REP (1.12) as much as possible to transform it into a GEP (and hence a SEP), and it only slightly increases the size of the GEP, compared to the size of the original REP (1.12).

The REP discussed in this thesis is a quadratic matrix polynomial ($P(\lambda)$ in (1.9) with $d = 2$) together with low-rank rational terms (see Eq. (2.20)). We will use the trimmed linearizing skill and the shift-and-invert Arnoldi method with Krylov-Schur restarting to detect desired eigenpairs.

2

Arnoldi-Type Algorithms for Nonlinear Eigenvalue Problems Arising in Fluid-Solid Systems

Contents

2.1 Introduction

Efficient and correct computation of the damped vibration modes generated by an inviscid, compressible, barotropic fluid in a cavity, with absorbing walls is an important issue when for example one is interested in decreasing the level of noise in aircraft or cars. In general, one needs first a mathematical model consisted of partial differential equations with proper boundary and initial conditions. After this first phase of mathematical formulation, the next phase is to find efficient methods to compute the modes. This phase involves correct discretization of the mathematical formulation and computation of large scale nonlinear eigenvalue problems, be it quadratic, cubic, or even rational. Choosing correct discretization schemes to avoid spurious modes and finding efficient methods to locate eigenvalues that lie in the interior of the spectrum are among important issues to deal with. In the mathematical formulation phase, we have interaction between the fluid and structure (cavity walls), and the displacement variable natural for the solid could be chosen for the fluid as well so that compatibility and equilibrium (cf. (2.3) and (2.7) below) through the fluid-solid interface can be satisfied automatically. A drawback lurking behind the displacement formulation is the possible presence of nonphysical zero-frequency spurious circulation modes, if one is not careful in choosing the discretization scheme associated with the underlying partial differential system. For example discretization by standard finite elements or finite differences often exhibit such a phenomenon. Approaches circumventing this drawback can be found in [4, 12, 20, 24, 71], among others.

One of the discretizations we will be using in this chapter is the edge-based or Raviart-Thomas finite elements for the displacement field, following [5, 7]. The main concerns in [5, 51] are pure mathematical issues of proving that their numerical approximation is free of spurious modes and has second order convergence rate.

Efficient computation of the modes is not a concern, as they solved the associated quadratic eigenvalue problem by the standard eigensolver `eigs` from MATLAB that employs Arnoldi iterations.

In this chapter our primary concern is to develop and study efficient eigensolvers for the spectral approximation of the damped vibration modes. Two approximations are investigated, one constructed from the edge-based displacement space (cf. Eq. (2.11) below), which results in quadratic eigenvalue problems (QEPs) and one from the node-based pressure space (cf. Eq. (2.12)), which results in rational eigenvalues problems (REPs). Our first approximation is identical to that in [5, 7], but we further develop efficient methods for solving the associated QEP. However, we show in Section 2.2 that this problem has a large zero-frequency or null space and this fact may influence the efficiency of Arnoldi-type algorithms. Motivated by this, we extensively explore the second approximation of using the pressure space, which has a much smaller eigenvalue system to solve and which has a one dimensional null space. While there is an extensive literature on QEPs problems [66], REPs are much less studied [64, 68, 69]. Although on the surface the REP (Eq. (2.20)) could be turned into a cubic one by multiplying out the denominator, we will preserve its rational structure and design efficient methods to numerically solve it in Section 2.3.

The organization of this chapter is as follows. We describe the underlying model fluid-solid problem of this chapter in Section 2.2, where the edge-based displacement approximation and the node-based pressure approximation are derived. We pay particular attention to identifying the dimension of the associated null space, which may influence performance of the numerical method introduced later. In Section 2.3, we use the general strategy of turning a nonlinear eigenvalue problem into a standard one by some sort of linearization techniques. We then apply the Arnoldi type algorithms to solve it. For the two nonlinear eigenvalue problems, the QEP is as usual turned into a generalized eigenvalue problem (GEP), from which two types of standard eigenvalue problems (SEP) (2.19.1) and (2.19.2) are derived. The REP is trimmed-linearized into two types of three by three block SEPs (2.31.1) and (2.31.2). The important issue of residual error bound analysis is addressed here. We then apply Arnoldi method with Schur-restarting described in Section 2.4 to the resulting SEPs. The important issues of stopping criteria and computational costs for applying Arnoldi method to the QEP and REP are also derived in this section. In Section 2.5, we present numerical results and evaluate the merits of the schemes involved where we also demonstrate the role of normwise scaling in preprocessing the eigenvalue problems. Summaries are included in Section 2.6.

2.2 The Model Problem

Let us consider a simple model of a rigid container filled with an inviscid compressible barotropic fluid and its acoustic energy is absorbed through a thin layer of a viscoelastic material applied to some or all of its walls. For simplicity we assume the fluid domain $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) 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 Γ. An example of the setup is in Figure 2.1(i) on Section 2.5, where the top boundary is absorbing and the remaining boundary is rigid.

The dynamic variables of our model problem are the fluid pressure P and the displacement field $\mathbf{U}$, which satisfy ([8, 35])

$$\begin{aligned} \rho\frac{\partial^2\mathbf{U}}{\partial t^2} &+ \nabla P = \mathbf{0} && \text{in } \Omega, && (2.1)\\ P &= -\rho c^2 \operatorname{div}\mathbf{U} && \text{in } \Omega, && (2.2)\\ P &= \left(\alpha\mathbf{U}\cdot\mathbf{n} + \beta\frac{\partial\mathbf{U}}{\partial t}\cdot\mathbf{n}\right) && \text{on } \Gamma_A, && (2.3)\\ \mathbf{U}\cdot\mathbf{n} &= 0 && \text{on } \Gamma_R. && (2.4) \end{aligned}$$

Here ρ is the fluid density, c, the acoustic speed, and $\mathbf{n}$, the unit outer normal vector along $\partial\Omega$. At the absorbing boundary (2.3) indicates that the pressure is balanced by the effects of the viscous damping (the β term) and the elastic behavior (the α term). We assume the coefficients α and β are given positive constants.

To look for the damped vibration modes we assume (2.1)–(2.4) has complex solution of the form $\mathbf{U}(\mathbf{x},t) = e^{\lambda t}\mathbf{u}(\mathbf{x})$ and $P(\mathbf{x},t) = e^{\lambda t}p(\mathbf{x})$. This leads to a problem of finding $\lambda \in \mathbb{C}, \mathbf{u} : \Omega \to \mathbb{C}^n$ and $p : \Omega \to \mathbb{C}, (\mathbf{u}, p) \neq (\mathbf{0}, 0)$ such that

$$\begin{aligned} \rho\lambda^2\mathbf{u} &+ \nabla p = \mathbf{0} && \text{in } \Omega, && (2.5)\\ p &= -\rho c^2 \operatorname{div}\mathbf{u} && \text{in } \Omega, && (2.6)\\ p &= (\alpha + \lambda\beta)\mathbf{u}\cdot\mathbf{n} && \text{on } \Gamma_A, && (2.7)\\ \mathbf{u}\cdot\mathbf{n} &= 0 && \text{on } \Gamma_R. && (2.8) \end{aligned}$$

The boundary condition (2.7) makes this eigenvalue problem nonlinear. For each damped vibration mode, $\omega := \mathrm{Im}(\lambda)$ is the vibration angular frequency and $\mathrm{Re}(\lambda)$ the decay rate. In practice, we select a range of ω values and are interested in the least decaying modes in this range. We next describe the natural variational formulation of the above problem on which the numerical approximation will be based.

Let

$$\mathcal{V} := \{\mathbf{v} \in H(\mathrm{div}, \Omega) : \mathbf{v} \cdot \mathbf{n} \in L^2(\partial\Omega) \text{ and } \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \Gamma_R\}.$$

Here we employ standard Sobolev spaces notation. For example, $H(\mathrm{div}, \Omega)$ stands for the space of all L^2 vector functions $\mathbf{v}$ on Ω with L^2 integrable divergence.

Testing (2.5) by $\bar{\mathbf{v}} \in \mathcal{V}$ and integrating by parts, we obtain a variational formulation of problem (2.5)–(2.8) involving only the displacement variable: Find $\lambda \in \mathbb{C}$ and $\mathbf{u} \in \mathcal{V}, \mathbf{u} \neq \mathbf{0}$, such that

$$\lambda^2 \int_\Omega \rho \mathbf{u} \cdot \bar{\mathbf{v}} + \lambda \int_{\Gamma_A} \beta \mathbf{u} \cdot \mathbf{n} \bar{\mathbf{v}} \cdot \mathbf{n} + \int_{\Gamma_A} \alpha \mathbf{u} \cdot \mathbf{n} \bar{\mathbf{v}} \cdot \mathbf{n} + \int_\Omega \rho c^2 \,\mathrm{div} \mathbf{u} \,\mathrm{div} \bar{\mathbf{v}} = 0 \quad \forall\, \mathbf{v} \in \mathcal{V}. \tag{2.9}$$

This is a quadratic eigenvalue problem. Note that $\lambda = 0$ is an eigenvalue and the dimension of its eigenspace

$$\mathcal{N} := \{\mathbf{u} \in \mathcal{V} : \ \mathrm{div}\, \mathbf{u} = 0 \text{ in } \Omega \text{ and } \mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega\}$$

is infinity. All nonzero eigenvalues have finite multiplicity (the dimension of the eigenspace is finite) [6]. It is shown in [6] that all the other solutions of (2.9), the decay rate is strictly negative. That is, if an eigenpair $0 \neq \lambda \in \mathbb{C}$ and $\mathbf{0} \neq \mathbf{u} \in \mathcal{V}$ is a solution of problem (2.9) then $\mathrm{Re}(\lambda) < 0$.

Alternatively we can derive a variational formulation in terms of the pressure: Find $\lambda \in \mathbb{C}$ and $p \in H^1(\Omega) := \{p \in L^2(\Omega) : \nabla p \in L^2(\Omega)\}$ such that

$$\frac{\lambda^2}{c^2} \int_\Omega p\bar{q} + \frac{\lambda^2}{\alpha + \lambda\beta} \int_{\Gamma_A} \rho p \bar{q} + \int_\Omega \nabla p \cdot \nabla \bar{q} = 0 \qquad \forall\, q \in H^1(\Omega). \tag{2.10}$$

However, in this case the eigenvalue problem is rational, which is rarely studied compared with linear and quadratic eigenvalue problems. Note that in contrast to the displacement formulation, the eigenspace corresponding to $\lambda = 0$ is now one dimensional. Thus this formulation has a much smaller null space or kernel, which may be more stable and efficient when used in conjunction with projection-like spectral approximation methods.

2.2.1 Spectral approximation

We now turn to the finite element methods for approximating the solutions of the quadratic eigenvalue problem (2.9) and the rational eigenvalue problem (2.10). Spurious modes are usually present when standard finite elements are used in a displacement formulation. However Bermúdez *et. al.* [6] successfully demonstrated that the spurious modes can be avoided by using the lowest order Raviart-Thomas elements in $\mathbb{R}^d, d = 2, 3$ (see, for instance, [10, 50]). For simplicity we will consider only the two dimensional case. Let $\{\mathcal{T}_h\}$ be a regular family of triangulations of Ω indexed by h, the maximum diameter of the elements. Let

$$\mathcal{V}_h := \{\mathbf{v}_h \in H(\mathrm{div}, \Omega) : \mathbf{v}_h|_T \in \mathcal{P}_0^d \oplus \mathcal{P}_0\mathbf{x} \quad \forall\, T \in \mathcal{T}_h \text{ and } \mathbf{v}_h \cdot \mathbf{n} = 0 \text{ on } \Gamma_R\} \subset \mathcal{V},$$

where $d = 2$ and $\mathcal{P}_k$ denotes the set of polynomials of degree at most k. Thus locally $\mathbf{v}_h$ takes the form $(a + sx, b + sy)^\top$. The discrete problem associated with (2.9) is : Find $\lambda \in \mathbb{C}$ and $\mathbf{u}_h \in \mathcal{V}_h, \mathbf{u}_h \neq \mathbf{0}$, such that

$$\lambda^2 \int_\Omega \rho \mathbf{u}_h \cdot \bar{\mathbf{v}}_h + \lambda \int_{\Gamma_A} \beta \mathbf{u}_h \cdot \mathbf{n}\, \bar{\mathbf{v}}_h \cdot \mathbf{n} + \int_{\Gamma_A} \alpha \mathbf{u}_h \cdot \mathbf{n}\, \bar{\mathbf{v}}_h \cdot \mathbf{n} + \int_\Omega \rho c^2\, \mathrm{div}\mathbf{u}_h\, \mathrm{div}\bar{\mathbf{v}}_h = 0,\ \forall\, \mathbf{v}_h \in \mathcal{V}_h. \tag{2.11}$$

Theorem 2.1. *The dimension of the zero eigenspace $\mathcal{E}_0$ associated with (2.11) equals the number of interior nodes in the triangulation.*

Proof. Setting $\mathbf{v}_h = \mathbf{u}_h$ and $\lambda = 0$ in (2.11), we see that

$$\mathrm{div}\mathbf{u}_h = 0 \ \text{ on } \Omega \qquad \text{and} \qquad \mathbf{u}_h \cdot \mathbf{n} = 0 \text{ on } \partial\Omega.$$

Since $\mathbf{u}_h = (a + sx, b + sy)^\top$ on $T \in \mathcal{T}_h$, the divergence free condition implies that $\mathbf{u}_h$ is a constant vector $(a, b)^\top$ on T. By direct computation, we see that there exists a linear polynomial ψ_T such that

$$\frac{\partial \psi_T}{\partial x} = -b \text{ and } \frac{\partial \psi_T}{\partial y} = a.$$

Let $\mathbf{n} = (n_1, n_2)^\top$ be a unit normal to an edge e of T, so $\mathbf{t} = (-n_2, n_1)^\top$ is a unit tangent vector to e. We see that

$$\mathbf{u}_h \cdot \mathbf{n} = \nabla \psi_T \cdot \mathbf{t} = \frac{\partial \psi_T}{\partial \mathbf{t}}.$$

So if an edge e is common to T_1 and T_2 then in general ψ_{T_1} and ψ_{T_2} differ by a constant only by the continuity of $\mathbf{u}_h \cdot \mathbf{n}$ across e. At an interior node N_j, we can assign a common value for all ψ_T at that node. Here T are all triangles sharing N_j as the common node. We then spread this defining process outward to all Ω using the induced values on other nodes. Consequently, Ψ is continuous piecewise linear over Ω. Let $\nabla^\perp := (-\frac{\partial}{\partial y}, \frac{\partial}{\partial x})^\top$ and define

$$\nabla^\perp S_h := \{\nabla^\perp \Psi_h : \Psi_h \text{ is continuous piecewise linear and vanishes on the boundary}\}.$$

Thus we have just shown the zero eigenspace $\mathcal{E}_0$ is contained $\nabla^\perp S_h$ and the opposite inclusion is also easily checked. Hence

$$\mathcal{E}_0 = \nabla^\perp S_h.$$

We now find the dimension of $\nabla^\perp S_h$. Let N be the number of interior nodes and let $\Psi_j, j = 1, \ldots, N$, be the nodal basis functions such that $\Psi_j(N_k) = \delta_{kj}$. The linear independence of Ψ_j's is preserved by the perp-gradient operation. In fact, suppose $\sum_{j=1}^{N} c_j \nabla^{\perp}\Psi_j = 0$. Then this implies $\sum_{j=1}^{N} c_j \Psi_j = c$ for some constant c. Hence $c_j = c$ by the condition $\Psi_j(N_k) = \delta_{kj}$. Consequently, $c(\sum_j \Psi_j - 1) = 0$. But we know $\sum_{j=1}^{N} \Psi_j \neq 1$ due to the vanishing boundary condition. Thus $c_j = c = 0$ and we conclude that the dimension of the zero eigenspace $\dim \mathcal{E}_0 = \dim \nabla^{\perp} S_h$ equals the number of interior nodes in the mesh. □

Define the conforming P_1 finite element space

$$H_h := \{p_h \in H^1(\Omega) : p_h|_T \in \mathcal{P}_1 \quad \forall\, T \in \mathcal{T}_h\}.$$

This is the subspace of $H^1(\Omega)$ consisted of continuous piecewise linears. The alternative discrete problem in terms of the approximate pressure field is: Find $\lambda \in \mathbb{C}$ and $p_h \in H_h$ such that

$$\frac{\lambda^2}{c^2}\int_{\Omega} p_h \bar{q}_h + \frac{\lambda^2}{\alpha + \lambda\beta}\int_{\Gamma_A} \rho p_h \bar{q}_h + \int_{\Omega} \nabla p_h \cdot \nabla \bar{q}_h = 0 \qquad \forall\, q_h \in H_h. \tag{2.12}$$

Letting $q_h = p_h$ and $\lambda = 0$ in (2.12) we can easily see that the dimension of the zero eigenspace in this case is one, which is the same as the original problem (2.10).

Again we see that the pressure formulation has a much smaller null space than the displacement formulation. Also the number of unknowns is much smaller. Thus the pressure formulation turns out to be a very good alternative, once in addition we show in the remaining sections that its associated eigenvalue problem can be efficiently solved. A minor remark is in order here.

Remark 2.2. *Suppose an eigenpair $(\lambda, p_h), \lambda \neq 0$ has been computed, what if, in addition, one wants to know a corresponding displacement approximation $\mathbf{u}_h$? One must not find $\mathbf{u}_h$ by solving an additional system linear equations again so as to maintain the advantage of the pressure formulation. It should be given by a simple* *formula. A naive way is to use the relation* (2.5) *to evaluate a* $\mathbf{u}_h$*, but this would be ill conceived since the computed displacement would be piecewise constant. Consequently,* $\nabla \cdot \mathbf{u}_h = 0$*, which certainly does not approximate* (2.6)*. Fortunately, a general principle for such a problem (recovery of* $\mathbf{u}_h$ *from the pressure approximation* p_h*) has been provided in [13] where one can obtain an accurate* $\mathbf{u}_h$ *in the Raviart-Thomas space by a simple evaluation formula which is a modification of the above naive formula.*

2.3 Linearization of Nonlinear Eigenvalue Problems

In this section we start to address the computational issues related to the displacement approximation (2.11) and the pressure approximation (2.12).

2.3.1 Linearization of quadratic eigenvalue problems

Suppose the total number of interior and absorbing edges is n_1. Let $\{\phi_j\}_{j=1}^{n_1}$ denote the cardinal basis of $\mathcal{V}_h$, so that on the edge e_j, ϕ_j has the unit normal flux and zero normal flux on the remaining $n_1 - 1$ edges. That is, $\int_{e_i} \phi_j \cdot \mathbf{n}_i d\varsigma = \delta_{ij}$. For $\mathbf{u}_h \in \mathcal{V}_h$, we write $\mathbf{u}_h = \sum_{j=1}^{n_1} u_j \phi_j$ and denote $\mathbf{u} = [u_1^\top, \cdots, u_{n_1}^\top]^\top$. Note that the unknown vector $\mathbf{u}$ contains normal fluxes in its components. Then, the discrete problem (2.11) can be expressed as the following QEP:

$$Q(\lambda)\mathbf{u} \equiv (\lambda^2 M_u + (\alpha + \lambda\beta)A_u + K_u)\mathbf{u} = \mathbf{0}, \tag{2.13}$$

where $M_u \equiv [M_{ij}^u]$ and $K_u \equiv [K_{ij}^u]$ are mass and stiffness matrices, respectively, and $A_u \equiv [A_{ij}^u]$ is used to describe the effect of the absorbing wall. Here

$$M_{ij}^u = \int_\Omega \rho\phi_i \cdot \bar{\phi}_j, \quad K_{ij}^u = \int_\Omega \rho c^2 \,\mathrm{div}\phi_i \,\mathrm{div}\bar{\phi}_j, \quad A_{ij}^u = \int_{\Gamma_A} \phi_i \cdot \mathbf{n}\, \bar{\phi}_j \cdot \mathbf{n}, \tag{2.14}$$

for $i, j = 1, \ldots, n_1$. For this problem, we are only interested in eigenvalues that are located in the interior of the spectrum. Suppose that the eigenvalues near σ are of interest. Accordingly, the QEP (2.13) is shifted into

$$\left(\mu^2 \widetilde{M}_u + \mu \widetilde{D}_u + \widetilde{K}_u\right) \mathbf{u} = \mathbf{0} \tag{2.15}$$

with $\mu = \lambda - \sigma$ and

$$\begin{cases} \widetilde{M}_u = M_u, \\ \widetilde{D}_u = 2\sigma M_u + \beta A_u, \\ \widetilde{K}_u = \sigma^2 M_u + (\alpha + \sigma\beta) A_u + K_u. \end{cases} \tag{2.16}$$

On the one hand, one can numerically solve (2.15) without transforming it further. Among such direct methods we mention the second-order Arnoldi (SOAR) method [3] and the Jacobi-Davidson algorithm applied to polynomial eigenvalue problems [57]. On the other hand, it is more common to transform or linearize (2.15) into a SEP [66]. In this chapter, we let

$$\mathcal{A}_u = \begin{bmatrix} \mathbf{0} & -\widetilde{M}_u \\ I_{n_1} & -\widetilde{D}_u \end{bmatrix}, \quad \mathcal{B}_u = \begin{bmatrix} I_{n_1} & \mathbf{0} \\ \mathbf{0} & \widetilde{K}_u \end{bmatrix} \tag{2.17}$$

and linearize (2.15) into the GEP

$$\mathcal{A}_u \boldsymbol{\varphi} = \frac{1}{\mu} \mathcal{B}_u \boldsymbol{\varphi} \quad \text{with} \quad \boldsymbol{\varphi} \equiv \begin{bmatrix} -\mu \widetilde{M}_u \mathbf{u} \\ \mathbf{u} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix}. \tag{2.18}$$

The matrix $\widetilde{K}_u$ in (2.17) is nonsingular owing to the fact that the shift value σ is not an eigenvalue of (2.13). Furthermore, the GEP (2.18) can then be transformed into two types of SEPs of the forms $(\mathcal{B}_u^{-1}\mathcal{A}_u)\boldsymbol{\varphi} = \mu^{-1}\boldsymbol{\varphi}$ and $(\mathcal{A}_u\mathcal{B}_u^{-1})\boldsymbol{\psi} = \mu^{-1}\boldsymbol{\psi}$, respectively, where $\boldsymbol{\psi} = \mathcal{B}_u \boldsymbol{\varphi}$. Therefore, from (2.17) and (2.18) we have

$$(\textbf{Q-SEP1}) \quad \mathcal{B}_u^{-1}\mathcal{A}_u \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\widetilde{M}_u \\ \widetilde{K}_u^{-1} & -\widetilde{K}_u^{-1}\widetilde{D}_u \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix} = \frac{1}{\mu} \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix} \tag{2.19.1}$$

and

$$(\textbf{Q-SEP2}) \quad \mathcal{A}_u\mathcal{B}_u^{-1} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\widetilde{M}_u\widetilde{K}_u^{-1} \\ I_{n_1} & -\widetilde{D}_u\widetilde{K}_u^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} = \frac{1}{\mu} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}, \mathbf{w} = \widetilde{K}_u\mathbf{u}. \tag{2.19.2}$$

Note that the SEPs of (2.19.1) and (2.19.2) derived by the QEP in (2.15), are called **Q-SEP1** and **Q-SEP2**, respectively. The standard Arnoldi method can then be applied to solve **Q-SEPs**, and the details will be given in Section 2.4.

2.3.2 Trimmed linearization for rational eigenvalue problems

Let $\{\psi_j\}_{j=1}^{n_2}$ be a nodal basis of H_h. For $p_h \in H_h$, we write $p_h = \sum\limits_{j=1}^{n_2} p_j\psi_j$ and denote $\mathbf{p} = [p_1, \cdots, p_{n_2}]^\top$. Then, the discrete problem (2.12) can be written as the following REP:

$$R(\lambda)\mathbf{p} \equiv \left(\frac{\lambda^2}{c^2} M_p + K_p + \frac{\lambda^2}{\lambda\beta + \alpha} A_p \right) \mathbf{p} = \mathbf{0}, \tag{2.20}$$

where $M_p \equiv [M_{ij}^p]$ and $K_p \equiv [K_{ij}^p]$ are mass and stiffness matrices, respectively, and $A_p \equiv [A_{ij}^p]$ describes the effect of the absorbing wall. Here,

$$M_{ij}^p = \int_\Omega \psi_i \bar{\psi}_j, \quad K_{ij}^p = \int_\Omega \nabla\psi_i \cdot \nabla\bar{\psi}_j, \quad A_{ij}^p = \int_{\Gamma_A} \rho\psi_i\bar{\psi}_j \tag{2.21}$$

for $i, j = 1, \ldots, n_2$.

To solve REP (2.20), one approach is to multiply equation (2.20) by the scalar $\lambda\beta+\alpha$ and expand it into a cubic polynomial eigenvalue problem, and then solve it by

Jacobi-Davidson method [30]. An alternative approach is to treat (2.20) as nonlinear eigenvalue problem and solve it by a nonlinear eigensolver, such as Newton's method, nonlinear Arnoldi method, or nonlinear Jacobi-Davidson method [52, 68, 69]. Recently, a trimmed linearization is proposed in [64] which linearizes (2.20) into a GEP so that the standard Arnoldi method can be applied. We introduce the trimmed linearization below.

Given a shift value σ. With $\mu = \lambda - \sigma$, the rational λ-matrix $R(\lambda)$ in (2.20) can be rewritten as

$$\begin{aligned} R(\lambda) &= \frac{(\lambda-\sigma+\sigma)^2}{c^2} M_p + K_p + \frac{(\lambda-\sigma+\sigma)^2}{(\lambda-\sigma+\sigma)\beta+\alpha} A_p \\ &= \frac{(\lambda-\sigma)^2+2(\lambda-\sigma)\sigma+\sigma^2}{c^2} M_p + K_p + \frac{(\lambda-\sigma)^2+2(\lambda-\sigma)\sigma+\sigma^2}{(\lambda-\sigma)\beta+\sigma\beta+\alpha} A_p \\ &= \mu^2\left(\frac{1}{c^2} M_p\right) + \mu\left(\frac{2\sigma}{c^2} M_p\right) + \left(\frac{\sigma^2}{c^2} M_p + K_p\right) + \frac{\mu^2+2\mu\sigma+\sigma^2}{\mu\beta+\sigma\beta+\alpha} A_p. \end{aligned} \tag{2.22}$$

By applying the long division, the rational term in (2.22) can be simplified into the following

$$\begin{aligned} \frac{\mu^2+2\mu\sigma+\sigma^2}{\mu\beta+\sigma\beta+\alpha} &= \mu^2\left[\frac{\alpha^2}{(\sigma\beta+\alpha)^3}\right] + \mu\left[\frac{\sigma^2\beta+2\sigma\alpha}{(\sigma\beta+\alpha)^2}\right] \\ &\quad + \frac{\sigma^2}{\sigma\beta+\alpha} - \mu^2\left[\frac{(\sigma\beta+\alpha)^3}{\alpha^2} + \frac{(\sigma\beta+\alpha)^4}{\alpha^2\beta\mu}\right]^{-1}. \end{aligned}$$

This implies that

$$\begin{aligned} R(\lambda) &= \mu^2\left(\frac{1}{c^2} M_p + \frac{\alpha^2}{(\sigma\beta+\alpha)^3} A_p\right) + \mu\left(\frac{2\sigma}{c^2} M_p + \frac{\sigma^2\beta+2\sigma\alpha}{(\sigma\beta+\alpha)^2} A_p\right) \\ &\quad + \left(\frac{\sigma^2}{c^2} M_p + K_p + \frac{\sigma^2}{\sigma\beta+\alpha} A_p\right) - \mu^2\left(\frac{(\sigma\beta+\alpha)^3}{\alpha^2} + \frac{(\sigma\beta+\alpha)^4}{\alpha^2\beta\mu}\right)^{-1} A_p \\ &= \mu^2 \widetilde{M}_p + \mu \widetilde{D}_p + \widetilde{K}_p - \mu^2\left(\vartheta - \varrho\mu^{-1}\right)^{-1} L_p R_p^\top, \end{aligned} \tag{2.23}$$

where

$$\widetilde{M}_p = \frac{1}{c^2}M_p + \frac{\alpha^2}{(\sigma\beta+\alpha)^3}A_p, \tag{2.24}$$

$$\widetilde{D}_p = \frac{2\sigma}{c^2}M_p + \frac{\sigma^2\beta+2\sigma\alpha}{(\sigma\beta+\alpha)^2}A_p, \tag{2.25}$$

$$\widetilde{K}_p = \frac{\sigma^2}{c^2}M_p + K_p + \frac{\sigma^2}{\sigma\beta+\alpha}A_p, \tag{2.26}$$

$$\vartheta = \frac{(\sigma\beta+\alpha)^3}{\alpha^2}, \quad \varrho = -\frac{(\sigma\beta+\alpha)^4}{\alpha^2\beta}, \tag{2.27}$$

and $L_pR_p^\top = A_p$ is the full-rank decomposition of A_p with $L_p, R_p \in \mathbb{R}^{n_2\times\ell}$, $\ell \ll n_2$. Introducing an auxiliary vector

$$\mathbf{q} = \frac{\mu}{\vartheta\mu-\varrho}R_p^\top\mathbf{p}, \tag{2.28}$$

the REP in (2.20) can be reformulated as

$$\left(\mu^2\widetilde{M}_p + \mu\widetilde{D}_p + \widetilde{K}_p\right)\mathbf{p} - \mu^2L_p\mathbf{q} = \mathbf{0}. \tag{2.29}$$

Using (2.28) and (2.29), we get the GEP

$$\mathcal{A}_p\boldsymbol{\varphi} \equiv \begin{bmatrix} \mathbf{0} & -\widetilde{M}_p & L_p \\ I_{n_2} & -\widetilde{D}_p & \mathbf{0} \\ \mathbf{0} & -R_p^\top & \vartheta I_\ell \end{bmatrix}\boldsymbol{\varphi} = \frac{1}{\mu}\begin{bmatrix} I_{n_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \widetilde{K}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \varrho I_\ell \end{bmatrix}\boldsymbol{\varphi} \equiv \frac{1}{\mu}\mathcal{B}_p\boldsymbol{\varphi}, \tag{2.30}$$

where $\boldsymbol{\varphi} = [((\mu^{-1}\widetilde{K}_p + \widetilde{D}_p)\mathbf{p})^\top, \mathbf{p}^\top, \mathbf{q}^\top]^\top$. As before, the matrix $\widetilde{K}_p$ in (2.26) is nonsingular due to the fact that the shift value σ is not an eigenvalue of (2.20). As in (2.19.1) and (2.19.2), the GEP (2.30) can then be, respectively, transformed into the following two types of the SEPs of the forms $(\mathcal{B}_p^{-1}\mathcal{A}_p)\boldsymbol{\varphi} = \mu^{-1}\boldsymbol{\varphi}$ and $(\mathcal{A}_p\mathcal{B}_p^{-1})\boldsymbol{\psi} =$

$\mu^{-1}\boldsymbol{\psi}$ where $\boldsymbol{\psi} = \mathcal{B}_p\boldsymbol{\varphi}$. Consequently, we have

$$(\textbf{R-SEP1}) \quad \mathcal{B}_p^{-1}\mathcal{A}_p\boldsymbol{\varphi} = \begin{bmatrix} \mathbf{0} & -\widetilde{M}_p & L_p \\ \widetilde{K}_p^{-1} & -\widetilde{K}_p^{-1}\widetilde{D}_p & \mathbf{0} \\ \mathbf{0} & -\varrho^{-1}R_p^{\top} & \varrho^{-1}\vartheta I_\ell \end{bmatrix}\boldsymbol{\varphi} = \frac{1}{\mu}\boldsymbol{\varphi}, \tag{2.31.1}$$

and

$$(\textbf{R-SEP2}) \quad \mathcal{A}_p\mathcal{B}_p^{-1}\boldsymbol{\psi} = \begin{bmatrix} \mathbf{0} & -\widetilde{M}_p\widetilde{K}_p^{-1} & \varrho^{-1}L_p \\ I_{n_2} & -\widetilde{D}_p\widetilde{K}_p^{-1} & \mathbf{0} \\ \mathbf{0} & -R_p^{\top}\widetilde{K}_p^{-1} & \varrho^{-1}\vartheta I_\ell \end{bmatrix}\boldsymbol{\psi} = \frac{1}{\mu}\boldsymbol{\psi}, \quad \boldsymbol{\psi} = \mathcal{B}_p\boldsymbol{\varphi}. \tag{2.31.2}$$

Note that the SEPs of (2.31.1) and (2.31.2) derived by the REP in (2.29) are called **R-SEP1** and **R-SEP2**, respectively.

2.3.3 Error analysis

In this subsection, we will discuss residuals of QEP (2.13) and REP (2.20) by using linearizations (2.19) and (2.31), respectively.

We first derive residual bounds of approximate eigenpairs for QEP (2.13) by by using linearizations **Q-SEP1** and **Q-SEP2**, respectively. Let $(\mu_1^{-1}, \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{u}_1 \end{bmatrix})$ be an approximate eigenpair of (2.19.1) and $\begin{bmatrix} \mathbf{f}_{11} \\ \mathbf{f}_{12} \end{bmatrix}$ be the associated residual vector. That is,

$$\begin{aligned} \begin{bmatrix} \mathbf{f}_{11} \\ \mathbf{f}_{12} \end{bmatrix} &= \begin{bmatrix} \mathbf{0} & -\widetilde{M}_u \\ \widetilde{K}_u^{-1} & -\widetilde{K}_u^{-1}\widetilde{D}_u \end{bmatrix}\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{u}_1 \end{bmatrix} - \frac{1}{\mu_1}\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{u}_1 \end{bmatrix} \\ &= \frac{1}{\mu_1}\begin{bmatrix} -\mathbf{v}_1 - \mu_1\widetilde{M}_u\mathbf{u}_1 \\ \widetilde{K}_u^{-1}(\mu_1\mathbf{v}_1 - \mu_1\widetilde{D}_u\mathbf{u}_1 - \widetilde{K}_u\mathbf{u}_1) \end{bmatrix}. \end{aligned}$$

It follows that

$$\begin{aligned} \mu_1^2\widetilde{M}_u\mathbf{u}_1 + \mu_1\widetilde{D}_u\mathbf{u}_1 + \widetilde{K}_u\mathbf{u}_1 &= \mu_1(-\mathbf{v}_1 - \mu_1\mathbf{f}_{11}) + \mu_1\mathbf{v}_1 - \mu_1\widetilde{K}_u\mathbf{f}_{12} \\ &= -\mu_1^2\mathbf{f}_{11} - \mu_1\widetilde{K}_u\mathbf{f}_{12}. \end{aligned}$$

Let $\lambda_1 = \mu_1 + \sigma$. From (2.13) we have

$$\frac{\|Q(\lambda_1)\mathbf{u}_1\|}{\|\mathbf{u}_1\|} = \frac{\|\mu_1^2\widetilde{M}_u\mathbf{u}_1 + \mu_1\widetilde{D}_u\mathbf{u}_1 + \widetilde{K}_u\mathbf{u}_1\|}{\|\mathbf{u}_1\|} \leq \frac{|\mu_1|^2\|\mathbf{f}_{11}\| + |\mu_1|\|\widetilde{K}_u\|\|\mathbf{f}_{12}\|}{\|\mathbf{u}_1\|}. \quad (2.32)$$

On the other hand, let $(\mu_2^{-1}, \begin{bmatrix}\mathbf{v}_2\\ \mathbf{w}_2\end{bmatrix})$ be an approximate eigenpair of (2.19.2) and $\begin{bmatrix}\mathbf{f}_{21}\\ \mathbf{f}_{22}\end{bmatrix}$ be the associated residual vector. That is,

$$\begin{aligned} \begin{bmatrix}\mathbf{f}_{21}\\ \mathbf{f}_{22}\end{bmatrix} &= \begin{bmatrix}\mathbf{0} & -\widetilde{M}_u\widetilde{K}_u^{-1}\\ I & -\widetilde{D}_u\widetilde{K}_u^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{v}_2\\ \mathbf{w}_2\end{bmatrix} - \frac{1}{\mu_2}\begin{bmatrix}\mathbf{v}_2\\ \mathbf{w}_2\end{bmatrix} \\ &= \begin{bmatrix}-\widetilde{M}_u\widetilde{K}_u^{-1}\mathbf{w}_2 - \frac{1}{\mu_2}\mathbf{v}_2\\ \mathbf{v}_2 - \widetilde{D}_u\widetilde{K}_u^{-1}\mathbf{w}_2 - \frac{1}{\mu_2}\mathbf{w}_2\end{bmatrix}. \end{aligned}$$

It follows that

$$\begin{aligned} \mu_2^2\widetilde{M}_u\widetilde{K}_u^{-1}\mathbf{w}_2 + \mu_2\widetilde{D}_u\widetilde{K}_u^{-1}\mathbf{w}_2 + \mathbf{w}_2 &= \mu_2(-\mathbf{v}_2 - \mu_2\mathbf{f}_{21}) + \mu_2\mathbf{v}_2 - \mu_2\mathbf{f}_{22} \\ &= -\mu_2^2\mathbf{f}_{21} - \mu_2\mathbf{f}_{22}. \end{aligned}$$

Letting $\mathbf{u}_2 = \widetilde{K}_u^{-1}\mathbf{w}_2$ and $\lambda_2 = \mu_2 + \sigma$. From (2.13) we have,

$$\frac{\|Q(\lambda_2)\mathbf{u}_2\|}{\|\mathbf{u}_2\|} = \frac{\|\mu_2^2\widetilde{M}_u\mathbf{u}_2 + \mu_2\widetilde{D}_u\mathbf{u}_2 + \widetilde{K}_u\mathbf{u}_2\|}{\|\mathbf{u}_2\|} \leq \frac{|\mu_2|^2\|\mathbf{f}_{21}\| + |\mu_2|\|\mathbf{f}_{22}\|}{\|\mathbf{u}_2\|}. \quad (2.33)$$

Now, we derive residual bounds of approximate eigenpairs for REP (2.20) by using linearizations **R-SEP1** and **R-SEP2**, respectively. Let $(\mu_1^{-1}, [\mathbf{s}_1^\top, \mathbf{p}_1^\top, \mathbf{q}_1^\top]^\top)$ be an approximate eigenpair of (2.31.1) and $[\mathbf{g}_{11}^\top, \mathbf{g}_{12}^\top, \mathbf{g}_{13}^\top]^\top$ be the associated residual vector. That is,

$$\begin{bmatrix} \mathbf{g}_{11} \\ \mathbf{g}_{12} \\ \mathbf{g}_{13} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\widetilde{M}_p & L_p \\ \widetilde{K}_p^{-1} & -\widetilde{K}_p^{-1}\widetilde{D}_p & \mathbf{0} \\ \mathbf{0} & -\varrho^{-1}R_p^\top & \varrho^{-1}\vartheta I_\ell \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{p}_1 \\ \mathbf{q}_1 \end{bmatrix} - \frac{1}{\mu_1} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{p}_1 \\ \mathbf{q}_1 \end{bmatrix}.$$

This implies that

$$\mathbf{s}_1 = -\mu_1\widetilde{M}_p\mathbf{p}_1 + \mu_1 L_p\mathbf{q}_1 - \mu_1\mathbf{g}_{11}, \tag{2.34}$$

$$\mathbf{g}_{12} = \widetilde{K}_p^{-1}\mathbf{s}_1 - \widetilde{K}_p^{-1}\widetilde{D}_p\mathbf{p}_1 - \frac{1}{\mu_1}\mathbf{p}_1, \tag{2.35}$$

$$\mathbf{q}_1 = \left(\mu_1\varrho^{-1}\vartheta - 1\right)^{-1}\mu_1\left(\mathbf{g}_{13} + \varrho^{-1}R_p^\top\mathbf{p}_1\right). \tag{2.36}$$

Substituting (2.36) into (2.34), $\mathbf{s}_1$ can be represented by

$$\mathbf{s}_1 = -\mu_1\widetilde{M}_p\mathbf{p}_1 + \mu_1^2\left(\mu_1\varrho^{-1}\vartheta - 1\right)^{-1}\left(L_p\mathbf{g}_{13} + \varrho^{-1}L_pR_p^\top\mathbf{p}_1\right) - \mu_1\mathbf{g}_{11}. \tag{2.37}$$

Substituting (2.37) into (2.35) and taking $\lambda_1 = \mu_1 + \sigma$. From (2.23) and (2.27),

$$\begin{aligned} R(\lambda_1)\mathbf{p}_1 &= \mu_1^2\widetilde{M}_p\mathbf{p}_1 + \mu_1\widetilde{D}_p\mathbf{p}_1 + \widetilde{K}_p\mathbf{p}_1 - \mu_1^2\left[\frac{(\sigma\beta+\alpha)^3}{\alpha^2} + \frac{(\sigma\beta+\alpha)^4}{\alpha^2\beta\mu_1}\right]^{-1} L_pR_p^\top\mathbf{p}_1 \\ &= -\mu_1^2\mathbf{g}_{11} - \mu_1\widetilde{K}_p\mathbf{g}_{12} - \mu_1^2\left(\frac{\beta}{\sigma\beta+\alpha} + \frac{1}{\mu_1}\right)^{-1} L_p\mathbf{g}_{13} \end{aligned}$$

which implies that

$$\begin{aligned} \frac{\|R(\lambda_1)\mathbf{p}_1\|}{\|\mathbf{p}_1\|} \leq \frac{1}{\|\mathbf{p}_1\|}\Bigg\{&|\mu_1|^2\|\mathbf{g}_{11}\| + |\mu_1|\|\widetilde{K}_p\|\;\|\mathbf{g}_{12}\| \\ &+ \left|\mu_1^2\left(\frac{\beta}{\sigma\beta+\alpha} + \frac{1}{\mu_1}\right)^{-1}\right|\|L_p\|\;\|\mathbf{g}_{13}\|\Bigg\}. \end{aligned} \tag{2.38}$$

On the other hand, let $(\mu_2^{-1}, [\mathbf{s}_2^\top, \mathbf{t}_2^\top, \mathbf{q}_2^\top]^\top)$ be an approximate eigenpair of (2.31.2) and $[\mathbf{g}_{21}^\top, \mathbf{g}_{22}^\top, \mathbf{g}_{23}^\top]^\top$ be the associated residual vector. That is,

$$\begin{bmatrix} \mathbf{g}_{21} \\ \mathbf{g}_{22} \\ \mathbf{g}_{23} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\widetilde{M}_p \widetilde{K}_p^{-1} & \varrho^{-1} L_p \\ I_{n_2} & -\widetilde{D}_p \widetilde{K}_p^{-1} & \mathbf{0} \\ \mathbf{0} & -R_p^\top \widetilde{K}_p^{-1} & \varrho^{-1}\vartheta I_\ell \end{bmatrix} \begin{bmatrix} \mathbf{s}_2 \\ \mathbf{t}_2 \\ \mathbf{q}_2 \end{bmatrix} - \frac{1}{\mu_2} \begin{bmatrix} \mathbf{s}_2 \\ \mathbf{t}_2 \\ \mathbf{q}_2 \end{bmatrix}.$$

This implies that

$$\mathbf{g}_{21} = -\widetilde{M}_p \widetilde{K}_p^{-1} \mathbf{t}_2 + \varrho^{-1} L_p \mathbf{q}_2 - \frac{1}{\mu_2} \mathbf{s}_2, \tag{2.39}$$

$$\mathbf{s}_2 = \widetilde{D}_p \widetilde{K}_p^{-1} \mathbf{t}_2 + \frac{1}{\mu_2} \mathbf{t}_2 + \mathbf{g}_{22}, \tag{2.40}$$

$$\mathbf{q}_2 = \left(\varrho^{-1}\vartheta - \frac{1}{\mu_2} \right)^{-1} \left(R_p^\top \widetilde{K}_p^{-1} \mathbf{t}_2 + \mathbf{g}_{23} \right). \tag{2.41}$$

Substituting (2.40) and (2.41) into (2.39), we have

$$\begin{aligned} & \mu_2^2 \widetilde{M}_p \widetilde{K}_p^{-1} \mathbf{t}_2 + \mu_2 \widetilde{D}_p \widetilde{K}_p^{-1} \mathbf{t}_2 + \mathbf{t}_2 - \mu_2^2 \left(\vartheta - \varrho \mu_2^{-1} \right)^{-1} L_p R_p^\top \widetilde{K}_p^{-1} \mathbf{t}_2 \\ = \; & -\mu_2^2 \mathbf{g}_{21} - \mu_2 \mathbf{g}_{22} + \mu_2^2 \left(\vartheta - \varrho \mu_2^{-1} \right)^{-1} L_p \mathbf{g}_{23}. \end{aligned}$$

Letting $\mathbf{p}_2 = \widetilde{K}_p^{-1} \mathbf{t}_2$ and setting $\lambda_2 = \mu_2 + \sigma$. From (2.23) we get

$$\begin{aligned} R(\lambda_2)\mathbf{p}_2 &= \mu_2^2 \widetilde{M}_p \mathbf{p}_2 + \mu_2 \widetilde{D}_p \mathbf{p}_2 + \widetilde{K}_p \mathbf{p}_2 \\ &\qquad - \mu_2^2 \left[\frac{(\sigma\beta + \alpha)^3}{\alpha^2} + \frac{(\sigma\beta + \alpha)^4}{\alpha^2 \beta \mu_2} \right]^{-1} L_p R_p^\top \mathbf{p}_2 \\ &= -\mu_2^2 \mathbf{g}_{21} - \mu_2 \mathbf{g}_{22} + \mu_2^2 \frac{\alpha^2 \beta}{(\sigma\beta + \alpha)^4} \left(\frac{\beta}{\sigma\beta + \alpha} + \frac{1}{\mu_2} \right)^{-1} L_p \mathbf{g}_{23}. \end{aligned}$$

Hence,

$$\begin{aligned}\frac{\|R(\lambda_2)\mathbf{p}_2\|}{\|\mathbf{p}_2\|} \;\le\; & \frac{1}{\|\mathbf{p}_2\|}\Bigg\{|\mu_2|^2\|\mathbf{g}_{21}\| + |\mu_2|\|\mathbf{g}_{22}\| \\ & + \left|\mu_2^2\frac{\alpha^2\beta}{(\sigma\beta+\alpha)^4}\left(\frac{\beta}{\sigma\beta+\alpha}+\frac{1}{\mu_2}\right)^{-1}\right|\|L_p\|\|\mathbf{g}_{23}\|\Bigg\}.\end{aligned} \tag{2.42}$$

Remark 2.3. *In order to check the tightness of upper bounds in* (2.32) *and* (2.33)*, as well as,* (2.38) *and* (2.42) *for residuals, respectively, we refer to the coefficient matrices generated in Example 2.1 of Section 2.5. For* (2.9) *we adopt the data as in [6] by setting* $\rho = 1\mathtt{kg}/\mathtt{m}^3$, $c = 340\ \mathtt{m}/\mathtt{s}$, $\alpha = 5\times10^4\ \mathtt{N}/\mathtt{m}^3$, *and* $\beta = 200\ \mathtt{Ns}/\mathtt{m}^3$. *In addition, we choose* $\sigma = -25 + 600\pi\mathtt{i}$ *as the shift value. Then*

(i) from (2.14)*, the element mass and stiffness matrices are*

$$\frac{h^2}{6}\rho\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad \textit{and} \quad \rho c^2\begin{bmatrix} 2 & 2 & 2\sqrt{2} \\ 2 & 2 & 2\sqrt{2} \\ 2\sqrt{2} & 2\sqrt{2} & 4 \end{bmatrix},$$

respectively. Hence, by (2.16) *the infinity norm of* $\widetilde{K}_u$ *can be estimated by* $\|\widetilde{K}_u\|_\infty \approx \|K_u\|_\infty = \mathcal{O}(\rho c^2) = \mathcal{O}(10^5)$. *From* (2.32) *and* (2.33)*, we conclude that the upper bound for the residual of the approximate eigenpair* $(\mu_1+\sigma, \mathbf{u}_1)$ *of* (2.13) *by solving* ***Q-SEP1*** *is larger than that of the approximate eigenpair* $(\mu_2+\sigma, \mathbf{u}_2)$ *of* (2.13) *by solving* ***Q-SEP2****.*

(ii) From (2.21)*, the element mass and stiffness matrices are*

$$\frac{h^2}{24}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad \textit{and} \quad \begin{bmatrix} 1 & -1/2 & -1/2 \\ -1/2 & 1/2 & 0 \\ -1/2 & 0 & 1/2 \end{bmatrix},$$

respectively. Hence, by (2.26) *we have that* $\|\widetilde{K}_p\|_\infty \approx \|K_p\|_\infty = \mathcal{O}(1)$*. If the eigenvalue* λ *is one of the desired eigenvalues in Figure 2.2, then with* $\mu = \lambda - \sigma$ *we have*

$$4 \times 10^7 < \left| \mu^2 \left(\frac{\beta}{\sigma\beta + \alpha} + \frac{1}{\mu} \right)^{-1} \right| < 3.1 \times 10^{10}$$

and

$$0.001 < \left| \mu^2 \frac{\alpha^2 \beta}{(\sigma\beta + \alpha)^4} \left(\frac{\beta}{\sigma\beta + \alpha} + \frac{1}{\mu} \right)^{-1} \right| < 0.8.$$

Clearly, from (2.38) *and* (2.42) *we conclude that the upper bound for the residual of the approximate eigenpair* $(\mu_1 + \sigma, \mathbf{p}_1)$ *of REP* (2.20) *by solving* ***R-SEP1*** *is larger than that of* $(\mu_2 + \sigma, \mathbf{p}_2)$ *of* (2.20) *by solving* ***R-SEP2****.*

2.4 Arnoldi Method with Schur-restarting

The Arnoldi method is the most popular method for solving large sparse SEPs: $A\mathbf{x} = \lambda\mathbf{x}$. In Arnoldi process, an orthonormal matrix V_{m+1} is generated to satisfy

$$AV_m = V_m H_m + h_{m+1,m} \mathbf{v}_{m+1} \mathbf{e}_m^\top, \tag{2.43}$$

where $H_m \in \mathbb{C}^{m \times m}$ is an upper Hessenberg matrix. If the dimension of the Krylov subspace span$\{V_m\}$ is larger than a certain value, then the process of Arnoldi decomposition will be restarted.

For the restarting process, we can use an implicit restart scheme [46, 59]. The package ARPACK [39] includes a very successful implementation of the implicitly restarted Arnoldi algorithm. It has been used by numerous engineering fields and remains a popular choice for solving eigenvalue problems. However, these implicitly restart type schemes may suffer from numerical instability due to rounding errors. Stewart proposed the Krylov-Schur method [28, 61, 63] that relaxes the need to preserve the structure of the Arnoldi decomposition and therefore ease the complications of the purging and deflating.

We state the Schur-restarting scheme as follows. Let

$$H_m = [U_k \;\; U_\ell] \begin{bmatrix} T_k & T_f \\ \mathbf{0} & T_\ell \end{bmatrix} \begin{bmatrix} U_k^H \\ U_\ell^H \end{bmatrix} \tag{2.44}$$

be a Schur decomposition of H_m where T_k and T_ℓ are upper triangular, and the eigenvalues of T_k are of interest. Substituting (2.44) into (2.43), we see that

$$A(V_m\,[U_k \;\; U_\ell]) = (V_m\,[U_k \;\; U_\ell]) \begin{bmatrix} T_k & T_f \\ \mathbf{0} & T_\ell \end{bmatrix} + h_{m+1,m}\mathbf{v}_{m+1}(\mathbf{e}_m^\top\,[U_k \;\; U_\ell]),$$

which implies that

$$A\tilde{V}_k = \tilde{V}_k T_k + \tilde{\mathbf{v}}_{k+1}\mathbf{t}_k^H, \tag{2.45}$$

where $\tilde{V}_k \equiv V_m U_k$, $\tilde{\mathbf{v}}_{k+1} = \mathbf{v}_{m+1}$ and $\mathbf{t}_k^H \equiv h_{m+1,m}\mathbf{e}_m^\top U_k$.

Let Q_1 be a Householder matrix with $\mathbf{t}_k^H Q_1 = \tau \mathbf{e}_k^\top$. Then (2.45) can be rewritten as

$$A(\tilde{V}_k Q_1) = (\tilde{V}_k Q_1)(Q_1^H T_k Q_1) + \tau \tilde{\mathbf{v}}_{k+1}\mathbf{e}_k^\top. \tag{2.46}$$

The matrix $Q_1^H T_k Q_1$ can be reduced to a new Hessenberg matrix H_k^+ by using Householder matrices Q_i for $i = 2, \ldots, k-1$ with

$$\begin{aligned} Q_{k-1}^H \cdots Q_2^H (Q_1^H T_k Q_1) Q_2 \cdots Q_{k-1} &= H_k^+ \\ \mathbf{e}_k^\top Q_2 \cdots Q_{k-1} &= \mathbf{e}_k^\top. \end{aligned}$$

Multiplying (2.46) by Q_i, $i = 2, \ldots, k-1$, a new Arnoldi decomposition of order k

$$AV_k^+ = V_k^+ H_k^+ + \tau \mathbf{v}_{k+1}^+ \mathbf{e}_k^\top$$

is obtained where $V_k^+ := \tilde{V}_k Q_1 \cdots Q_{k-1}$, $\mathbf{v}_{k+1}^+ = \tilde{\mathbf{v}}_{k+1} = \mathbf{v}_{m+1}$ and the Arnoldi process can be applied to generate it to order m in (2.43). One repeats the above process until the desired eigenvalues are convergent. The process is summarized in Algorithm 2.1.

Algorithm 2.1 Arnoldi method with Schur-restarting for solving $A\mathbf{x} = \lambda\mathbf{x}$

```
Input: $A$: coefficient matrix, tol$_A$: tolerance for convergence, $r_{\max}$: maximum number of Schur-restartings.
Output: The desired $k$ eigenpairs.
Build an initial Arnoldi decomposition of order $k$ as in (2.43) and set $r = 0$.
restart
   Extend Arnoldi decomposition of order $k$ to order $m = k + \ell$ and set $r = r + 1$.
   Compute all Ritz pairs $(\mu_i^{-1}, \mathbf{z}_i)$ with $H_k \mathbf{z}_i = \mu_i^{-1} \mathbf{z}_i$, $i = 1, \ldots, m$ and sorting Ritz values so that $\{(\mu_1, z_1), \ldots, (\mu_k, z_k)\}$ are wanted.
   for $i = 1, \ldots, k$ do
      Check convergence by $|h_{m+1,m}||\mathbf{e}_m^\top \mathbf{z}_i| < \mathrm{tol}_A$.
   end for
   if ( Not all $m$ desired eigenvalues are convergent and $r < r_{\max}$ ) then
      Compute the Schur decomposition of $H_m$ as in (2.44), where the eigenvalues of $T_k$ are of interest.
      Set $V_k := V_m U_k$, $\mathbf{v}_{k+1} := \mathbf{v}_{m+1}$ and $\mathbf{t}_k^H := h_{m+1,m} \mathbf{e}_m^\top U_k$.
      Compute Householder transformation $Q_1$ such that $\mathbf{t}_k^H Q_1 = \tau \mathbf{e}_k^\top$.
      Reduce $Q_1^H T_k Q_1$ to a new Hessenberg matrix $H_k$ by using Householder transformations $Q_i$ for $i = 2, \ldots, k-1$.
      Set $V_k := V_k Q_1 \cdots Q_{k-1}$ and $h_{k+1,k} = \tau$ to get the new Arnoldi decomposition with order $k$:
         $AV_k = V_k H_k + h_{k+1,k} \mathbf{v}_{k+1} \mathbf{e}_k^\top$.   (2.47)
   end if
until ( desired $k$ eigenpairs are convergent or $r \geq r_{\max}$ )
```

Now, we will apply the Algorithm 2.1 to solve QEP (2.13) and REP (2.20), respectively, by setting A to be the coefficient matrices in (2.19) and (2.31), respectively.

2.4.1 Stopping criteria

Let $(\mu^{-1}, \mathbf{z})$ be a Ritz pair and satisfy $H_m\mathbf{z} = \mu^{-1}\mathbf{z}$. From (2.43) and **Q-SEP1** in (2.19.1) we have

$$\begin{bmatrix} \mathbf{0} & -\widetilde{M}_u \\ \widetilde{K}_u^{-1} & -\widetilde{K}_u^{-1}\widetilde{D}_u \end{bmatrix} \begin{bmatrix} V_{m1} \\ V_{m2} \end{bmatrix} \mathbf{z} = \frac{1}{\mu} \begin{bmatrix} V_{m1} \\ V_{m2} \end{bmatrix} \mathbf{z} + h_{m+1,m} \begin{bmatrix} \mathbf{v}_{m+1,1} \\ \mathbf{v}_{m+1,2} \end{bmatrix} \mathbf{e}_m^\top \mathbf{z}, \quad (2.48)$$

where $V_m = \left[\begin{smallmatrix} V_{m1} \\ V_{m2} \end{smallmatrix}\right]$ and $\mathbf{v}_{m+1} = \left[\begin{smallmatrix} \mathbf{v}_{m+1,1} \\ \mathbf{v}_{m+1,2} \end{smallmatrix}\right]$ are partitioned with compatible sizes. Using the first equation of (2.48), we can eliminate $V_{m1}\mathbf{z}$ in the second equation and get

$$\frac{\|Q(\lambda)\mathbf{u}_1\|}{\|\mathbf{u}_1\|} = \frac{\|(\mu^2\widetilde{M}_u + \mu\widetilde{D}_u + \widetilde{K}_u)\mathbf{u}_1\|}{\|\mathbf{u}_1\|} = \frac{|\mu|\,|h_{m+1,m}|\,|\mathbf{e}_m^\top\mathbf{z}|\,\zeta_1}{\|\mathbf{u}_1\|} \equiv q_1(\mu), \quad (2.49)$$

where $\mathbf{u}_1 = V_{m2}\mathbf{z}$, $\lambda = \mu + \sigma$ and $\zeta_1 = \|\mu\mathbf{v}_{m+1,1} + \widetilde{K}_u\mathbf{v}_{m+1,2}\|$. Without ambiguity by using the same notations as above in Algorithm 2.1, from (2.43) and **Q-SEP2** in (2.19.2) we also have

$$\begin{bmatrix} \mathbf{0} & -\widetilde{M}_u\widetilde{K}_u^{-1} \\ I_{n_1} & -\widetilde{D}_u\widetilde{K}_u^{-1} \end{bmatrix} \begin{bmatrix} V_{m1} \\ V_{m2} \end{bmatrix} \mathbf{z} = \frac{1}{\mu} \begin{bmatrix} V_{m1} \\ V_{m2} \end{bmatrix} \mathbf{z} + h_{m+1,m} \begin{bmatrix} \mathbf{v}_{m+1,1} \\ \mathbf{v}_{m+1,2} \end{bmatrix} \mathbf{e}_m^\top \mathbf{z}$$

and

$$\frac{\|Q(\lambda)\mathbf{u}_2\|}{\|\mathbf{u}_2\|} = \frac{\|(\mu^2\widetilde{M}_u + \mu\widetilde{D}_u + \widetilde{K}_u)\mathbf{u}_2\|}{\|\mathbf{u}_2\|} = \frac{|\mu|\,|h_{m+1,m}|\,|\mathbf{e}_m^\top\mathbf{z}|\,\zeta_2}{\|\mathbf{u}_2\|} \equiv q_2(\mu), \quad (2.50)$$

where $\mathbf{u}_2 = \widetilde{K}_u^{-1}V_{m2}\mathbf{z}$, $\lambda = \mu + \sigma$ and $\zeta_2 = \|\mu\mathbf{v}_{m+1,1} + \mathbf{v}_{m+1,2}\|$. Therefore, $q_1(\mu)$ in (2.49) and $q_2(\mu)$ in (2.50), respectively, can be used as stopping criteria for residuals while Algorithm 2.1 is applied to solved QEPs (2.13).

Similarly, we can apply Algorithm 2.1 to solve REPs (2.20). As above, we let $(\mu^{-1}, \mathbf{z})$ be a Ritz pair and satisfy $H_m\mathbf{z} = \mu^{-1}\mathbf{z}$. From (2.43), and **R-SEP1**, **R-**

SEP2 in (2.31) we have

$$\begin{bmatrix} \mathbf{0} & -\widetilde{M}_p & L_p \\ \widetilde{K}_p^{-1} & -\widetilde{K}_p^{-1}\widetilde{D}_p & \mathbf{0} \\ \mathbf{0} & -\varrho^{-1}R_p^\top & \varrho^{-1}\vartheta I_\ell \end{bmatrix} \begin{bmatrix} V_{m1} \\ V_{m2} \\ V_{m3} \end{bmatrix} \mathbf{z} = \frac{1}{\mu} \begin{bmatrix} V_{m1} \\ V_{m2} \\ V_{m3} \end{bmatrix} \mathbf{z} + h_{m+1,m} \begin{bmatrix} \mathbf{v}_{m+1,1} \\ \mathbf{v}_{m+1,2} \\ \mathbf{v}_{m+1,3} \end{bmatrix} \mathbf{e}_m^\top \mathbf{z} \tag{2.51}$$

and

$$\begin{bmatrix} \mathbf{0} & -\widetilde{M}_p\widetilde{K}_p^{-1} & \varrho^{-1}L_p \\ I_{n_2} & -\widetilde{D}_p\widetilde{K}_p^{-1} & \mathbf{0} \\ \mathbf{0} & -R_p^\top\widetilde{K}_p^{-1} & \varrho^{-1}\vartheta I_\ell \end{bmatrix} \begin{bmatrix} V_{m1} \\ V_{m2} \\ V_{m3} \end{bmatrix} \mathbf{z} = \frac{1}{\mu} \begin{bmatrix} V_{m1} \\ V_{m2} \\ V_{m3} \end{bmatrix} \mathbf{z} + h_{m+1,m} \begin{bmatrix} \mathbf{v}_{m+1,1} \\ \mathbf{v}_{m+1,2} \\ \mathbf{v}_{m+1,3} \end{bmatrix} \mathbf{e}_m^\top \mathbf{z}, \tag{2.52}$$

where $V_m = [V_{m1}^\top, V_{m2}^\top, V_{m3}^\top]^\top$ and $\mathbf{v}_{m+1} = [\mathbf{v}_{m+1,1}^\top, \mathbf{v}_{m+1,2}^\top, \mathbf{v}_{m+1,3}^\top]^\top$ are partitioned with compatible sizes. Using the first and the third equations of (2.51) and (2.52), we can eliminate $V_1\mathbf{z}$ and $V_3\mathbf{z}$ in the second equation of (2.51) and (2.52), respectively, and get

$$\begin{aligned} \frac{\|R(\lambda)\mathbf{p}_1\|}{\|\mathbf{p}_1\|} &= \frac{\|[\mu^2\widetilde{M}_p + \mu\widetilde{D}_p + \widetilde{K}_p - \mu^2(\vartheta - \varrho\mu^{-1})^{-1}A_p]\mathbf{p}_1\|}{\|\mathbf{p}_1\|} \\ &= \frac{|\mu|\,|h_{m+1,m}|\,|\mathbf{e}_m^\top\mathbf{z}|\,\xi_1}{\|\mathbf{p}_1\|} \equiv r_1(\mu), \end{aligned} \tag{2.53}$$

where $\mathbf{p}_1 = V_{m2}\mathbf{z}$, $\lambda = \mu + \sigma$ and $\xi_1 = \|\mu\mathbf{v}_{m+1,1} + \widetilde{K}_p\mathbf{v}_{m+1,2} - \frac{\varrho\mu^2}{\vartheta\mu-\varrho}L_p\mathbf{v}_{m+1,3}\|$, and

$$\begin{aligned} \frac{\|R(\lambda)\mathbf{p}_2\|}{\|\mathbf{p}_2\|} &= \frac{\|\left[\mu^2\widetilde{M}_p + \mu\widetilde{D}_p + \widetilde{K}_p - \mu^2(\vartheta - \varrho\mu^{-1})^{-1}A_p\right]\mathbf{p}_2\|}{\|\mathbf{p}_2\|} \\ &= \frac{|\mu|\,|h_{m+1,m}|\,|\mathbf{e}_m^\top\mathbf{z}|\,\xi_2}{\|\mathbf{p}_2\|} \equiv r_2(\mu), \end{aligned} \tag{2.54}$$

where $\mathbf{p}_2 = \widetilde{K}_p^{-1}V_{m2}\mathbf{z}$, $\lambda = \mu + \sigma$ and $\xi_2 = \|\mu\mathbf{v}_{m+1,1} + \mathbf{v}_{m+1,2} - \frac{\mu^2}{\vartheta\mu-\varrho}L_p\mathbf{v}_{m+1,3}\|$. Therefore, $r_1(\mu)$ in (2.53) and $r_2(\mu)$ in (2.54) can be used as stopping criteria for residuals while Algorithm 2.1 is applied to solve REPs (2.20).

Applying Algorithm 2.1 to solve QEPs (2.13) and REPs (2.20) are summarized in Algorithm 2.2 and Algorithm 2.3, respectively.

Algorithm 2.2 Arnoldi method with Schur-restarting for solving QEP in (2.13)

```
Input: Coefficient matrices $M_u$, $D_u$ and $K_u$, parameters $c$, $\alpha$ and $\beta$, $\sigma$: shift value,
    tol$_Q$: tolerance for convergence, $r_{\max}$: maximum number of Schur-restartings.
Output: The desired eigenpairs $(\lambda_i, \mathbf{u}_i)$ for $i = 1, \dots, k$.
Construct matrices $\widetilde{M}_u$, $\widetilde{D}_u$ and $\widetilde{K}_u$ defined in (2.16) and set $r = 0$.
Compute initial Arnoldi decomposition in Line 1 of Algorithm 2.1 with $A$ in
    Q-SEP1 or Q-SEP2.
restart
    Do the steps in Lines 3 and 4 of Algorithm 2.1.
    for $i = 1, \dots, k$ do
        Compute
            $$\varphi(\mu_i) = (|\sigma + \mu_i^{-1}|^2 \|M_u\| + |\alpha + (\sigma + \mu_i^{-1})\beta| \|A_u\| + \|K_u\|).$$
        Check convergence of QEP by
            $$\frac{q_\ell(\mu_i)}{\varphi(\mu_i)} < \text{tol}_Q$$
        with $q_\ell(\mu_i)$ in (2.49) or (2.50), $\ell = 1, 2$.
    end for
    if ( Not all $k$ desired eigenvalues are convergent and $r < r_{\max}$ ) then
        Do the Schur-restarting in Lines 9–13 of Algorithm 2.1.
    end if
until ( desire $m$ eigenpairs are convergent or $r \geq r_{\max}$ )
Set $\lambda_i = \sigma + \mu_i^{-1}$ and $\mathbf{u}_i = V_{m2}\mathbf{z}_i$ for $i = 1, \dots, k$.
if Q-SEP2 is solved then
    $\mathbf{u}_i \leftarrow \widetilde{K}_u^{-1}\mathbf{u}_i$, $i = 1, \dots, k$.
end if
```

2.4.2 Computational costs

In this subsection, we compare the computational costs of the j-th Arnoldi step of Algorithm 2.1 for solving **Q-SEP**s (2.19) and **R-SEP**s (2.31), respectively. This is of general interest, because a comparison of the CPU time is sensible only if the number of outer iterations of Algorithm 2.2 or 2.3 is the same for each algorithm.

Algorithm 2.3 Arnoldi method with Schur-restarting for solving REP in (2.20)

```
Input: Coefficient matrices $M_p$, $K_p$ and $A_p$, parameters $c$, $\alpha$ and $\beta$, $\sigma$: shift value,
    tol$_R$: tolerance for convergence, $r_{\max}$: maximum number of Schur-restartings.
Output: The desired eigenpairs $(\lambda_i, \mathbf{p}_i)$ for $i = 1, \ldots, k$.
Construct matrices $\widetilde{M}_p$, $\widetilde{D}_p$ and $\widetilde{K}_p$ defined in (2.24), (2.25) and (2.26), respectively, and set $r = 0$.
Compute the full-rank decomposition of $A_p$: $L_p R_p^\top = A_p$.
Compute initial Arnoldi decomposition in Line 1 of Algorithm 2.1 with $A$ in R-SEP1 or R-SEP2.
restart
    Do the steps in Lines 3 and 4 of Algorithm 2.1.
    for $i = 1, \ldots, m$ do
        Compute
            $\psi(\mu_i) = |\frac{(\sigma + \mu_i^{-1})^2}{c^2}| \|M_p\| + \|K_p\| + |\frac{(\sigma + \mu_i^{-1})^2}{\alpha + (\sigma + \mu_i^{-1})\beta}| \|A_p\|.$
        Check convergence by
            $\frac{r_\ell(\mu_i)}{\psi(\mu_i)} < \mathrm{tol}_R$
        with $r_\ell(\mu_i)$ in (2.53) or (2.54), $\ell = 1, 2$.
    end for
    if ( Not all $k$ desired eigenvalues are convergent and $r < r_{\max}$ ) then
        Do the Schur-restarting in Lines 9–13 of Algorithm 2.1.
    end if
until ( desire $m$ eigenpairs are convergent or $r \geq r_{\max}$ )
Set $\lambda_i = \sigma + \mu_i^{-1}$ and $\mathbf{p}_i = V_{m2}\mathbf{z}_i$ for $i = 1, \ldots, k$.
if R-SEP2 is solved then
    $\mathbf{p}_i \leftarrow \widetilde{K}_p^{-1}\mathbf{p}_i$, $i = 1, \ldots, k$.
end if
```

From (2.47), the unit vector $\mathbf{v}_{j+1}$ is generated by

$$A\mathbf{v}_j = \sum_{i=1}^{j} h_{ji}\mathbf{v}_i + h_{j+1,j}\mathbf{v}_{j+1},$$

where $h_{ji} = \mathbf{v}_i^* A\mathbf{v}_j$ for $i = 1, \ldots, j$ and $h_{j+1,j} = \|A\mathbf{v}_j - \sum_{i=1}^{j} h_{ji}\mathbf{v}_i\|_2$. For convenience, we let $\mathbf{v}_j = \begin{bmatrix} \mathbf{v}_{j1} \\ \mathbf{v}_{j2} \end{bmatrix}$ with $\mathbf{v}_{j1}, \mathbf{v}_{j2} \in \mathbb{C}^n$. The matrix-vector product $A\mathbf{v}_j$ in Algorithm 2.2 for solving QEP (2.13) by **Q-SEP1** (2.19.1) and **Q-SEP2** (2.19.2) can be, respectively, represented by

$$\mathcal{B}_u^{-1}\mathcal{A}_u\mathbf{v}_j = \begin{bmatrix} -\widetilde{M}_u\mathbf{v}_{j2} \\ \widetilde{K}_u^{-1}(\mathbf{v}_{j1} - \widetilde{D}_u\mathbf{v}_{j2}) \end{bmatrix} \quad \text{and} \quad \mathcal{A}_u\mathcal{B}_u^{-1}\mathbf{v}_j = \begin{bmatrix} -\widetilde{M}_u\mathbf{g}_u \\ \mathbf{v}_{j1} - \widetilde{D}_u\mathbf{g}_u \end{bmatrix}$$

with $\mathbf{g}_u = \widetilde{K}_u^{-1}\mathbf{v}_{j2}$. This implies that Algorithm 2.2 for **Q-SEP1** and **Q-SEP2** needs the same computational costs for generating the unit vector $\mathbf{v}_{j+1}$ for each j.

On the other hand, by letting $\mathbf{v}_j = [\mathbf{v}_{j1}^\top, \mathbf{v}_{j2}^\top, \mathbf{v}_{j3}^\top]^\top$ with $\mathbf{v}_{j1}, \mathbf{v}_{j2} \in \mathbb{C}^n$ and $\mathbf{v}_{j3} \in \mathbb{C}^\ell$, the matrix-vector product $A\mathbf{v}_j$ in Algorithm 2.3 for solving REPs by **R-SEP1** (2.31.1) and **R-SEP2** (2.31.2) can be, respectively, represented by

$$\mathcal{B}_p^{-1}\mathcal{A}_p\mathbf{v}_j = \begin{bmatrix} L_p\mathbf{v}_{j3} - \widetilde{M}_p\mathbf{v}_{j2} \\ \widetilde{K}_p^{-1}(\mathbf{v}_{j1} - \widetilde{D}_p\mathbf{v}_{j2}) \\ \varrho^{-1}\vartheta\mathbf{v}_{j3} - \varrho^{-1}R_p^\top\mathbf{v}_{j2} \end{bmatrix} \quad \text{and} \quad \mathcal{A}_p\mathcal{B}_p^{-1}\mathbf{v}_j = \begin{bmatrix} \varrho^{-1}L_p\mathbf{v}_{j3} - \widetilde{M}_p\mathbf{g}_p \\ \mathbf{v}_{j1} - \widetilde{D}_p\mathbf{g}_p \\ \varrho^{-1}\vartheta\mathbf{v}_{j3} - R_p^\top\mathbf{g}_p \end{bmatrix}$$

with $\mathbf{g}_p = \widetilde{K}_p^{-1}\mathbf{v}_{j2}$. Consequently, the computational cost of $\mathcal{A}_p\mathcal{B}_p^{-1}\mathbf{v}_j$ needs an extra cost for the computation of $\varrho^{-1}L_p\mathbf{v}_{j3}$ compared to that $\mathcal{B}_p^{-1}\mathcal{A}_p\mathbf{v}_j$. The cost for generating the unit vector $\mathbf{v}_{j+1}$ by **R-SEP1** is slightly cheaper than that by **R-SEP2**. We summarize the computational costs of generating $\mathbf{v}_{j+1}$ for by **Q-SEP2** and **R-SEP2** in Table 2.1.

Remark 2.4. *In the numerical implementation, the vectors* $\mathbf{g}_u = \widetilde{K}_u^{-1}\mathbf{v}_{j2}$ *and*

	Q-SEP2 (2.19.2)	**R-SEP2** (2.31.2)
Solving linear system	$\widetilde{K}_u\mathbf{x}_u = \mathbf{b}_u$	$\widetilde{K}_p\mathbf{x}_p = \mathbf{b}_p$
Matrix-vector products	$\widetilde{M}_u\mathbf{b}_u,\ \widetilde{D}_u\mathbf{b}_u$	$\widetilde{M}_p\mathbf{b}_p,\ \widetilde{D}_p\mathbf{b}_p,\ L_p\mathbf{c}_p,\ R_p^\top\mathbf{c}_p^\top$
Inner products	$j+1$	$j+1$
Saxpy operators	$j+1$	$j+2$
Scale-vector product	1	1

Table 2.1: Computational costs of the j-th Arnoldi step of Algorithm 2.1 for **Q-SEP2** and **R-SEP2**.

$\mathbf{g}_p = \widetilde{K}_p^{-1}\mathbf{v}_{j2}$ *for* $j = 1, \ldots, k$ *can be saved in* $G_u \equiv [\widetilde{K}_u^{-1}\mathbf{v}_{12} \ \cdots \ \widetilde{K}_u^{-1}\mathbf{v}_{m2}]$ *and* $G_p \equiv [\widetilde{K}_p^{-1}\mathbf{v}_{12} \ \cdots \ \widetilde{K}_p^{-1}\mathbf{v}_{m2}]$*, respectively, so that the vectors* $\mathbf{u}_2$*,* $\mathbf{p}_2$ *in* (2.50) *and* (2.54) *can be computed by* $\mathbf{u}_2 = G_u\mathbf{z}$ *and* $\mathbf{p}_2 = G_p\mathbf{z}$ *directly. Hence, it requires the same computational costs for computing* $\mathbf{u}_1$*,* $\mathbf{u}_2$ *in* (2.49) *and* (2.50)*, as well as,* $\mathbf{p}_1$*,* $\mathbf{p}_2$ *in* (2.53) *and* (2.54)*, respectively. Consequently, the computational costs of **Q-SEP1** for the convergence test in Algorithm 2.2 need one extra matrix-vector product* $\widetilde{K}_u\mathbf{v}_{m+1,2}$ *than those of **Q-SEP2** in computing* ζ_1 *and* ζ_2*. Similarly, the computational costs of **R-SEP1** for the convergence test in Algorithm 2.3 need one extra matrix-vector product* $\widetilde{K}_p\mathbf{v}_{m+1,2}$ *than those of **R-SEP2** in computing* ξ_1 *and* ξ_2*. Therefore, we conclude that Algorithm 2.2 for **Q-SEP1** and **Q-SEP2**, as well as, Algorithm 2.3 for **R-SEP1** and **R-SEP2**, respectively, almost have the same computational costs provided that they have the same outer iterations.*

2.5 Numerical Results

We conduct numerical experiments to evaluate performance and accuracy of the eigenvalue solvers described in Section 2.4. To distinguish between various eigenvalue problems, we use notations **Q1**, **Q2**, **R1** and **R2** defined as follows:

- **Q1**: Applying Algorithm 2.2 to solve the QEP (2.13) with **Q-SEP1** in (2.19.1).

- **Q2**: Applying Algorithm 2.2 to solve the QEP (2.13) with **Q-SEP2** in (2.19.2).
- **R1**: Applying Algorithm 2.3 to solve the REP (2.20) with **R-SEP1** in (2.31.1).
- **R2**: Applying Algorithm 2.3 to solve the REP (2.20) with **R-SEP2** in (2.31.2).

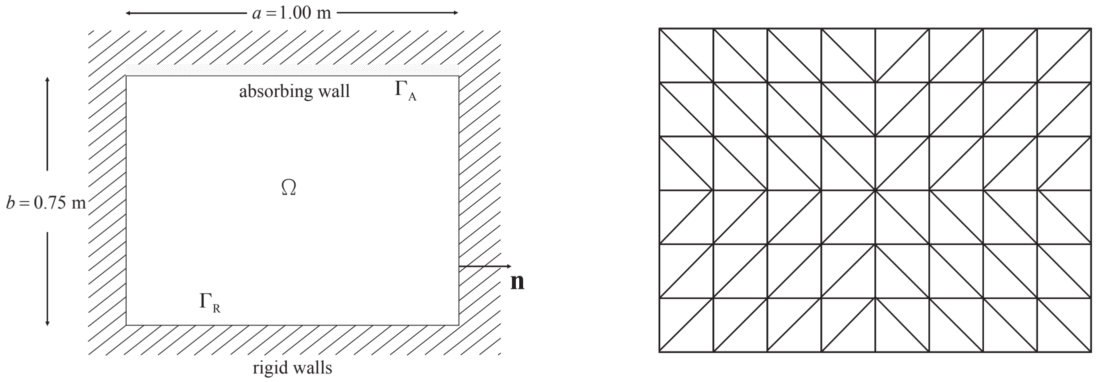
All computations are carried out in MATLAB 2009a on a HP workstation with an Intel Quad-Core Xeon X5570 2.93GHz and 72 GB main memory, using IEEE double-precision floating-point arithmetic. We apply Algorithms 2.2 and 2.3 to solve the following examples arising in fluid-solid systems. The order m of Arnoldi decomposition in Line 3 of Algorithm 2.1 is set $m = 40$, the maximum number $r_{\max}$ of Schur-restartings is set $r_{\max} = 15$ and the number of desired eigenpairs is $k = 10$. The relative residuals of approximate eigenpairs $(\lambda_i, \mathbf{u}_i)$ and $(\lambda_i, \mathbf{p}_i)$ computed by **Q1** and **Q2**, as well as, **R1** and **R2** are, respectively, defined by

$$\frac{\|Q(\lambda_i)\mathbf{u}_i\|}{\varphi(\lambda_i)\|\mathbf{u}_i\|} \quad \text{and} \quad \frac{\|R(\lambda_i)\mathbf{p}_i\|}{\psi(\lambda_i)\|\mathbf{p}_i\|},$$

where $\varphi(\lambda_i)$ and $\psi(\lambda_i)$ are given in Algorithm 2.2 and 2.3, respectively. Tolerances for relative residuals of QEPs and REPs are chosen by $\text{tol}_Q = \text{tol}_R = 5 \times 10^{-15}$. The linear systems in Algorithms 2.2 and 2.3 are solved by LU-factorization with the shift value $\sigma = -25 + 600\pi\mathtt{i}$. Fronbenius norm for matrices and 2-norm for vectors are used.

Example 2.1. [6] We take the geometrical data: the domain $\Omega = [0\mathtt{m}, 1\mathtt{m}] \times [-0.75\mathtt{m}, 0\mathtt{m}]$, $\Gamma_A = [0\mathtt{m}, 1\mathtt{m}] \times \{0\mathtt{m}\}$ given in Figure 2.1(i) and the following physical data: $\rho = 1\mathtt{kg}/\mathtt{m}^3$, $c = 340\ \mathtt{m}/\mathtt{s}$, $\alpha = 5 \times 10^4\ \mathtt{N}/\mathtt{m}^3$, and $\beta = 200\ \mathtt{Ns}/\mathtt{m}^3$.

The rectangular domain Ω is uniformly partitioned into n_ℓ by n_w rectangles and each rectangle is further refined into two triangles, see Figure 2.1(ii). The dimensions of coefficient matrices in QEP (2.13) and REP (2.20) are $(3n_\ell - 1) \times n_w$ and



(i) Fluid in a cavity with one absorbing wall. **(ii)** Initial mesh.

Figure 2.1: Fluid in a cavity with one absorbing wall and initial mesh

$(n_\ell+1)\times(n_w+1)$, respectively. Figure 2.2 plots the analytic solutions of the desired eigenvalues $\lambda_1,\ldots,\lambda_{10}$ of (2.5)–(2.8) (see [6]) with the lowest positive vibration frequencies satisfying $0 < \frac{\mathrm{Im}(\lambda_i)}{2\pi} < 600\mathrm{Hz}$.

Convergence test: We first demonstrate convergence rates of **Q2** and **R2** while computing the desired eigenvalues in Figure 2.2. To measure the convergence rate, we run the test over the five successively refined meshes (See the first column of Table 2.2) and then calculate the rates by

$$rate_{[i,j]} = \log_2\left(\frac{|\lambda_{[i,j]} - \lambda_{[i,j+1]}|}{|\lambda_{[i,j+1]} - \lambda_{[i,j+2]}|}\right), \quad \text{for} \ \ i = 1,\ldots,10, \ j = 1,2,3,$$

where $\lambda_{[i,j]}$ for $j = 1,\ldots,5$ denote the approximate eigenvalues computed by **Q2** and **R2** corresponding to λ_i obtained from the meshes described in Table 2.2. The 5-th and the 6-th columns of Table 2.2 illustrate the quadratic convergence of $rate_{[1,j]}$ $j = 1,2,3$ for λ_1 of QEP (2.13) and REP (2.20), respectively. In our numerical experiment, the convergence rate are always close to 2 for all desired eigenvalues, λ_i, $i = 1,\ldots,10$, computed by **Q2** and **R2** as well as **Q1** and **R1**.

(n_ℓ, n_w)	Matrix size (QEP)	Matrix size (REP)		Conv. rate	
	$(3n_\ell - 1) \times n_w$	$(n_\ell + 1) \times (n_w + 1)$	λ_1	**Q2**	**R2**
(48, 36)	$5,148$	$1,813$			
(96, 72)	$20,664$	$7,081$			
(192, 144)	$82,800$	$27,985$	$rate_{[1,1]}$	1.9979	2.0010
(384, 288)	$331,488$	$111,265$	$rate_{[1,2]}$	1.9995	2.0003
(768, 576)	$1,326,528$	$443,713$	$rate_{[1,3]}$	1.9999	2.0001

Table 2.2: Dimension information and convergence rates of λ_1.

Normwise scaling of QEP: Balancing norms of coefficient matrices is an important issue [66] before solving a QEP of the form

$$P(\lambda)\mathbf{x} \equiv (\lambda^2 P_2 + \lambda P_1 + P_0)\mathbf{x} = \mathbf{0}. \tag{2.55}$$

In [15] authors give an elegant way to scale the norms of coefficient matrices of (2.55) as follows. Define

$$\widehat{P}(\nu)\mathbf{x} \equiv (\nu^2 \widehat{P}_2 + \nu \widehat{P}_1 + \widehat{P}_0)\mathbf{x} = \mathbf{0}$$

with $\nu = \lambda/\zeta$, $\widehat{P}_2 = \zeta^2 \eta P_2$, $\widehat{P}_1 = \zeta \eta P_1$ and $\widehat{P}_0 = \eta P_0$, where ζ and η are scaling factors. Taking ζ and η as $\zeta_* = \sqrt{\gamma_0/\gamma_2}$ and $\eta_* = 2/(\gamma_0 + \gamma_1 \zeta_*)$ with $\gamma_2 := \|P_2\|_2$, $\gamma_1 := \|P_1\|_2$, $\gamma_0 := \|P_0\|_2$, it is proved in [15] that the problem

$$\min_{\zeta,\eta} \max \left\{ |\|\widehat{P}_2\|_2 - 1|, |\|\widehat{P}_1\|_2 - 1|, |\|\widehat{P}_0\|_2 - 1| \right\}$$

achieves the optimum at ζ_* and η_*. In our implementation, the values of γ_i, for $i = 0, 1, 2$ are computed by $\gamma_2 = \|\widetilde{M}_u\|_F$, $\gamma_1 = \|\widetilde{D}_u\|_F$, $\gamma_0 = \|\widetilde{K}_u\|_F$ and $\gamma_2 = \|\widetilde{M}_p\|_F$, $\gamma_1 = \|\widetilde{D}_p\|_F$, $\gamma_0 = \|\widetilde{K}_p\|_F$ for QEP (2.15) and REP (2.29), respectively. We denote "#It" the number of Schur-restartings (outer iterations). In Table 2.3, we show #Its for computing 10 desired eigenvalues of Example 2.1 with $(n_\ell, n_w) =$

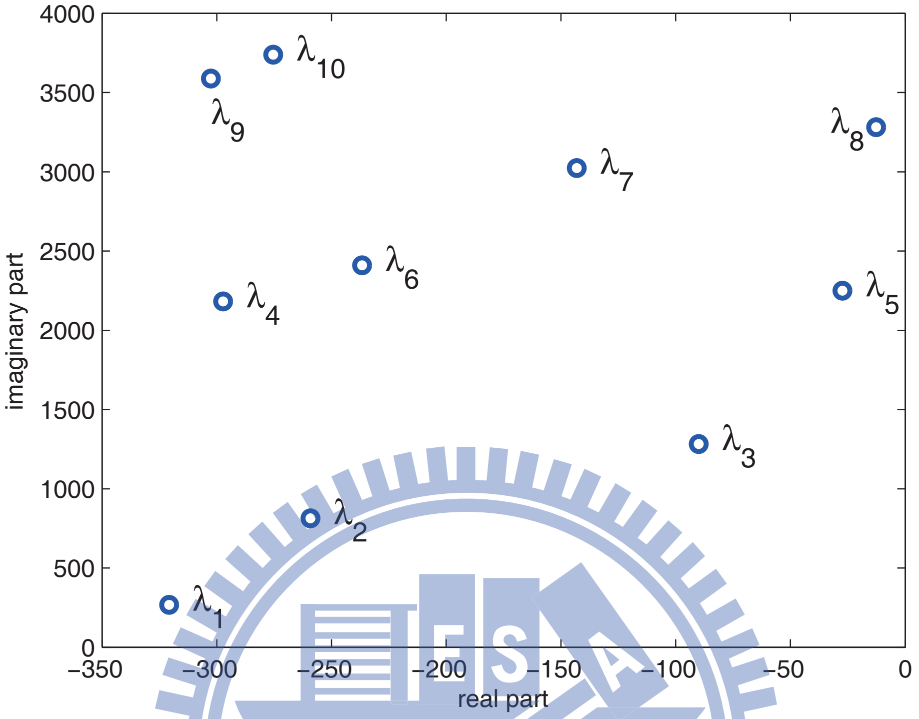


Figure 2.2: The distribution of the ten desired eigenvalues $\lambda_1, \ldots, \lambda_{10}$.

$(768, 576)$ by **Q1**, **Q2**, **R1** and **R2** with/without scaling. The tolerances tol_Q and tol_R for relative residuals are chosen to be 5×10^{-15}. We see that the convergence rate of scaled Q-SEPs or R-SEPs is faster than that of unscaled Q-SEPs or R-SEPs. The performance of **Q2** and **R2** is also better than that of **Q1** and **R1**, respectively. In the case of unscaled REP, the norms of $\widetilde{M}_p$, $\widetilde{D}_p$ and $\widetilde{K}_p$ in (2.24)–(2.26) are $\mathcal{O}(10^{-10})$, $\mathcal{O}(10^{-5})$ and $\mathcal{O}(1)$, respectively. Since the norms of coefficient matrices vary too much, **R1** can even fail to converge to 10 eigenpairs after 15 outer iterations.

	Q1	**Q2**	**R1**	**R2**
#It (scaled)	3	2	4	3
#It (unscaled)	4	3	15	3

Table 2.3: #Its for $\lambda_1, \ldots, \lambda_{10}$ of Q-SEPs and R-SEPs with/without scaling.

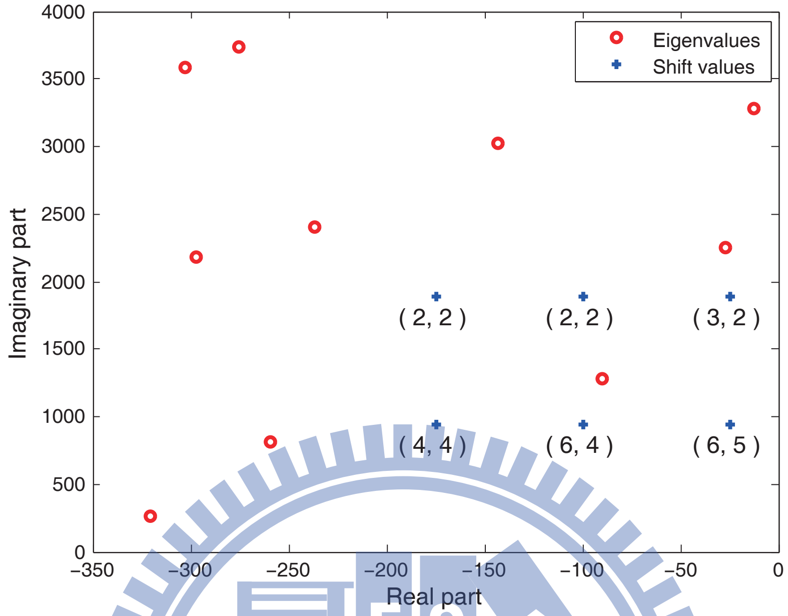


Figure 2.3: The #Its of **Q1** and **Q2** with different shift values. "o" denotes desired eigenvalues $\lambda_1, \ldots, \lambda_{10}$. "$(i, j)$" denotes the #Its for **Q1** and **Q2**, respectively.

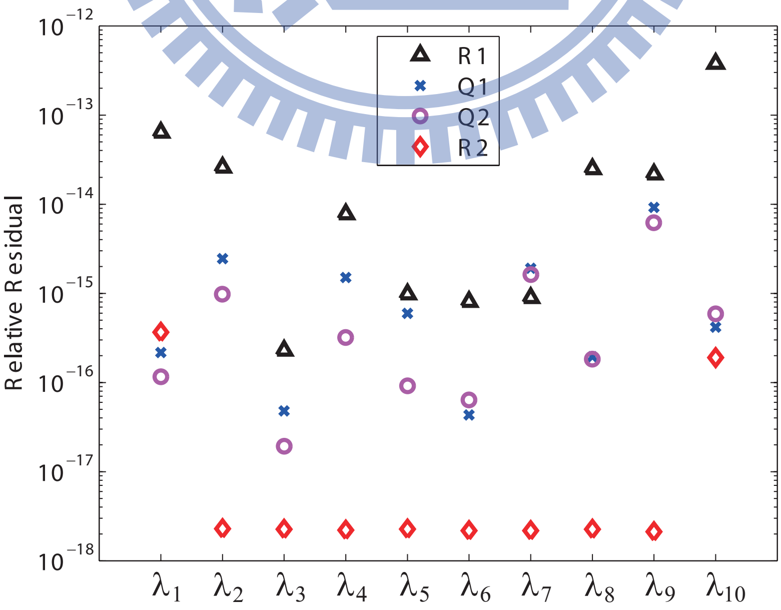


Figure 2.4: The relative residuals of computed eigenpairs, obtained by **Q1**, **Q2** for QEP (2.13) and **R1**, **R2** for REP (2.20) with $(n_\ell, n_w) = (768, 576)$.

	Q2		**R2**		$\frac{T_{R2}}{T_{Q2}}$
(n_ℓ, n_w)	#It	T_{Q2}	#It	T_{R2}	
(48, 36)	2	1.316	2	0.471	0.36
(96, 72)	2	7.717	2	2.387	0.31
(192, 144)	2	55.27	2	14.95	0.27
(384, 288)	2	567.8	2	134.0	0.24
(768, 576)	2	8152	2	1645	0.20

Table 2.4: Iteration numbers and CPU time for **Q2** and **R2**.

No spurious eigenmodes: In [6], it has been proved that there are no spurious eigenmodes for the discretization based on Raviart-Thomas finite elements. We compute twenty desired eigenvalues of QEP (2.13) and REP (2.20) by **Q2** and **R2**, respectively, with scaling and various mesh sizes as shown in Table 2.2 (we computed 20 instead of 10 eigenvalues to be better confirmed). The desired eigenvalues of REP are in one-to-one correspondence to those of QEP which match well with relative error less than 10^{-6}, that is, no spurious eigenmodes ever appear. We numerically conclude that there are no spurious eigenmodes for the discretization in terms of pressure nodal finite elements.

Null space considerations: Theorem 2.1 shows that the dimension of the null space of QEP (2.13) is equal to the number of interior nodes, i.e., $(n_\ell - 1)(n_w - 1)$. In order to observe the interference of such a large null space in the convergence of **Q1** and **Q2**, we give six different shift values denoted by the "+" in Figure 2.3 to observe variation in the #Its for **Q1** and **Q2**. The integer pair (i, j) under each shift value "+" denotes the #Its for **Q1** and **Q2**, respectively. The results in Figure 2.3 demonstrate that the #It needed decreases, as the shift value σ is chosen relatively far away from zero.

Comparison of pressure and displacement formulation: In this paragraph, we shall discuss the advantages of using the nodal pressure finite elements with various mesh sizes described in Table 2.2. The notations "T_{Q2}" and "T_{R2}" denote the total CPU time for **Q2** and **R2**, respectively. We summarize the results as follows:

- Accuracy of eigenpairs: From Remark 2.3, the upper bound for relative residual of the approximate eigepairs of QEP (2.13) (or REP (2.20)) by using **Q-SEP2** (2.19.2) (or **R-SEP2** (2.31.2)) is much smaller than that by using **Q-SEP1** (2.19.1) (or **R-SEP1** (2.31.1)). On applying **Q1** and **Q2** to solve QEP (2.13) with #It = 2, in Figure 2.4, we see that the relative residuals of eigenpairs corresponding to λ_4 and λ_5 computed by **Q2** are improved by about 1 significant digit than those by **Q1**. The other eigenpairs almost have the same accuracy. On applying **R1** and **R2** to solve REP (2.20) with #It = 2, in Figure 2.4, we see that the relative residuals of eigenpairs computed by **R2** are improved by about 2 to 4 significant digits than those by **R1**.
- Comparison **R2** with **Q2**: From Subsection 2.4.2 we see that **Q1** and **Q2**, as well as, **R1** and **R2** have the same computational costs, respectively. From Figure 2.4, we favor applying **Q2** and **R2** to solve QEP (2.13) and REP (2.20), respectively. From column 12 of Table 2.4, we see that the CPU time for solving the REP (2.20) by **R2** is only 1/5 to 1/3 of that for solving the QEP (2.13) by **Q2**. The accuracy of the computed eigenpairs for REP (2.20) is also better than that of QEP (2.13). These results tell us that using **R2** to solve nodal pressure finite elements for the discrete problem (2.12) is better than that using **Q2** to solve Raviart-Thomas displacement finite elements for the discrete problem (2.11).

We now want to apply our methods to a more complicated configuration in which the absorbing walls are located on three sides.

Example 2.2. We use the same geometric data and physical data in Example 2.1 except that the absorbing wall is extended to one half of the rigid walls in the left and right boundaries, that is $\Gamma = [0, 1] \times \{0\} \cup \{0\} \times [-0.375, 0] \cup \{1\} \times [-0.375, 0]$.

In Example 2.1, we numerically show that there are no spurious eigenmodes for the discretization in terms of pressure nodal finite elements. Moreover, the computational cost for solving the associated REP (2.20) is obviously less than that of solving QEP (2.13) which is obtained from using Raviart-Thomas displacement finite elements to the discrete problem (2.11). Therefore, in this example we only use nodal finite elements to discretize the model and compare the accuracy of **R1** and **R2** for solving the associated REP. The computed eigenvalues $\lambda_1, \ldots, \lambda_{10}$ with lowest positive vibration frequencies satisfying $0 < \frac{\mathrm{Im}(\lambda_i)}{2\pi} < 600\mathrm{Hz}$ are shown in Figure 2.5. The convergence rates for $\lambda_1, \ldots, \lambda_{10}$ obtained from various the mesh sizes described in Table 2.2 are also close to 2. The relative residuals computed by **R1** and **R2** are presented in Figure 2.6 which shows that the accuracy of the eigenpairs produced by **R2** is better than **R1**.

2.6 Summary

We consider the problem for computing damped vibration modes of an acoustic fluid confined in a cavity, with absorbing walls capable of dissipating acoustic energy. The discretization in terms of edge-based finite elements for the displacement field induces the QEP (2.13) and the pressure nodal finite elements gives rise to the REP (2.20). We utilize the linearization process to rewrite these two nonlinear eigenvalue problems into four different types of SEPs, namely **Q1**, **Q2**, **R1** and **R2**, which have defined in Section 2.5. From these numerical results, we have the following conclusions.

1. There are no spurious eigenmodes for the discretization in terms of pressure nodal finite elements.

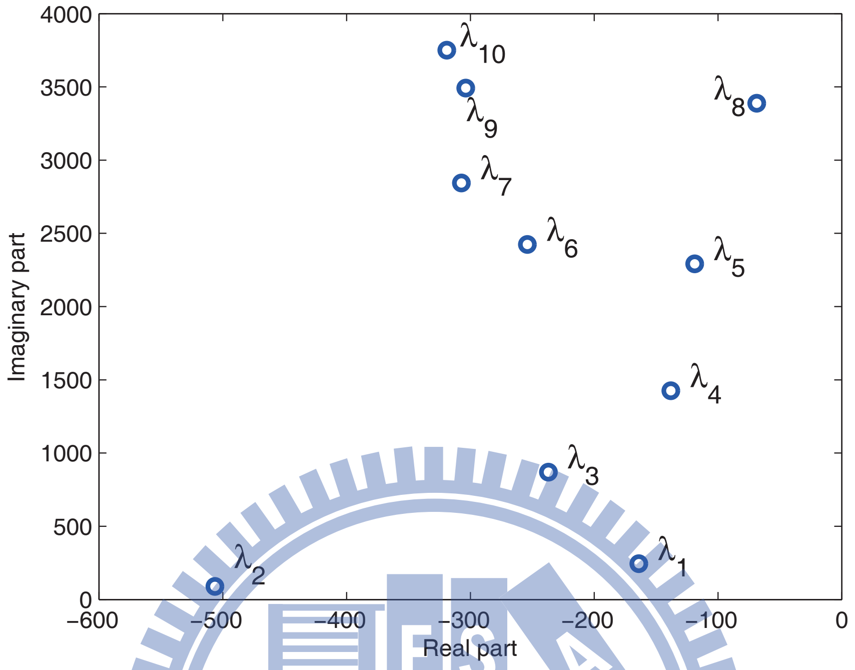


Figure 2.5: The distribution of the ten desired eigenvalues $\lambda_1, \ldots, \lambda_{10}$ for Example 2.2.

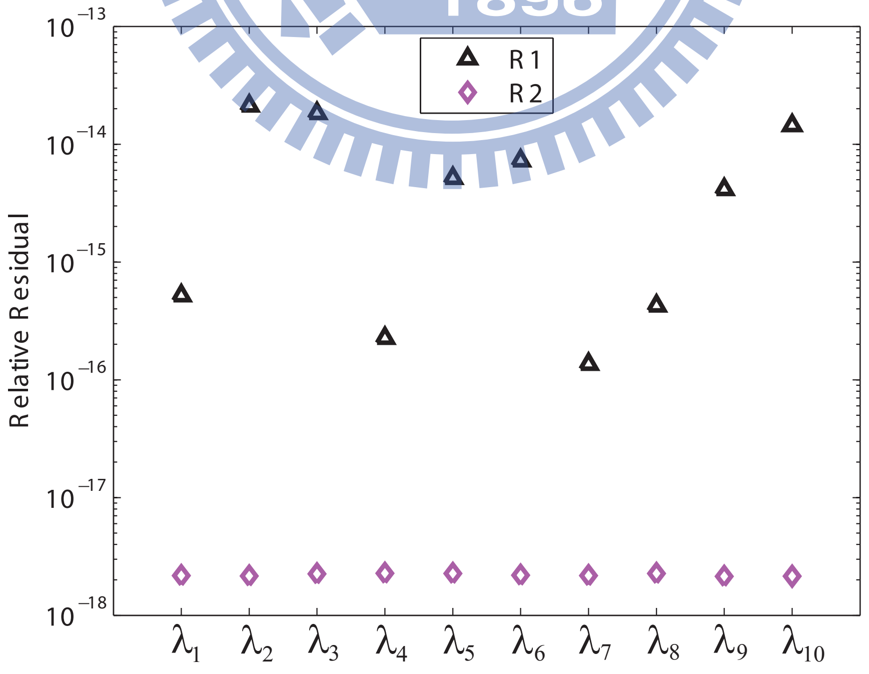


Figure 2.6: Relative residuals of computed eigenpairs obtained by **R1** and **R2** for REP in Example 2.2 with $(n_\ell, n_w) = (768, 576)$.

2. The dimension of the null space associated with the edge-based displacement (QEP) equals the number of interior nodes in the triangulation; the nodal-based pressure model (REP), however, only has one dimensional null space.

3. The convergence of the eigensolver for the QEP would be disturbed by a large null space when the shift value is close to zero.

4. The size of the QEP is larger than the size of the REP.

5. The CPU time for solving the corresponding REP (2.20) are only 1/5 to 1/3 of the CPU time for solving the QEP (2.13).

6. The accuracy of **Q2** and **R2** algorithms are better than **Q1** and **R1** respectively.

3

The Semiorthogonal Generalized Arnoldi (SGA) Method for Quadratic Eigenvalue Problems

Contents

3.1 Introduction

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

$$(\lambda^2 M + \lambda D + K)\mathbf{x} = \mathbf{0}, \tag{3.1}$$

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 (3.1).

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 [14, section 11.9], [66] and the references therein.

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

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

where $\lambda_\sigma = \lambda - \sigma$, $M_\sigma = M$, $D_\sigma = 2\sigma M + D$ and $K_\sigma = \sigma^2 M + \sigma D + K$. For simplicity, we assume, without loss of generality, that $\sigma = 0$. Therefore, throughout this chapter, we delve in 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 (3.1) or, equivalently, that K is nonsingular.

Through the so-called "linearization" process, one may first construct a suitable matrix pair $(\mathcal{A}, \mathcal{B})$ of size $2n$ and a vector $\boldsymbol{\varphi}$ in $\mathbb{C}^{2n}$ to rewrite the QEP (3.1) equivalently into a generalized eigenvalue problem (GEP)

$$\mathcal{A}\boldsymbol{\varphi} = \frac{1}{\lambda}\mathcal{B}\boldsymbol{\varphi}. \tag{3.2}$$

If $\mathcal{B}$ is chosen to be nonsingular, one can further reduce (3.2) as a standard eigenvalue problem (SEP)

$$(\mathcal{B}^{-1}\mathcal{A})\boldsymbol{\varphi} = \frac{1}{\lambda}\boldsymbol{\varphi} \tag{3.3}$$

or

$$(\mathcal{A}\mathcal{B}^{-1})\boldsymbol{\psi} = \frac{1}{\lambda}\boldsymbol{\psi}, \tag{3.4}$$

where $\boldsymbol{\psi} = \mathcal{B}\boldsymbol{\varphi}$. We call (3.3) and (3.4) 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 [66].

The way of linearization is not unique [66]. Here, we consider the second companion form of linearization [21] for the QEP (3.1)

$$\begin{bmatrix} -D & I_n \\ -M & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix}, \tag{3.5}$$

where $\widetilde{\mathbf{x}} = -\lambda M\mathbf{x}$. The computational advantage of using the second companion form will be revealed in section 3.3.

Since K is nonsingular, from (3.5) the corresponding ℓ-SEP and r-SEP of (3.1) are, respectively, given by

$$\begin{bmatrix} -K^{-1}D & K^{-1} \\ -M & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix} \tag{3.6}$$

and

$$\begin{bmatrix} -DK^{-1} & I_n \\ -MK^{-1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} K\mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} K\mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix}. \tag{3.7}$$

In addition to solving the QEP (3.1) by SEP (3.6) or (3.7), one may also work with the GEP (3.5) to find the desired eigenpairs of (3.1). The QZ algorithm [45] is the most prevalent algorithm for solving the dense GEP of the form (3.2). This procedure reduces the matrix pair $(\mathcal{A}, \mathcal{B})$ equivalently to a Hessenberg-triangular pair (H, R) via unitary transformations in a finite number of steps. This truncated QZ method proposed by Sorensen [60] 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 [26], 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 [65].

In this chapter, we introduce a *Semiorthogonal Generalized Arnoldi* (SGA) algorithm for the particular linearized problem (3.5) to generate a SGA decomposition. The SGA algorithm is a variation of the generalized Arnoldi reduction [60]. We then propose an orthogonal projection approach termed as the SGA method to solve the QEP (3.1) where the projection subspace is defined through its orthonormal basis obtained from the SGA decomposition. We will extend this idea of refinement in [32] 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 [60], 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 (3.6) and (3.7), the SGA-type methods, namely IRSGA and IRRSGA, demonstrate better convergence behaviors and require less CPU time in numerical experiments.

This chapter is organized as follows. In section 3.2, we first introduce the SGA algorithm associated with the GEP (3.5). In section 3.3, we propose an orthogonal projection method based on the orthonormal basis generated by the SGA algorithm for solving the QEP (3.1). In section 3.4, we present a refinement scheme to get better Ritz vectors by taking advantage of the SGA decomposition. In section 3.5, 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 3.6 and the concluding remarks are given in section 3.7.

3.2 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 3.1 (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* (3.1) *to be the relation of the form*

$$\begin{bmatrix} -D & I_n \\ -M & \mathbf{0} \end{bmatrix} \begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} V_m \\ U_m \end{bmatrix} H_m + \begin{bmatrix} \mathbf{g}_m \\ \mathbf{f}_m \end{bmatrix} \mathbf{e}_m^{\top}, \tag{3.8a}$$

$$\begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix} \begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} V_m \\ U_m \end{bmatrix} R_m, \tag{3.8b}$$

$$Q_m^H Q_m = I_m, \quad V_m^H V_m = I_m \quad \textit{and} \quad V_m^H \mathbf{g}_m = \mathbf{0}, \tag{3.8c}$$

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 3.2. *(i) The orthogonality requirements in* (3.8c) *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 (3.8c) *is replaced by*

$$Q_m^H Q_m + P_m^H P_m = I_m, \quad V_m^H V_m + U_m^H U_m = I_m \quad \textit{and} \quad V_m^H \mathbf{g}_m + U_m^H \mathbf{f}_m = \mathbf{0},$$

we actually obtain a generalized Arnoldi reduction [60] associated with the GEP (3.5)*. Therefore, the SGA decomposition can be also viewed as a variation of the generalized Arnoldi reduction.*

3.2.1 Existence and uniqueness

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

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

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

From (3.8), if we set

$$\mathcal{A} = \begin{bmatrix} -D & I_n \\ -M & \mathbf{0} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix}, \tag{3.9}$$

and

$$Z_m = \begin{bmatrix} Q_m \\ P_m \end{bmatrix}, \quad Y_m = \begin{bmatrix} V_m \\ U_m \end{bmatrix}, \quad \boldsymbol{\eta}_m = \begin{bmatrix} \mathbf{g}_m \\ \mathbf{f}_m \end{bmatrix}, \tag{3.10}$$

then the SGA decomposition (3.8) can be compactly written as

$$\mathcal{A}Z_m = Y_m H_m + \boldsymbol{\eta}_m \mathbf{e}_m^\top, \tag{3.11a}$$

$$\mathcal{B}Z_m = Y_m R_m, \tag{3.11b}$$

$$Q_m^H Q_m = V_m^H V_m = I_m, \ V_m^H \mathbf{g}_m = \mathbf{0}. \tag{3.11c}$$

Using equations (3.8)–(3.10) of the SGA decomposition (3.11) and based on the proof technique of Theorem 3.3 in [17], we give the following theorem.

Theorem 3.3. *Given* $\mathbf{z}_1 \equiv \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{p}_1 \end{bmatrix} \in \mathbb{C}^{2n}$ *with* $\|\mathbf{q}_1\|_2 = 1$ *and set*

$$\mathcal{B}\mathbf{z}_1 = \rho_1 \mathbf{y}_1 \equiv \rho_1 \begin{bmatrix} \mathbf{v}_1 \\ \\ \mathbf{u}_1 \end{bmatrix}$$

with $\|\mathbf{v}_1\|_2 = 1$ *and* $\rho_1 > 0$. *Let*

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

Suppose that $\mathbb{K}_{\ell,1}$ *is of full column rank and* $\mathbb{K}_{\ell,1} = Q_m R_{\ell,m}$ *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*

(i) $\mathbb{K}_{r,1}$ *is of full column rank. Moreover, if* $\mathbb{K}_{r,1} = V_m R_{r,m}$ *is the* QR_+*-factorization, then* $V_m \mathbf{e}_1 = \mathbf{v}_1$.

(ii) Let

$$P_m = \mathbb{K}_{\ell,2} R_{\ell,m}^{-1} \quad \text{and} \quad U_m = \mathbb{K}_{r,2} R_{r,m}^{-1}.$$

Then there exist an unreduced upper Hessenberg matrix H_m *with positive subdiagonal entries and an upper triangular* R_m *with positive diagonal entries satisfying the SGA decomposition* (3.11).

(iii) The SGA decomposition (3.11) *is uniquely determined by* $Z_m \mathbf{e}_1 = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{p}_1 \end{bmatrix}$ *with* $\|\mathbf{q}_1\|_2 = 1$.

Proof. (i) Since

$$\begin{aligned}
\begin{bmatrix} \mathbb{K}_{r,1} \\ \mathbb{K}_{r,2} \end{bmatrix} &= \mathbb{K}[\![mathcalA\mathcal{B}^{-1}, \mathbf{y}_1, m]\!] = \begin{bmatrix} \mathbf{y}_1 & (\mathcal{A}\mathcal{B}^{-1})\mathbf{y}_1 & \cdots & (\mathcal{A}\mathcal{B}^{-1})^{m-1}\mathbf{y}_1 \end{bmatrix} \\
&= \frac{1}{\rho_1}\mathcal{B}\begin{bmatrix} \mathbf{z}_1 & (\mathcal{B}^{-1}\mathcal{A})\mathbf{z}_1 & \cdots & (\mathcal{B}^{-1}\mathcal{A})^{m-1}\mathbf{z}_1 \end{bmatrix} \\
&= \frac{1}{\rho_1}\mathcal{B}\mathbb{K}_\ell = \frac{1}{\rho_1}\begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix}\begin{bmatrix} \mathbb{K}_{\ell,1} \\ \mathbb{K}_{\ell,2} \end{bmatrix} = \frac{1}{\rho_1}\begin{bmatrix} K\mathbb{K}_{\ell,1} \\ \mathbb{K}_{\ell,2} \end{bmatrix},
\end{aligned} \tag{3.12}$$

the matrix $\mathbb{K}_{r,1} = \rho_1^{-1}K\mathbb{K}_{\ell,1}$ is of full column rank and has the unique QR_+-factorization $\mathbb{K}_{r,1} = V_m R_{r,m}$ with $V_m \mathbf{e}_1 = \mathbf{v}_1$.

(ii) By assumptions and (3.10), we get $\mathbb{K}_\ell = \left[\begin{smallmatrix} Q_m \\ P_m \end{smallmatrix}\right] R_{\ell,m} = Z_m R_{\ell,m}$. From (i) and (3.10), we also have $\mathbb{K}_r = \left[\begin{smallmatrix} V_m \\ U_m \end{smallmatrix}\right] R_{r,m} = Y_m R_{r,m}$. It follows from (3.12) that

$$\mathcal{B}Z_m = \mathcal{B}\mathbb{K}_\ell R_{\ell,m}^{-1} = \rho_1 \mathbb{K}_r R_{\ell,m}^{-1} = Y_m(\rho_1 R_{r,m} R_{\ell,m}^{-1}) \equiv Y_m R_m,$$

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

$$(\mathcal{B}^{-1}\mathcal{A})\mathbb{K}[\![\mathcal{B}^{-1}\mathcal{A}, \mathbf{z}_1, m]\!] = \mathbb{K}[\![\mathcal{B}^{-1}\mathcal{A}, \mathbf{z}_1, m]\!]H_0 + (\mathcal{B}^{-1}\mathcal{A})^m \mathbf{z}_1 \mathbf{e}_m^\top, \tag{3.13}$$

where H_0 is the lower shift matrix, i.e., a matrix with ones below the main diagonal and zeros elsewhere. From (3.13) and (3.12) we have

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

where $\mathbf{z}_m = R_{\ell,m}^{-1}(m,m)\mathcal{B}(\mathcal{B}^{-1}\mathcal{A})^m\mathbf{z}_1 \equiv \begin{bmatrix}\mathbf{z}_{m,1}\\ \mathbf{z}_{m,2}\end{bmatrix}$ and $\widetilde{Y}_m^H = [V_m^H \ \mathbf{0}_{m,n}]$. 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 $V_m^H\mathbf{g}_m = V_m^H[(I_n - V_mV_m^H)\mathbf{z}_{m,1}] = \mathbf{0}$.

(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 $Z_m(:,1) = \begin{bmatrix}\mathbf{q}_1\\ \mathbf{p}_1\end{bmatrix}$ with $\|\mathbf{q}_1\|_2 = 1$. From (3.11), we have

$$\mathcal{A}Z_m = \mathcal{B}Z_m(R_m^{-1}H_m) + \boldsymbol{\eta}_m\mathbf{e}_m^\top.$$

Let $Z_m = \widetilde{Z}_mT_m$ be the QR_+-factorization of Z_m. Then we have the standard Arnoldi decomposition

$$\begin{cases} (\mathcal{B}^{-1}\mathcal{A})\widetilde{Z}_m = \widetilde{Z}_m\widetilde{H}_m + \widetilde{\boldsymbol{\eta}}_m\mathbf{e}_m^\top, \\ \widetilde{H}_m = (T_mR_m^{-1}H_m + \widetilde{Z}_m^H\mathcal{B}^{-1}\boldsymbol{\eta}_m\mathbf{e}_m^\top)T_m^{-1}, \\ \widetilde{\boldsymbol{\eta}}_m = (I_m - \widetilde{Z}_m\widetilde{Z}_m^H)\mathcal{B}^{-1}\boldsymbol{\eta}_m\mathbf{e}_m^\top T_m^{-1}. \end{cases} \tag{3.14}$$

Note that the standard Arnoldi decomposition (3.14) is unreduced, it is essentially unique. It follows that Q_m and T_m^{-1} of the QR_+-factorization $\widetilde{Q}_m = Q_mT_m^{-1}$ are unique, and then $P_m = \widetilde{P}_mT_m$ is unique. This concludes the proof. □

Theorem 3.4. *If the mth order SGA decomposition* (3.11) *exists, then*

$$\mathbb{K}_\ell = \mathbb{K}[\![\mathcal{B}^{-1}\mathcal{A}, \mathbf{z}_1, m]\!] = Z_m[\mathbf{e}_1 \quad R_m^{-1}H_m\mathbf{e}_1 \quad \cdots \quad (R_m^{-1}H_m)^{m-1}\mathbf{e}_1], \tag{3.15a}$$

$$\mathbb{K}_r = \mathbb{K}[\![\mathcal{A}\mathcal{B}^{-1}, \mathbf{y}_1, m]\!] = Y_m[\mathbf{e}_1 \quad H_mR_m^{-1}\mathbf{e}_1 \quad \cdots \quad (H_mR_m^{-1})^{m-1}\mathbf{e}_1]. \tag{3.15b}$$

Proof. It suffices to show

$$\begin{aligned}\mathbb{K}_\ell &= [\mathbf{z}_1\ (\mathcal{B}^{-1}\mathcal{A})\mathbf{z}_1\ \cdots\ (\mathcal{B}^{-1}\mathcal{A})^{m-1}\mathbf{z}_1] \\ &= Z_m[\mathbf{e}_1\ R_m^{-1}H_m\mathbf{e}_1\ \cdots\ (R_m^{-1}H_m)^{m-1}\mathbf{e}_1]. \end{aligned} \tag{3.16}$$

Since $Z_m\mathbf{e}_1 = \mathbf{z}_1$, we suppose

$$(\mathcal{B}^{-1}\mathcal{A})^{i-1}\mathbf{z}_1 = Z_m(R_m^{-1}H_m)^{i-1}\mathbf{e}_1, \quad \forall i < m,$$

and prove (3.16) by induction. From (3.11) we have

$$\begin{aligned}(\mathcal{B}^{-1}\mathcal{A})^i\mathbf{z}_1 &= (\mathcal{B}^{-1}\mathcal{A})Z_m(R_m^{-1}H_m)^{i-1}\mathbf{e}_1 \\ &= [Z_m(R_m^{-1}H_m) + \mathcal{B}^{-1}\boldsymbol{\eta}_m\mathbf{e}_m^\top](R_m^{-1}H_m)^{i-1}\mathbf{e}_1 \\ &= Z_m(R_m^{-1}H_m)^i\mathbf{e}_1 + \mathcal{B}^{-1}\boldsymbol{\eta}_m(\mathbf{e}_m^\top(R_m^{-1}H_m)^{i-1}\mathbf{e}_1) \\ &= Z_m(R_m^{-1}H_m)^i\mathbf{e}_1 \end{aligned} \tag{3.17}$$

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 (3.11) follows

$$(\mathcal{A}\mathcal{B}^{-1})Y_m = Y_m(H_mR_m^{-1}) + \widetilde{\boldsymbol{\eta}}_m\mathbf{e}_m^\top, \tag{3.18}$$

where $\widetilde{\boldsymbol{\eta}}_m = R_m(m,m)^{-1}\boldsymbol{\eta}_m$. Similar to (3.17), The equation (3.15b) follows from (3.18) immediately. □

Remark 3.5. *Theorem 3.3 shows that $\mathbb{K}_{\ell,1}$ has the QR_+-factorization, $\mathbb{K}_{\ell.1} = Q_mR_{\ell,m}$, then the SGA decomposition* (3.11) *exists and unique up to $Y_m\mathbf{e}_1 = \mathbf{y}_1$. Theorem 3.4 shows that if the SGA decomposition* (3.11) *exists, then $\mathbb{K}_\ell$ and $\mathbb{K}_r$ have the QR_+-factorizations* (3.15a) *and* (3.15b), *respectively.*

3.2.2 The SGA algorithm

We now derive an algorithm termed as the SGA algorithm for the computation of the SGA decomposition (3.9). Given $\mathbf{q}_1, \mathbf{p}_1 \in \mathbb{C}^n$ with $\|\mathbf{q}\|_1 = 1$, let

$$R_1 = \|K\mathbf{q}_1\|_2 \neq 0, \quad \mathbf{v}_1 = K\mathbf{q}_1/R_1, \quad \mathbf{u}_1 = \mathbf{p}_1/R_1,$$

$$H_1 = \mathbf{v}_1^H(-D\mathbf{q}_1 + \mathbf{p}_1), \quad \mathbf{g}_1 = -D\mathbf{q}_1 + \mathbf{p}_1 - \mathbf{v}_1 H_1 \quad \text{and} \quad \mathbf{f}_1 = -M\mathbf{q}_1 - \mathbf{u}_1 H_1,$$

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 (3.8) with $m = 1$.

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

$$\begin{bmatrix} -D & I_n \\ -M & \mathbf{0} \end{bmatrix} \begin{bmatrix} Q_j \\ P_j \end{bmatrix} = \begin{bmatrix} V_j \\ U_j \end{bmatrix} H_j + \begin{bmatrix} \mathbf{g}_j \\ \mathbf{f}_j \end{bmatrix} \mathbf{e}_j^\top, \tag{3.19a}$$

$$\begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix} \begin{bmatrix} Q_j \\ P_j \end{bmatrix} = \begin{bmatrix} V_j \\ U_j \end{bmatrix} R_j, \tag{3.19b}$$

$$Q_j^H Q_j = I_j, \quad V_j^H V_j = I_j \quad \text{and} \quad V_j^H \mathbf{g}_j = \mathbf{0}. \tag{3.19c}$$

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 $j+1$

$$\begin{bmatrix} -D & I_n \\ -M & \mathbf{0} \end{bmatrix} \begin{bmatrix} Q_j & \mathbf{q} \\ P_j & \mathbf{p} \end{bmatrix} = \begin{bmatrix} V_j & \mathbf{v} \\ U_j & \mathbf{u} \end{bmatrix} \begin{bmatrix} H_j & \mathbf{h} \\ \gamma \mathbf{e}_j^\top & \alpha \end{bmatrix} + \begin{bmatrix} \mathbf{g}_{j+1} \\ \mathbf{f}_{j+1} \end{bmatrix} \mathbf{e}_{j+1}^\top, \tag{3.20a}$$

$$\begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix} \begin{bmatrix} Q_j & \mathbf{q} \\ P_j & \mathbf{p} \end{bmatrix} = \begin{bmatrix} V_j & \mathbf{v} \\ U_j & \mathbf{u} \end{bmatrix} \begin{bmatrix} R_j & \mathbf{r} \\ \mathbf{0} & \rho \end{bmatrix}, \tag{3.20b}$$

$$Q_{j+1}^H Q_{j+1} = I_{j+1}, \quad V_{j+1}^H V_{j+1} = I_{j+1} \quad \text{and} \quad V_{j+1}^H \mathbf{g}_{j+1} = \mathbf{0}, \tag{3.20c}$$

where $Q_{j+1} = [Q_j \ \mathbf{q}]$ and $V_{j+1} = [V_j \ \mathbf{v}]$. Comparing the leading j columns of (3.20a) with (3.19a), we get

$$\gamma = \|\mathbf{g}_j\|_2 \neq 0, \quad \mathbf{v} = \mathbf{g}_j/\gamma \neq \mathbf{0} \quad \text{and} \quad \mathbf{u} = \mathbf{f}_j/\gamma. \tag{3.21}$$

Equating the $(j+1)$st column on both sides of (3.20b) and noting (3.20c), the vector $\mathbf{q}$ must satisfy

$$K\mathbf{q} = V_j\mathbf{r} + \mathbf{v}\rho \quad \text{and} \quad Q_j^H\mathbf{q} = \mathbf{0}. \tag{3.22}$$

Premultiplying (3.22) by $Q_j^H K^{-1}$ and applying the relation $KQ_j = V_jR_j$ 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

$$\mathbf{r} = -R_j Q_j^H K^{-1}\mathbf{v}\rho. \tag{3.23}$$

Substituting (3.23) into (3.22), we have

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

where $\rho \equiv \|(I_j - Q_jQ_j^H)K^{-1}\mathbf{v}\|_2^{-1}$ so that $Q_j^H\mathbf{q} = \mathbf{0}$ and $\|\mathbf{q}\|_2 = 1$. Note that ρ is well-defined, otherwise, $\|(I_j - Q_jQ_j^H)K^{-1}\mathbf{v}\|_2 = 0$ implies $K^{-1}\mathbf{v} \in \operatorname{span}\{Q_j\}$ and hence $\mathbf{v} = KQ_j\mathbf{c} = V_jR_j\mathbf{c}$ for some constant vector $\mathbf{c}$. However, $V_j^H\mathbf{v} = \mathbf{0}$ implies $\mathbf{v} = \mathbf{0}$ which contradicts to the fact (3.21). After determining $\mathbf{u}$, $\mathbf{r}$ and ρ, (3.20b) 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 (3.20a), we know that if we take

$$\begin{bmatrix}\mathbf{h}\\ \alpha\end{bmatrix} = \begin{bmatrix} V_j^H(-D\mathbf{q}+\mathbf{p})\\ \mathbf{v}^H(-D\mathbf{q}+\mathbf{p})\end{bmatrix} \quad\text{and}\quad \begin{bmatrix}\mathbf{g}_{j+1}\\ \mathbf{f}_{j+1}\end{bmatrix} = \begin{bmatrix}-D & I_n\\ -M & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{q}\\ \mathbf{p}\end{bmatrix} - \begin{bmatrix}V_j & \mathbf{v}\\ U_j & \mathbf{u}\end{bmatrix}\begin{bmatrix}\mathbf{h}\\ \alpha\end{bmatrix} \tag{3.24}$$

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 $\mathcal{K}_j(\mathcal{B}^{-1}\mathcal{A}, \mathbf{q}_1)$ and $\mathcal{K}_j(\mathcal{A}\mathcal{B}^{-1}, \mathbf{v}_1)$ 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 effortless to see that $\mathbf{v}$, $\mathbf{u}$ and γ satisfy the jth column of (3.20a) 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 (3.20b) shows that $\mathbf{q} = K^{-1}V_j\mathbf{r} = Q_j(R_j^{-1}\mathbf{r})$ (since $KQ_j = V_jR_j$) and the orthogonality requirement $\{\mathbf{q}_1, \ldots, \mathbf{q}_j, \mathbf{q}\}$ in (3.20c) 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 (3.20b) 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 (3.24) are given by

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

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

$$\begin{bmatrix}-D & I_n\\ -M & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathring{Q}_m\\ \mathring{P}_m\end{bmatrix} = \begin{bmatrix}\mathring{V}_m\\ \mathring{U}_m\end{bmatrix}\mathring{H}_m + \begin{bmatrix}\mathbf{g}_m\\ \mathbf{f}_m\end{bmatrix}\mathbf{e}_m^\top, \tag{3.25a}$$

$$\begin{bmatrix}K & \mathbf{0}\\ \mathbf{0} & I_n\end{bmatrix}\begin{bmatrix}\mathring{Q}_m\\ \mathring{P}_m\end{bmatrix} = \begin{bmatrix}\mathring{V}_m\\ \mathring{U}_m\end{bmatrix}\mathring{R}_m, \tag{3.25b}$$

$$\mathring{Q}_m^H\mathring{Q}_m = J_m, \quad \mathring{V}_m^H\mathring{V}_m = J_m \quad \text{and} \quad \mathring{V}_m^H\mathbf{g}_m = \mathbf{0}, \tag{3.25c}$$

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 3.6 ([3], 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* $\mathbf{y} = \left[\begin{smallmatrix}\mathbf{0}\\ \mathbf{u}\end{smallmatrix}\right] \in \operatorname{span}\{\mathbf{y}_1, \ldots, \mathbf{y}_m\}$ *if and only if* $\mathbf{u} \in \operatorname{span}\{\mathbf{u}_{i_{j+1}}, \ldots, \mathbf{u}_{i_m}\}$.

The pseudocode for the SGA algorithm that iteratively generates an mth order (deflated) SGA decomposition is listed in Algorithm 3.1.

Algorithm 3.1 The 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: $Q_m$, $V_m$, $U_m$, $\mathbf{g}_m := \mathbf{g}$, $\mathbf{f}_m := \mathbf{f}$, $\mathtt{M}_m$, $\mathbb{D}_m$, 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 (3.8) of order
    $m$.
 1: $Q_1 := \mathbf{q}_1$;  $R_1 := \|K\mathbf{q}_1\|_2$;  $V_1 := K\mathbf{q}_1/R_1$;  $U_1 := \mathbf{p}_1/R_1$;  $\mathtt{M}_1 := M\mathbf{q}_1$;
    $\mathbb{D}_1 := D\mathbf{q}_1$;
 2: $\mathbf{g} := -\mathbb{D}_1 + \mathbf{p}_1$;  $H_1 := V_1^H\mathbf{g}$;  $\mathbf{g} := \mathbf{g} - V_1H_1$;  $\mathbf{f} := -\mathtt{M}_1 - U_1H_1$;
 3: for $j = 1, 2, \ldots, m-1$ do
 4:   if $\mathbf{g} \neq \mathbf{0}$ then
 5:     $\gamma := \|\mathbf{g}\|_2$;  $\mathbf{v} := \mathbf{g}/\gamma$;  $\mathbf{u} := \mathbf{f}/\gamma$;  $V_{j+1} := [V_j\ \mathbf{v}]$;  $U_{j+1} := [U_j\ \mathbf{u}]$;
        $H_j := \begin{bmatrix} H_j \\ \gamma\mathbf{e}_j^\top \end{bmatrix}$;
 6:     Solve $K\mathbf{q} = \mathbf{v}_{j+1}$ for $\mathbf{q}$
 7:     $\mathbf{r} := Q_j^H\mathbf{q}$;  $\mathbf{q} := \mathbf{q} - Q_j\mathbf{r}$;  $\rho := \|\mathbf{q}\|_2^{-1}$;  $\mathbf{q} := \mathbf{q}\rho$;  $\boldsymbol{\mu} := M\mathbf{q}$;  $\boldsymbol{\delta} := D\mathbf{q}$;
 8:     $Q_{j+1} := [Q_j\ \mathbf{q}]$;  $\mathtt{M}_{j+1} := [\mathtt{M}_j\ \boldsymbol{\mu}]$;  $\mathbb{D}_{j+1} := [\mathbb{D}_j\ \boldsymbol{\delta}]$;  $R_{j+1} := \begin{bmatrix} R_j & \mathbf{r} \\ \mathbf{0} & \rho \end{bmatrix}$;
 9:     $\mathbf{g} := -\boldsymbol{\delta} + U_{j+1}R_{j+1}(:, j+1)$;  $\mathbf{h} := V_{j+1}^H\mathbf{g}$;  $H_{j+1} := [H_j\ \mathbf{h}]$;
10:     $\mathbf{g} := \mathbf{g} - V_{j+1}\mathbf{h}$;  $\mathbf{f} := -\boldsymbol{\mu} - U_{j+1}\mathbf{h}$;
11:   else
12:     if $\mathbf{f} \in \operatorname{span}\{\mathbf{u}_i \mid i : \mathbf{v}_i = \mathbf{0},\ 1 \leq i \leq j\}$ then
13:       breakdown
14:     else
15:       $V_{j+1} := [V_j\ \mathbf{0}]$;  $U_{j+1} := [U_j\ \mathbf{f}]$;  $Q_{j+1} := [Q_j\ \mathbf{0}]$;
          $\mathtt{M}_{j+1} := [\mathtt{M}_j\ \mathbf{0}]$;  $\mathbb{D}_{j+1} := [\mathbb{D}_j\ \mathbf{0}]$;
16:       $\mathbf{h} := V_j^H\mathbf{f}$;  $H_{j+1} := \begin{bmatrix} H_j & \mathbf{h} \\ \mathbf{e}^\top & 0 \end{bmatrix}$;  $R_{j+1} := \begin{bmatrix} R_j & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$;
          $\mathbf{g} := \mathbf{f} - V_j\mathbf{h}$;  $\mathbf{f} := -U_j\mathbf{h}$;
17:     end if
18:   end if
19: end for
```

Remark 3.7. *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*

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

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* (3.8b)*, 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 [3].*

3.3 The SGA Method for Solving Quadratic Eigenvalue Problems

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 (3.25).

3.3.1 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

$$\boldsymbol{\omega}^*(\theta^2 M + \theta D + K)\boldsymbol{v} = 0 \quad \text{for all } \boldsymbol{\omega} \in \mathcal{Q}_m, \tag{3.26}$$

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 (3.26) implies that θ and $\boldsymbol{\xi}$ must satisfy the reduced QEP:

$$(\theta^2 M_m + \theta D_m + K_m)\boldsymbol{\xi} = \mathbf{0}, \tag{3.27}$$

where

$$M_m = Q_m^* M Q_m, \quad D_m = Q_m^* D Q_m, \quad K_m = Q_m^* K Q_m. \tag{3.28}$$

The eigenpair $(\theta, \boldsymbol{\xi})$ of the small-scale QEP (3.27) defines a Ritz pair $(\theta, Q_m\boldsymbol{\xi})$ of the QEP (3.1) 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 [36]. 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 (3.28). As we describe in Remark 3.7(ii), the resulting matrices

$$\mathbb{M}_m := MQ_m \quad \text{and} \quad \mathbb{D}_m := DQ_m$$

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 (3.8b) 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 $2n^2m$ flops. Therefore, the small-scale matrices M_m and D_m can be respectively generated by

$$M_m = Q_m^* \mathbb{M}_m, \quad D_m = Q_m^* \mathbb{D}_m \quad \text{and} \quad K_m = Q_m^* V_m R_m. \tag{3.29}$$

Totally, (3.28) needs about $6n^2m + 6nm^2$ flops to generate the coefficient matrices of the projected QEP (3.27), however, the matrix products (3.29) only need $8nm^2$ flops. Also note that if we consider the first companion form linearization of the QEP (3.1), there is no such an advantage. That is, (3.28) is the only way to generate the coefficient matrices of the reduced QEP (3.27).

3.3.2 The projection subspace

In this subsection we explain the motivation of choosing the projection subspace $\mathcal{Q}_m \equiv \text{span}\{Q_m\}$ where Q_m is generated from the SGA algorithm. We first recall a

Algorithm 3.2 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: $k$ Ritz pairs and their relative residuals.
1: Run the SGA algorithm (Algorithm 3.1) to generate an $m$th order SGA decomposition (3.8).
2: Compute $M_m$, $D_m$ and $K_m$ via (3.29).
3: Solve the reduced QEP (3.27) 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 (3.1) by the relative norms of residual vectors:
   $$\frac{\|(\theta_i^2 M + \theta_i D + K)\boldsymbol{v}_i\|_2}{|\theta_i|^2\|M\|_F + |\theta_i|\|D\|_F + \|K\|_F}, \quad i = 1, \ldots, k. \qquad (3.30)$$
```

lemma in the SOAR method [3].

Lemma 3.8 ([3], Lemma 2.2). *Let A 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*

$$AW_m = W_{m+1}\underline{H}_m$$

for an $(m+1) \times m$ upper Hessenberg matrix $\underline{H}_m$. Then there is an upper triangular matrix T_m such that

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

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

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

Next, we consider a Krylov subspace associated with the linearized eigenvalue problem (3.3) and show that it is embedded into a larger subspace spanned by some column vectors in the SGA decomposition (3.8).

Theorem 3.9. *Consider the SGA decomposition* (3.8) *of order m. Let*

$$\widehat{Q}_{\widehat{m}} = \begin{array}{c c} & \begin{array}{ccc} m & m & 1 \end{array} \\ \begin{array}{c} n \\ n \end{array} & \begin{bmatrix} Q_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -MQ_m & \mathbf{p}_1 \end{bmatrix} \end{array} \in \mathbb{C}^{2n\times(2m+1)}. \tag{3.31}$$

Then, for $\mathcal{A}$ and $\mathcal{B}$ defined in (3.9), *we have $\mathcal{K}_m(\mathcal{B}^{-1}\mathcal{A}, \left[\begin{smallmatrix}\mathbf{q}_1\\ \mathbf{p}_1\end{smallmatrix}\right]) \subseteq \operatorname{span}\{\widehat{Q}_{\widehat{m}}\}$.*

Proof. From (3.11b), we have

$$\begin{bmatrix} V_m \\ U_m \end{bmatrix} = \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix} \begin{bmatrix} Q_m \\ P_m \end{bmatrix} R_m^{-1}.$$

Substituting it into the equation (3.11a) and then premultiplying it by $\left[\begin{smallmatrix} K^{-1} & \mathbf{0} \\ \mathbf{0} & I_n \end{smallmatrix}\right]$, we get

$$\begin{bmatrix} -K^{-1}D & K^{-1} \\ -M & \mathbf{0} \end{bmatrix} \begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} Q_m & \mathbf{q}_m^{\ell} \\ P_m & \mathbf{p}_m^{\ell} \end{bmatrix} \begin{bmatrix} H_m^{\ell} \\ \mathbf{e}_m^{\top} \end{bmatrix}, \tag{3.32}$$

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 (3.32) and Lemma 3.8, we know that

$$\mathcal{K}_m\left(\mathcal{B}^{-1}\mathcal{A}, \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{p}_1 \end{bmatrix}\right) \equiv \mathcal{K}_m\left(\begin{bmatrix} -K^{-1}D & K^{-1} \\ -M & \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{p}_1 \end{bmatrix}\right) = \operatorname{span}\left\{\begin{bmatrix} Q_m \\ P_m \end{bmatrix}\right\} \tag{3.33}$$

and the set $\{\left[\begin{smallmatrix}\mathbf{q}_1\\ \mathbf{p}_1\end{smallmatrix}\right], \ldots, \left[\begin{smallmatrix}\mathbf{q}_m\\ \mathbf{p}_m\end{smallmatrix}\right]\}$ is a non-orthonormal basis of the above Krylov subspace (3.33). Next, we show that

$$\begin{bmatrix} \mathbf{q}_i \\ \mathbf{p}_i \end{bmatrix} \in \operatorname{span}\{\widehat{Q}_{\widehat{m}}\} \quad \text{for} \quad i = 1, \ldots, m, \tag{3.34}$$

and the conclusion of Theorem 3.9 follows directly from (3.33) and (3.34).

To prove (3.34), it suffices to show that $\mathbf{p}_i \in \operatorname{span}\{-MQ_m, \mathbf{p}_1\}$, $1 \leq i \leq m$. 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 $1 < i \leq m-1$. From the equality (3.32), we have $-MQ_m = P_m H_m^\ell + \mathbf{p}_m^\ell \mathbf{e}_m^\top$. Thus,

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

and it follows that

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

We complete the proof. □

Instead of using the Krylov subspace $\mathcal{K}_m(\mathcal{B}^{-1}\mathcal{A}, \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{p}_1 \end{bmatrix})$, we choose the larger subspace $\operatorname{span}\{\widehat{Q}_{\widehat{m}}\}$ to extract approximations of eigenpairs. To project the coefficient matrices of the GEP (3.5) onto the subspace $\operatorname{span}\{\widehat{Q}_{\widehat{m}}\}$, we get

$$\widehat{Q}_{\widehat{m}}^* \begin{bmatrix} -D & I_n \\ -M & \mathbf{0} \end{bmatrix} \widehat{Q}_{\widehat{m}} = \begin{array}{c|cc|c} & m & m & 1 \\ \hline m & -D_m & -M_m & Q_m^* \mathbf{p}_1 \\ m & N_m & \mathbf{0} & \mathbf{0} \\ \hline 1 & -\mathbf{p}_1^* M Q_m & \mathbf{0} & 0 \end{array} \equiv \widehat{\mathcal{A}}, \tag{3.35a}$$

$$\widehat{Q}_{\widehat{m}}^* \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & I_n \end{bmatrix} \widehat{Q}_{\widehat{m}} = \begin{array}{c|cc|c} & m & m & 1 \\ \hline m & K_m & \mathbf{0} & \mathbf{0} \\ m & \mathbf{0} & N_m & -Q_m^* M^* \mathbf{p}_1 \\ \hline 1 & \mathbf{0} & -\mathbf{p}_1^* M Q_m & \mathbf{p}_1^* \mathbf{p}_1 \end{array} \equiv \widehat{\mathcal{B}}, \tag{3.35b}$$

where M_m, D_m, K_m are defined in (3.28) and $N_m = Q_m^* M^* M Q_m$. Therefore, the

GEP (3.5) is reduced to the problem

$$\widehat{\mathcal{A}}\mathbf{s} = \nu\widehat{\mathcal{B}}\mathbf{s} \tag{3.36}$$

with $\widehat{\mathcal{A}}$ and $\widehat{\mathcal{B}}$ defined in (3.35). Observe that if we premultiply (3.36) 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 resulting GEP $(L\widehat{\mathcal{A}})\mathbf{s} = \mu(L\widehat{\mathcal{B}})\mathbf{s}$ are respectively of the forms

$$L\widehat{\mathcal{A}} \equiv \left[\begin{array}{cc|c} -D_m & -M_m & Q_m^*\mathbf{p}_1 \\ N_m & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & 0 \end{array}\right], \quad L\widehat{\mathcal{B}} \equiv \left[\begin{array}{cc|c} K_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & N_m & -Q_m^* M^* \mathbf{p}_1 \\ \hline \mathbf{0} & \mathbf{0} & c \end{array}\right], \tag{3.37}$$

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 (3.37) 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 (3.37) to consider the leading $2m \times 2m$ submatrices, which is just the first companion form linearization [21] of the reduced QEP (3.27), for solving QEPs.

3.4 Refined SGA Method

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

$$\boldsymbol{v}_\theta^+ \equiv \underset{\boldsymbol{v}\in\mathcal{Q}_m,\ \|\boldsymbol{v}\|_2=1}{\arg\min} \|(\theta^2 M + \theta D + K)\boldsymbol{v}\|_2. \tag{3.38}$$

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 [34].

Let $(\theta, \boldsymbol{\xi}_\theta)$ be an eigenpair obtained from the small-scale QEP (3.27), then $(\theta, \boldsymbol{v}_\theta) = (\theta, Q_m\boldsymbol{\xi}_\theta)$ is a Ritz pair the QEP (3.1). To solve the optimization problem (3.38), we find that

$$\begin{aligned}
&(\theta^2 M + \theta D + K)Q_m \\
&= \theta^2(-U_m H_m - \mathbf{f}_m\mathbf{e}_m^\top) + \theta(P_m - V_m H_m - \mathbf{g}_m\mathbf{e}_m^\top) + V_m R_m \\
&= V_m(-\theta H_m + R_m) + \mathbf{g}_m(-\theta\mathbf{e}_m^\top) + U_m(-\theta^2 H_m + \theta R_m) + \mathbf{f}_m(-\theta^2\mathbf{e}_m^\top) \\
&= \begin{bmatrix} V_m & \mathbf{g}_m & U_m & \mathbf{f}_m \end{bmatrix} \begin{bmatrix} -\theta H_m + R_m \\ -\theta\mathbf{e}_m^\top \\ -\theta^2 H_m + \theta R_m \\ -\theta^2\mathbf{e}_m^\top \end{bmatrix},
\end{aligned} \tag{3.39}$$

where we use the SGA decomposition (3.8) in the first two equalities. Since V_m is a column orthonormal matrix, the QR factorization of $[V_m \ \ \mathbf{g}_m \ \ U_m \ \ \mathbf{f}_m]$ is of the form

$$[V_m \;\; \mathbf{g}_m \;\; U_m \;\; \mathbf{f}_m] = \begin{bmatrix} V_m & \widetilde{\mathbf{g}}_m & \widetilde{U}_m & \widetilde{\mathbf{f}}_m \end{bmatrix} \begin{bmatrix} I_m & \mathbf{t}_{12} & T_{13} & \mathbf{t}_{14} \\ & t_{22} & \mathbf{t}_{23} & t_{24} \\ & & T_{33} & \mathbf{t}_{34} \\ & & & t_{44} \end{bmatrix}, \tag{3.40}$$

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, (3.39) and (3.40) imply

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

where

$$S(m,\theta) \equiv \begin{bmatrix} I_m & \mathbf{t}_{12} & T_{13} & \mathbf{t}_{14} \\ & t_{22} & \mathbf{t}_{23} & t_{24} \\ & & T_{33} & \mathbf{t}_{34} \\ & & & t_{44} \end{bmatrix} \begin{bmatrix} -\theta H_m + R_m \\ -\theta \mathbf{e}_m^\top \\ -\theta^2 H_m + \theta R_m \\ -\theta^2 \mathbf{e}_m^\top \end{bmatrix} \in \mathbb{C}^{(2m+2)\times m}. \tag{3.41}$$

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 (3.38) with minimum $s_{\theta,\min}$. In summary, we have the following theorem.

Theorem 3.10. *Let* $(\theta, Q_m\boldsymbol{\xi}_\theta)$ *be a Ritz pair the QEP* (3.1) *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* (3.41) *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* (3.38) *with minimum* $s_{\theta,\min}$.

When applying the refinement strategy for several Ritz pairs, we compute the QR factorization (3.40) 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 3.3.

Algorithm 3.3 The 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: $k$ refined Ritz pairs and their relative residuals.
Run steps 1–3 of the SGA method to obtain $k$ wanted Ritz pairs $(\theta_i, Q_m\boldsymbol{\xi}_i)$, $i = 1, \ldots, k$.
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 (3.40).
for $i = 1, \ldots, k$ do
    Calculate the matrix $S(m, \theta_i)$ as defined in (3.41).
    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
        $\frac{\|(\theta_i^2 M + \theta_i D + K)\boldsymbol{v}_i^+\|_2}{|\theta_i|^2\|M\|_F + |\theta_i|\|D\|_F + \|K\|_F} = \frac{s_{\theta_i,\min}}{|\theta_i|^2\|M\|_F + |\theta_i|\|D\|_F + \|K\|_F}.$   (3.42)
end for
```

3.5 Implicit Restarting of the SGA Method

Similar to the standard implicitly restarted Arnoldi (IRA) method [59] 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 [60] to implicitly restart the SGA/RSGA method, namely IRSGA/IRRSGA.

3.5.1 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 [60]. For details, see [60, 62].

Suppose we have computed the mth order SGA decomposition (3.11). For given shifts $\vartheta_1, \ldots, \vartheta_p$, $p = m - k$, 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, $i = 1, \ldots, p$. Write $E^+ = E_1 \cdots E_p$ and $F^+ = F_1 \cdots F_p$. Note that F_i is upper Hessenberg, $i = 1, \ldots, p$.

Let

$$\begin{aligned} H_m^+ &\equiv (E^+)^H H_m F^+, \\ R_m^+ &\equiv (E^+)^H R_m F^+, \\ Z_m^+ &\equiv Z_m F^+, \\ Y_m^+ &\equiv Y_m E^+. \end{aligned}$$

Then H_m^+ and R_m^+ are again upper Hessenberg and upper triangular, respectively. Set

$$Q_m^+ = Q_m F^+ \quad \text{and} \quad V_m^+ = V_m E^+,$$

we then have $(Q_m^+)^H Q_m^+ = (V_m^+)^H V_m^+ = I_m$. Postmultiplying (3.11a) and (3.11b) by F^+, we get

$$\mathcal{A} Z_m^+ = Y_m^+ H_m^+ + \boldsymbol{\eta}_m \mathbf{e}_m^\top F^+, \tag{3.43a}$$

$$\mathcal{B} Z_m^+ = Y_m^+ R_k^+. \tag{3.43b}$$

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 (3.43a) and (3.43b), 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:

$$\mathcal{A}Z_k^+ = Y_k^+ H_k^+ + \boldsymbol{\eta}_k^+ \mathbf{e}_k^\top, \tag{3.44a}$$

$$\mathcal{B}Z_k^+ = Y_k^+ R_k^+, \tag{3.44b}$$

$$(Q_k^+)^H Q_k^+ = (V_k^+)^H V_k^+ = I_k, \;\; (V_k^+)^H \mathbf{g}_k^+ = \mathbf{0}. \tag{3.44c}$$

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

Remark 3.11. *Note that applying an implicitly restarted process on a deflated SGA decomposition* (3.25) *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 resulting 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 resulting 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 resulting 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.

Algorithm 3.4 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 $i = 1, 2, \dots$ do
  Run the SGA algorithm (Algorithm 3.1) to generate an $m$th order SGA decomposition.
  Run the SGA method (Algorithm 3.2) or the RSGA method (Algorithm 3.3) to compute $k$ candidates of Ritz pairs and check their convergence by (3.30) or (3.42).
  if $\#$(convergent Ritz pairs) $\geq k$ then
    break
  else
    Select $p := m - k$ shifts $\vartheta_1, \dots, \vartheta_p$.
    Let $\boldsymbol{\varepsilon} := \mathbf{e}_m^\top$ and $\boldsymbol{\eta} := \boldsymbol{\eta}_m$
    for $i = 1, \dots, p$ 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
        $H_m := E_i^H H_m F_i, \quad R_m := E_i^H R_m F_i,$
        $Z_m := Z_m F_i, \quad Y_m := Y_m E_i \quad \text{and} \quad \boldsymbol{\varepsilon} := \boldsymbol{\varepsilon} F_i.$
    end for
    Set $\boldsymbol{\eta}_k := H_m(k+1,k) Y_m(:,k+1) + \boldsymbol{\varepsilon}(k+1)\boldsymbol{\eta}$
    Set
      $Z_k := Z_m(:,1:k), \quad Y_k := Y_m(:,1:k),$
      $H_k := H_m(1:k,1:k), \quad R_k := R_m(1:k,1:k).$
  end if
end for
```

3.5.2 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 [59] is to choose unwanted Ritz values, and are called exact shifts in [59]. When we solve the reduced QEP (3.27) 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 (3.1) 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 3.4. 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 (3.1), 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 [25]

$$a_2\theta^2 + a_1\theta + a_0 = 0, \tag{3.45}$$

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 $\boldsymbol{\omega}^+$, (3.45) 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 (3.45) would be reduced as follows

$$a_2 = (\boldsymbol{\zeta}^+)^* M_m \boldsymbol{\zeta}^+, \quad a_1 = (\boldsymbol{\zeta}^+)^* D_m \boldsymbol{\zeta}^+ \quad \text{and} \quad a_0 = (\boldsymbol{\zeta}^+)^* K_m \boldsymbol{\zeta}^+, \tag{3.46}$$

where M_m, D_m and K_m is the projections of M, D and K onto the subspace span$\{Q_m\}$ respectively as described in (3.28).

Hence, if ϑ_1^+ and ϑ_2^+ are roots of the quadratic equation (3.45) with coefficients defined in (3.46) then their reciprocal values would be better candidates for the restarting process. Consequently, if $(\vartheta_1, Q_m\boldsymbol{\zeta}_1), \ldots, (\vartheta_p, Q_m\boldsymbol{\zeta}_p)$ 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 (3.45) 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.

3.6 Numerical Results

The purpose of this section is to present a few numerical experiments to validate that the IRRSGA method is viable for solving the QEP (3.1). In addition, the examples demonstrate the superior properties of the IRSGA method and the IRRSGA method than the two versions of the IRA method [59] for solving the QEP where one IRA method is applied to the ℓ-SEP (3.6) and the other is applied to the r-SEP (3.7), 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.

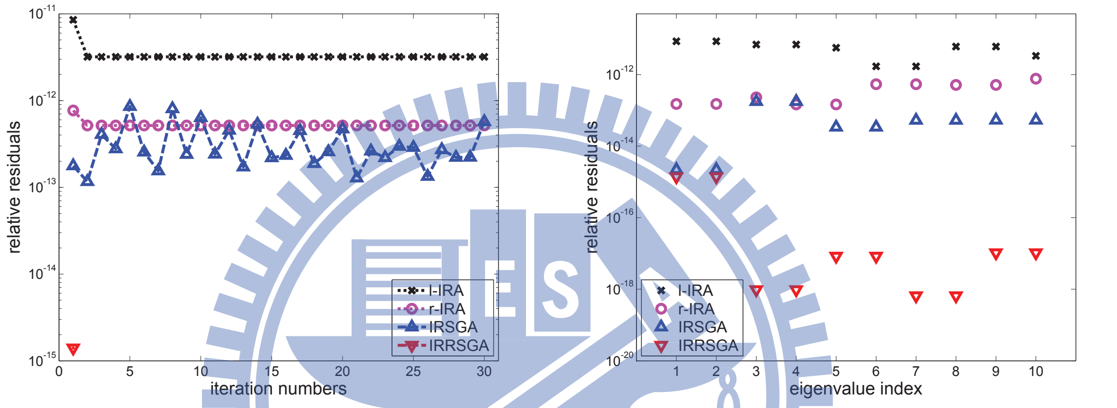
In our examples, the number m denotes the order of the SGA/Arnoldi decomposition, k denotes the number of desired eigenpairs. The starting vector of the SGA method and the standard Arnoldi method are chosen as a vector with all components equal to 1 and the stopping tolerance for relative residuals is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

Example 3.1. This example is obtained from "NLEVP: a collection of nonlinear eigenvalue problem" [9], 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 3.1(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 3.1(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 3.1. In summary, among the four methods the IRRSGA method is the only viable approach that accurately finds

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

Table 3.1: Iteration numbers and CPU time in Example 3.1.



(a) Evolution of relative residuals

(b) Relative residuals of the 1 step iteration

Figure 3.1: Convergence histories for methods ℓ-IRA, r-IRA, IRSGA and IRRSGA in Example 3.1.

desired eigenpairs within 30 iterations.

Example 3.2. In Example 3.1, 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 3.5.2 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 [6]. 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 2.1(i) gives an example of such a setup, where the top boundary is absorbing and the remaining boundary is rigid. The equations characterizing the wave motion in Ω are

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

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.

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

Table 3.2: Iteration numbers and CPU time in Example 3.2.

In this example, we adopt the geometry illustrated in Figure 2.1(i) and physical data used in Example 2.1. The rectangular domain is uniformly partitioned into

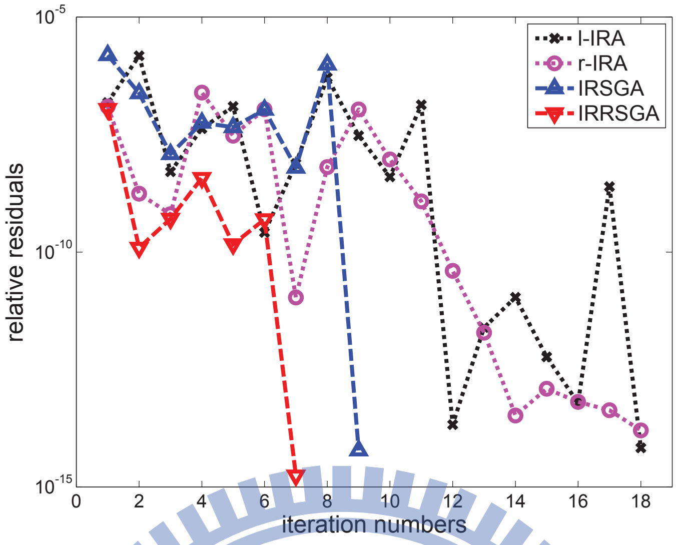


Figure 3.2: Convergence histories for methods ℓ-IRA, r-IRA, IRSGA and IRRSGA in Example 3.2.

384 by 288 rectangles and each rectangle is further refined into two triangles. The dimension of coefficient matrices in this problem is $n = 331,488$. We compute ten analytic solutions of the desired eigenvalues $\lambda_1, \ldots, \lambda_{10}$ plotted in Figure 2.2 with the lowest positive vibration frequencies satisfying $0 < \frac{\text{Im}(\lambda_i)}{2\pi} < 600$Hz. The order m is set to be $m = 20$. The shift target is taken by $\sigma = -25 + 600\pi\mathtt{i}$.

Table 3.2 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.

Example 3.3. This experiment consists of six benchmark examples from the NLEVP [9]. 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 3.3 depicts the maximum of the k residual norms versus restarts and show the convergence processes of each example. Correspondingly, Table 3.3 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 $-\Delta p - (2\pi f/c)^2 p = 0$ [11]. 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

$$\begin{aligned} M &= -4\pi^2 \tfrac{1}{n}(I_n - \tfrac{1}{2}\mathbf{e}_n\mathbf{e}_n^\top), \\ D &= 2\pi \mathtt{i} \tfrac{1}{\zeta}\mathbf{e}_n\mathbf{e}_n^\top, \\ K &= n\left(\text{tridiag}(-1, 2, -1) - \mathbf{e}_n\mathbf{e}_n^\top\right), \end{aligned}$$

where tridiag$(-1, 2, -1)$ is a tridiagonal matrix with -2 on the main diagonal and -1 above and below it. 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 [11] on $[0,1]\times[0,1]$. The coefficient matrices (M,D,K) are given by

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

where h denotes the mesh size, $q=1/h$, ζ is the impedance (possibly complex), $D_q = \mathrm{tridiag}(-1,4,-1) - 2\mathbf{e}_q\mathbf{e}_q^\top \in \mathbb{R}^{q\times q}$ 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 [16] and induces the QEP

$$(\lambda^2M+\lambda D+(1+\mu\mathtt{i})K)\mathbf{x}=\mathbf{0},$$

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 [23]. 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 [74]. Here M, D, K are $n \times n$ matrices defined by

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

where $d_{ij} = \frac{4ij}{i^2-j^2}\upsilon$ 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 [74]

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

where M, D and K are the same as in Wiresaw1 and η is a real nonnegative damping parameter. We use `nlevp('wiresaw2',10000,0.01,0.5)` with $\eta = 0.5$ 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 $\sigma = -0.5$) with $m = 20$.

3.7 Summary

We propose the SGA method, an orthogonal projection approach, for solving QEPs and deduce several variations:

(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

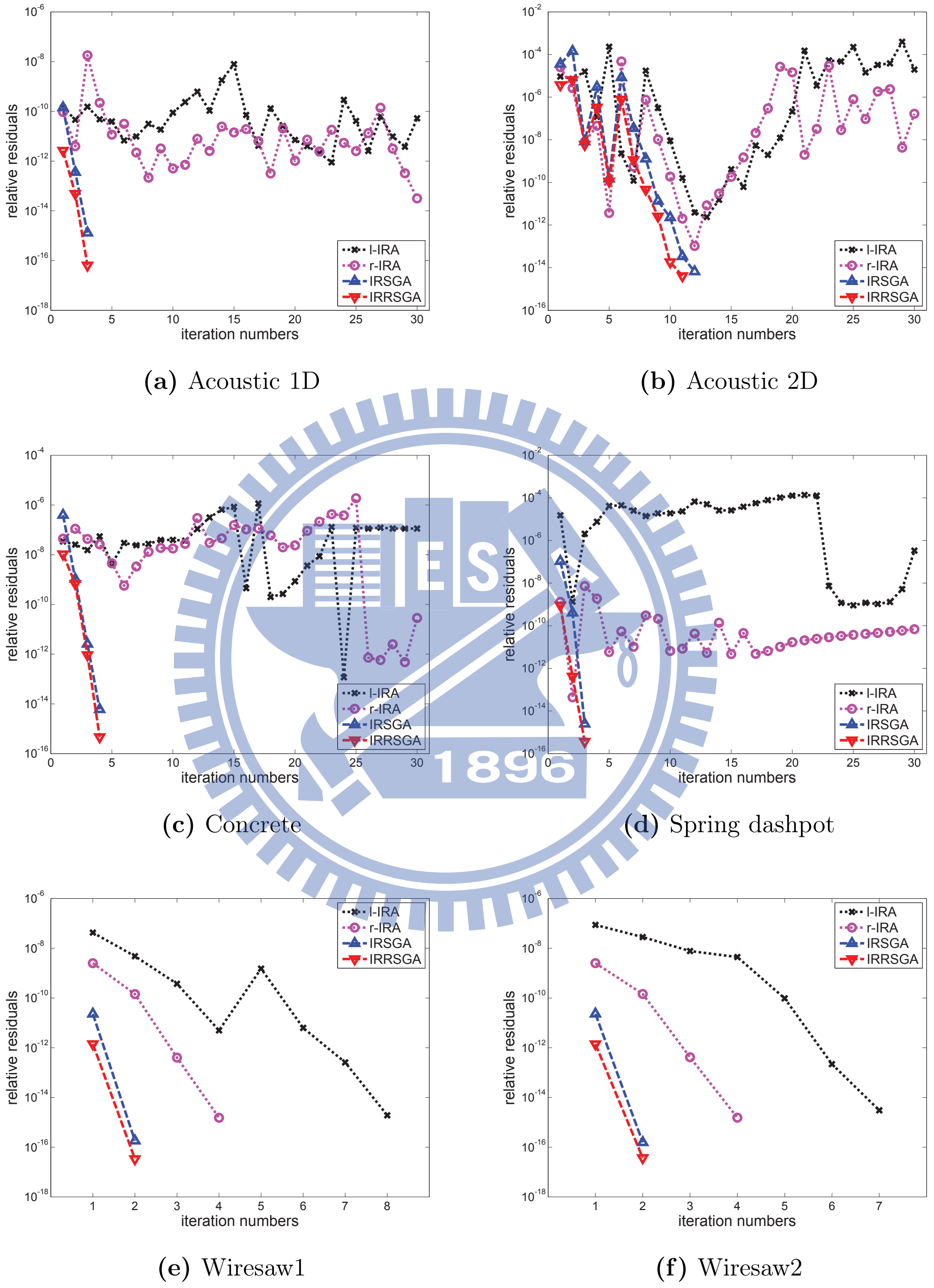
(e) Wiresaw1

	#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 3.3: Iteration numbers and CPU time in Example 3.3.



(a) Acoustic 1D

(b) Acoustic 2D

(c) Concrete

(d) Spring dashpot

(e) Wiresaw1

(f) Wiresaw2

Figure 3.3: Convergence histories for methods ℓ-IRA, r-IRA, IRSGA and IRRSGA in Example 3.3.

- **RSGA** : A refinable version of the SGA method.
- **IRSGA** : The SGA method combining the implicit restart technique.
- **IRRSGA**: A restartable and refinable variation of the SGA method.

The numerical results on computation of the approximate eigenpairs with small eigenvalues in modulus show that, compared to the standard IRA method, both IRSGA and IRRSGA are superior in accuracy as well as convergence rate. Moreover, the IRRSGA method may significantly improve the accuracy for obtaining the desired eigenpairs.

4

Conclusions and Future Work

In this thesis, we consider two themes related to Arnoldi-type approaches for solving nonlinear eigenvalue problems.

In the first topic (Chapter 2), we propose efficient Arnoldi-type methods for computing damped vibration modes of an acoustic fluid confined in a cavity, with absorbing walls capable of dissipating acoustic energy. Two approximations are investigated. One constructed from the edge-based displacement space, which results in QEPs (2.13) and one from the node-based pressure space, which results in REPs (2.20). Our numerical results show that both nodal and edge-based finite elements have second-order convergence rate. We theoretically prove that the nullity of the QEP (2.13) equals the number of the interior grid points. These numerical results show that if the shift value is close to zero, then such a large null space interfere with the convergence of the eigensolver. Furthermore, the numerical evidences also show that (i) there are no spurious eigenmodes for the discretization in terms of pressure nodal finite elements and (ii) the CPU times for solving the corresponding REP (2.20) are only 1/5 to 1/3 of the CPU times for solving the QEP (2.13). For solving the nonlinear eigenvalue problems (2.13) and (2.20), a linearization and a trimmed-linearization method are used to linearize QEP (2.13) and REP (2.20) into four different types of SEPs which can be solved by **Q1** and **Q2** as well as **R1** and **R2**. Numerical accuracy shows that **Q2** and **R2** algorithms are better than **Q1** and **R1** respectively.

In Chapter 3, to deal with QEPs, we presented an orthogonal projection method (named the SGA method) based on a SGA decomposition. We have developed a practical algorithm to compute the SGA decomposition. The application of the SGA decomposition is three aspects. First of all, we compute an orthonormal basis of the projection subspace in the SGA decomposition. Secondly, the SGA decomposition (3.8) has computational advantage for generating the coefficient matrices of reduced QEP (3.28). Finally, 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 had significantly improved the accuracy of computing the desired eigenpairs when the standard IRA method and the IRSGA method cannot converge in a certain number of iterations.

Based on this research, the forthcoming work is to generalize the SGA method and its variations to provide orthogonal projection methods for solving the PEP (1.9) as well as the REP (1.12), respectively. Even though these PEPs/REPs can be solved by nonlinear eigensolvers, these approaches restricts the advantages of the underlying structure and property of PEPs/REPs. Therefore, the generalization of the SGA method may provide an alternative structure-preserved approach for solving PEPs/REPs. Moreover, how to efficiently compute refined eigenpairs using the partial-orthogonal Arnoldi-like decomposition and to appropriately select refined shift for implicitly restarting process will be challenging problems.

Bibliography

[1] Arnoldi WE. The principle of minimized iteration in the solution of matrix eigenvalue problem. *Quarterly of Applied Mathematics* 1951; **9**:17–29

[2] Bai Z, Demmel J, Dongarra J, Ruhe A, van der Vorst HA. Templates for the Solution of Algebraic Eigenvalue Problems: a Practical Guide. SIAM: Philadelphia, PA, 2000.

[3] 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.

[4] Bathe KJ, Nitikitpaiboon C, Wang X. A mixed displacement-based finite element formulation for acoustic fluid-structure interaction, *Computers & Structures* 1995; **56**(2/3):225–237.

[5] Bermúdez A, Durán R, Muschietti MA, Rodríguez R, Solomin J. Finite element vibration analysis of fluid-solid systems without spurious modes, *SIAM Journal on Numerical Analysis* 1995; **32**(4):1280–1295.

[6] Bermúdez A, Durán RG, Rodríguez R, Solomin J. Finite element analysis of a quadratic eigenvalue problem arising in dissipative acoustic, *SIAM Journal on Numerical Analysis* 2000; **38**(1):267–291.

[7] Bermúdez Am Rodríguez R. Finite element computation of the vibration modes of a fluid-solid system, *Computer Methods in Applied Mechanics and Engineering* 1994; **119**(3-4):355–370.

[8] Bermúdez A, Rodríguez R. Modeling and numerical solution of elastoacoustic vibrations with interface damping, *International Journal for Numerical Methods in Engineering* 1999; **46**(10):1763–1779.

[9] 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.

[10] Brezzi F, Fortin M. Mixed and Hybrid Finite Element Methods, *Springer-Verlag*, New York, 1991.

[11] 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**(6):487–512.

[12] Chen HC, Taylor RL. Vibration analysis of fluid-solid systems using a finite element displacement formulation, *International Journal for Numerical Methods in Engineering* 1990; **29**:683–698.

[13] Chou SH, Tang S. Conservative $P1$ conforming and nonconforming Galerkin FEMs: effective flux evaluation via a nonmixed method approach, *SIAM Journal on Numerical Analysis* 2000; **38**(2):660–680.

[14] Datta BN. Numerical Linear Algebra and Applications. 2nd ed. *SIAM*, Philadelphia, PA, 2010.

[15] Fan HY, Lin WW, Van Dooren P. Normwise scaling of second order polynomial matrices. *SIAM Journal on Matrix Analysis and Applications* 2004; **26**(1):252–256.

[16] 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**(13-14):1719–1739.

[17] Flaschka U, Lin WW, Wu JL. A KQZ algorithm for solving linear-response eigenvalue equations. *Linear Algebra and its Applications* 1992; **165**:93–123.

[18] Francis JGF. The QR transformation: A unitary analogue to the LR transformation–Parts 1. *The Computer Journal* 1961; **4**(3):265–271.

[19] Francis JGF. The QR transformation–Parts 2. *The Computer Journal* 1962; **4**(4):332–345.

[20] Gastaldi L. Mixed finite element methods in fluid structure systems, *Numerische Mathematik* 1996; **74**(2):153–176.

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

[22] Golub GH, van Loan CF. Matrix Computation. 3rd ed., The Johns Hopkins University Press, 1996.

[23] Gotts A. Report regarding model reduction, model compaction research project. Manuscript, University of Nottingham, February 2005.

[24] Hamdi M, Ouset Y, Verchery G. A displacement method for the analysis of vibrations of coupled fluid-structure systems, *International Journal for Numerical Methods in Engineering* 1978; **13**:139–150.

[25] 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.

[26] 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.

[27] 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.

[28] Hernandez V, Roman JE, Tomas A, Vidal V. Krylov-Schur methods in SLEPc, Technical report, Tech. Rep. CSE-2008-13, University of California, Davis, USA, 2008. Available at http://www. grycap. upv. es/slepc.

[29] Huang TM, Jia Z, Lin WW. Convergence of q-Ritz pairs, refined q-Ritz vectors and q-Rayleigh-Ritz method for quadratic eigenvalue problems. arXiv:math/1109.6426v1, 2011.

[30] Hwang TM, Lin WW, Liu JL, Wang W. Jacobi–Davidson methods for cubic eigenvalue problems. *Numerical Linear Algebra with Applications* 2005; **12**:605–624.

[31] Hwang TM, Lin WW, Wang WC, Wang W. Numerical simulation of three dimensional quantum dot. *Journal of Computational Physics* 2004; **196**(1):208–232.

[32] Jia Z. Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems. *Linear Algebra and its Applications* 1997; **259**:1–23.

[33] Jia Z, Stewart GW. An analysis of the Rayleigh-Ritz method for approximating eigenspaces. *Mathematics of Computation* 2001; **70**(234):637–647.

[34] Jia Z, Sun Y. A Refined second-order Arnoldi (RSOAR) method for the quadratic eigenvalue problem and implicit restarting, 2010. Available from: http://arxiv.org/abs/1005.3947.

[35] Kehr-Kandille V, Ohayon R. Elastoacoustic damped vibrations. Finite element and modal reduction methods, in New Advances in Computational Structural Mechanics, O. C. Zienkiewicz and P. Ladev‘eze, eds., Elsevier, Amsterdam, pp. 321–334, 1992.

[36] Lancaster P. Lambda-matrices and vibrating systems. Pergamon Press: Oxford, 1966.

[37] Lee CR. Residual Arnoldi methods: theory, package and experiments. Ph.D thesis, Department of Computer Science, University of Maryland at College Park, 2007.

[38] Lee CR, Stewart GW. Analysis of the residual Arnoldi method. Technical Report, UMIACS TR-2007-45, CMSC TR-4890, University of Maryland, 2007.

[39] Lehoucq RB, Sorensen DC, Yang C. ARPACK USERS GUIDE: Solution of large scale eigenvalue problems with implicitely restarted Arnoldi methods. *SIAM*, Philadelphia, 1998.

[40] 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.

[41] 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.

[42] Mackey DS, Mackey N, Mehl C, Mehrmann V. Vector spaces of linearizations for matrix polynomials. *SIAM Journal on Matrix Analysis and Applications* 2006; **28**(4):971–1004.

[43] Meerbergen K. Locking and restarting quadratic eigenvalue solvers. *SIAM Journal on Scientific Computing* 2001; **22**(5):1814–1839.

[44] Mehrmann V, Voss H. Nonlinear eigenvalue problems: a challenge for modern eigenvalue methods. *GAMM Mitteilungen* 2004; **27**.

[45] Moler CB, Stewart GW. An algorithm for generalized matrix eigenvalue problems. *SIAM Journal on Numerical Analysis* 1973; **10**(2):241–256.

[46] Morgan RB. On restarting the Arnoldi method for large non-symmetric eigenvalue problems. *Mathematics of Computation* 1996; **65**(215):1213–1230.

[47] Neumaier A. Residual inverse iteration for the nonlinear eigenvalue problem. *SIAM Journal on Numerical Analysis* 1985; **22**(5):914–923.

[48] Parlett BN, Saad Y. Complex shift and invert strategies for real matrices. *Linear Algebra and its Applications* 1987; **88-89**:575–595.

[49] Parlett BN. The symmetric eigenvalue problem. SIAM, Philadelphia, 1998.

[50] Raviart PA, Thomas JM. A mixed finite element method for second order elliptic problems, in Mathematical Aspects of Finite Element Methods, Lecture Notes in Math. 606, *Springer-Verlag*, Berlin, Heidelberg, pp. 292–315, 1977.

[51] Rodríguez R, Solomin J. The order of convergence of eigenfrequencies in finite element approximations of fluid-structure interaction problems, *Mathematics of Computation* 1996; **65**(216):1463–1475.

[52] Ruhe A. Algorithms for the nonlinear eigenvalue problem. *SIAM Journal on Numerical Analysis* 1973; **10**(4):674–689.

[53] Saad Y. Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices. *Linear Algebra and its Applications* 1980; **34**:269–295

[54] Saad Y. Numerical methods for large eigenvalue problems. Manchester University Press: Manchester, U.K., 1992.

[55] Sadkane M. Block Amoldi and Davidson methods for nonsymmetric large eigenvalue problems. *Numerische Mathematik* 1993; **64**(1):195–211.

[56] Simoncini V. Variable accuracy of matrix-vector products in projection methods for eigencomputation *SIAM Journal on Numerical Analysis* 2005; **43**(3):1155–1174.

[57] Sleijpen GLG, Booten AGL, Fokkema DR, van der Vorst HA. Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems. *BIT* 1996; **36**(3):595–633.

[58] Sleijpen GLG, van der Vorst HA, van Gijzen M. Quadratic eigenproblems are no problem. *SIAM News* 1996; **29**(7):8–9.

[59] 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.

[60] Sorensen DC. Truncated QZ methods for large scale generalized eigenvalue problems. *Electronic Transactions on Numerical Analysis* 1998; **7**:141–162.

[61] Stewart GW. A Krylov–Schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications* 2001; **23**(3):601–614.

[62] Stewart GW. Matrix Algorithms, Volume II: Eigenvalues. *SIAM*, Philadelphia, PA, 2001.

[63] Stewart GW. Addendum to "A Krylov–Schur algorithm for large eigenproblems". *SIAM Journal on Matrix Analysis and Applications* 2002; **24**(2):599–601.

[64] Su Y, Bai Z. Solving rational eigenvalue problems via linearization. *SIAM Journal on Matrix Analysis and Applications* 2011; **32**(1):201–216.

[65] Tisseur F. Backward Error and Condition of Polynomial Eigenvalue Problems. *Linear Algebra and its Applications* 2000; **309**:339–361.

[66] Tisseur F, Meerbergen K. The quadratic eigenvalue problem. *SIAM Review* 2001; **43**(2):235–286.

[67] Van der Vorst HA. Computational methods for large eigenvalue problems. In Handbook of Numerical Analysis, vol. VIII. North-Holland: Amsterdam, 2002; 3–179.

[68] Voss H. An Arnoldi method for nonlinear eigenvalue problems. *BIT* 2004; **44**(2):387–401.

[69] Voss H. Iterative projection methods for computing relevant energy states of a quantum dot. *Journal of Computational Physics* 2006; **217**:824–833.

[70] Voss H. A Jacobi–Davidson method for nonlinear and nonsymmetric eigenproblems. *Computers & Structures* 2007; **85**(17-18):1284–1292.

[71] Wang X, Bathe KJ. Displacement/pressure based mixed finite element formulations for acoustic fluid-structure interaction problems, *International Journal for Numerical Methods in Engineering* 1997; **40**:2001–2017.

[72] 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.

[73] Watkins DS. The matrix eigenvalue problem: GR and Krylov subpsace methods. SIAM Publications, Philadelphia, PA, 2007.

[74] Wei S, Kao I. Vibration analysis of wire and frequency response in the modern wiresaw manufacturing process. *Journal of Sound and Vibration* 2000; **231**(5):1383–1395.

[75] Ye Q. An iterated shift-and-invert Arnoldi algorithm for quadratic matrix eigenvalue problems. *Applied Mathematics and Computation* 2006; **172**(2):818–827.