A Bisection Method for the Banded Hyperbolic Quadratic Eigenvalue Problem

Ahmed T. Ali

A Bisection Method for the Banded Hyperbolic Quadratic Eigenvalue Problem

by

AHMED T. ALI

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2016

To my mother and my father

who set the example and who made me who I am.

To my wife Heba and my children Abdullah, Zainab and Ruqayyah

who supported me all the way.

ACKNOWLEDGEMENTS

I would like to thank my advisor Prof. Reng-Ceng Li for his continuous support, patience and caring. His guidance, motivation and immense knowledge helped me do my research and finish this thesis. I have learned a lot from him as a student and as a person. I could have never had a better advisor and mentor in my Ph.D study.

Besides my advisor, I'd like to thank other members of the thesis commitee Prof. David Jorgensen, Prof. Guojun Liao, Prof. Hristo Kojouharov,and Prof. Gaik Ambartsoumian for their encouragement and insightful comments.

My special thanks goes to Prof. Jianzhong Su, Department Chair, for his support and encouragement.

Last but not the least, I would like to thank my family especially my wife Heba for her continuous support and belief in my ability of high achievements.

April 11, 2016

ABSTRACT

A Bisection Method for the Banded Hyperbolic Quadratic Eigenvalue Problem

Ahmed T. Ali, Ph.D.

The University of Texas at Arlington, 2016

Supervising Professor: Ren-Cang Li

It is well-known that the eigenvalues of a Hermitian matrix in a given interval can be approximated within a predefined error tolerance using the bisection method as a direct application of the Sylvester's Law of Inertia. In this thesis, we will develop a bisection method for the hyperbolic quadratic eigenvalue problem (HQEP) which is guaranteed to have $2n$ real eigenvalues for a problem of size n. A number of numerical methods are available to solve HQEPs. Matlab's `polyeig` uses the QZ algorithm on the problem after linearizing it to a pencil of size $2n$. Another approach is by finding a solvent matrix. Both approaches ignore any banded structure of the problem. For the tri-diagonal HQEPs, an approach to approximate the eigenvalues by efficiently solving the characteristic equation was also proposed. The method can't be applied to higher banded HQEPs efficiently. Our method will avoid converting the HQEP to a definite pencil of order $2n$ by working on the HQEP directly taking into consideration any banded structure of the problem. Our method can be applied to large banded HQEPs and produces more accurate eigenvalue approximations compared to the approaches stated.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ALGORITHMS

CHAPTER 1

Introduction

The *quadratic eigenvalue problem* (QEP) is a nonlinear eigenvalue problem with important applications in dynamic analysis of structures, non-linear vibration theory and fluid dynamics. This thesis focuses on large scale banded *hyperbolic* quadratic eigenvalue problem (HQEP) which is known to admit real eigenvalues, similar to the definite generalized eigenvalue problem and the standard Hermitian eigenvalue problem.

This thesis was organized to be as self contained as possible in introducing HQEP. In chapter 1 we introduce HQEP, and the origins of the problem. We also introduce some results concerning matrix polynomials and finally we introduce the concept of eigenvalue curves which will be of great importance in our analysis of HQEP and in the derivation of our algorithms.

In chapter 2, we dig deeper into the theory behind our understanding of the problem, the properties of HQEP solutions, hyperbolicity and overdamping of HQEP, the concept of symmetric matrix inertia which will be the basis of our algorithms, and finally survey the available techniques to the solution of HQEP. Chapter 2 doesn't depend on chapter 1 except for section 1.1.

In Chapter 3, we begin implementing the bisection method on HQEP by first understanding the bisection method application on the symmetric eigenvalue problem (SEP) before extending the bisection method for HQEP. In this chapter, more analysis and results of HQEP are introduced and examined.

In chapter 4, we implement a stable and efficient algorithm for tri-diagonal HQEPs and examine the difficulties of extending the same algorithm to the banded HQEPs.

In chapter 5, we introduce a solution to the difficulties of extending a stable and efficient bisection algorithm to banded HQEPs wider than tri-diagonal. Section 5.2 is also self contained as it has all the final versions of the procedures needed to implement a complete solution using the bisection method on banded HQEPs. Finally, in section 5.3 we illustrate the results of our approach compared to two other approaches.

1.1 Definition

Definition 1.1.1 (HQEP)**.** The hyperbolic quadratic eigenvalue problem (HQEP) is to find scalars λ and nonzero vectors x satisfying

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

where A, B, C are $n \times n$ symmetric matrices, $A \succ 0$ (positive definite), and

$$(x^T Bx)^2 > 4(x^T Ax)(x^T Cx), \quad \forall\, 0 \neq x \in \mathbb{C}^n. \tag{1.2}$$

An HQEP is algebraically different from the standard eigenvalue problem

$$Ax = \lambda x,$$

and the generalized eigenvalue problem (GEP)

$$Ax = \lambda Bx,$$

in that HQEP has $2n$ real eigenvalues and $2n$ eigenvectors that are not linearly independent in $\mathbb{R}^n$. HQEP is a restricted eigenvalue problem of a larger class of nonlinear eigenvalue problems called the *symmetric quadratic eigenvalue problem* (QEP) where the only condition is that A, B, C are $n \times n$ Hermitian matrices. A QEP will also have $2n$ eigenvalues (finite or infinite) and they may be complex. Throughout this study, we will call, λ from the previous definition a ***quadratic eigenvalue***, x a ***quadratic eigenvector***, and (λ, x) a ***quadratic eigenpair*** of the HQEP in (1.1). The ***spectrum*** of $Q(\lambda)$ is the set of all quadratic eigenvalues of the HQEP.

1.2 The Rise of HQEP

HQEP arises in many applications where solving a system of second order linear differential equations is required,

$$Ay''(t) + By'(t) + Cy(t) = 0, \tag{1.3}$$

where the vector-valued function $y(t) = [y_1(t)\ y_2(t)\ \dots\ y_n(t)]^T$, and $A, B, C \in \mathbb{R}^{n\times n}$.

The general solution has the form

$$y(t) = \sum_{i=1}^{2n} \kappa_i e^{\lambda_i t} x_i, \tag{1.4}$$

where x_i is the quadratic eigenvector associated with the quadratic eigenvalue λ_i obtained from solving HQEP, and κ_i are scalar constants that can be determined using initial conditions.

Example 1.2.1. Here is a simple example of dynamic analysis of structural system. Consider a two mass damped system as in Figure 1.1 where m_i is the mass, y_i is the displacement of the mass, c_i is the damping coefficient, and k_i is the spring constant. Applying Newton's second law to m_1 and m_2 respectively yields

$$m_1 y_1'' + (c_1 + c_2)y_1' - c_2 y_2' + (k_1 + k_2)y_1 - k_2 y_2 = 0,$$

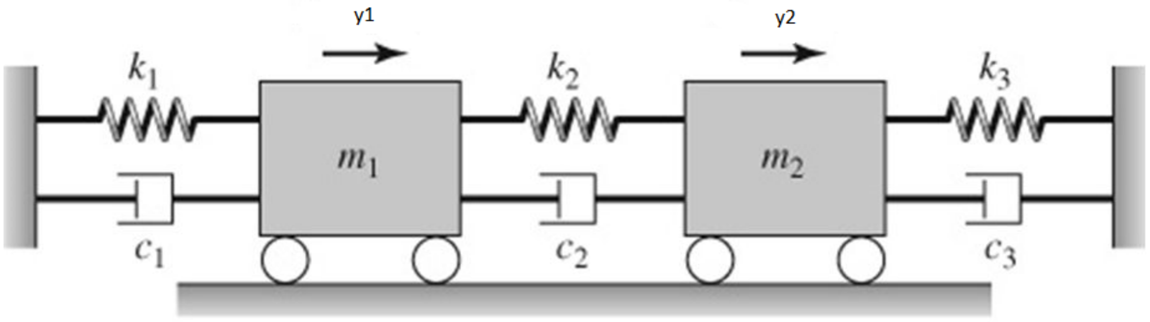
$$m_2 y_2'' + (c_2 + c_3)y_2' - c_2 y_1' + (k_2 + k_3)y_2 - k_2 y_1 = 0,$$

which can be written as

$$\underbrace{\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} y_1'' \\ y_2'' \end{bmatrix}}_{Y''(t)} + \underbrace{\begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 + c_3 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} y_1' \\ y_2' \end{bmatrix}}_{Y'(t)} + \underbrace{\begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}}_{Y(t)} = 0$$

and has the same form as (1.3). In case of n masses then we will get a tri-diagonal system of order n.

Figure 1.1. Two Mass Spring Damped system.



1.3 Symmetric Matrix Polynomials

The $Q(\lambda)$ of HQEP (and QEP in general) is a matrix polynomial of the second degree and from there comes the word *quadratic*. The HQEP therefore inherits many of the characteristics of matrix polynomial problems which themselves inherit from the matrix analytic function problems. In this section, we will study how the polynomial eigenvalues and eigenvectors behave as a function in λ and then apply the results to the HQEP.

Consider $n \times n$ symmetric matrix whose elements are polynomials with the highest degree d in the scalar parameter λ,

$$F(\lambda) = \begin{bmatrix} f_{11}(\lambda) & f_{12}(\lambda) & f_{13}(\lambda) & \dots & f_{1n}(\lambda) \\ f_{21}(\lambda) & f_{22}(\lambda) & f_{23}(\lambda) & \dots & f_{2n}(\lambda) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{n1}(\lambda) & f_{n2}(\lambda) & f_{n3}(\lambda) & \dots & f_{nn}(\lambda) \end{bmatrix}, \tag{1.5}$$

where $f_{ij}(\lambda)$ is a polynomial in λ of degree d, or equivalently

$$F(\lambda) = \sum_{i=0}^{d} \lambda^i A_i = \lambda^d A_d + \lambda^{d-1} A_{d-1} + ... + \lambda A_1 + A_0, \tag{1.6}$$

where $A_0, A_1, ..., A_d$ are $n \times n$ real symmetric matrices with constant elements, and A_0 is non-singular. This is a symmetric matrix polynomial of degree d and order n.

Definition 1.3.1 (Derivative of a matrix polynomial)**.** The first derivative of a matrix polynomial $F'(\lambda)$ is a matrix polynomial such that

$$F'(\lambda) = \begin{bmatrix} f'_{11}(\lambda) & f'_{12}(\lambda) & f'_{13}(\lambda) & \dots & f'_{1n}(\lambda) \\ f'_{21}(\lambda) & f'_{22}(\lambda) & f'_{23}(\lambda) & \dots & f'_{2n}(\lambda) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f'_{n1}(\lambda) & f'_{n2}(\lambda) & f'_{n3}(\lambda) & \dots & f'_{nn}(\lambda) \end{bmatrix},$$

or equivalently

$$F'(\lambda) = \sum_{i=1}^{d} i\lambda^{i-1} A_i = d\lambda^{d-1} A_d + (d-1)\lambda^{d-2} A_{d-1} + ... + \lambda A_2 + A_1.$$

For a given real value of λ, the symmetric matrix polynomial $F(\lambda)$ is a real symmetric matrix and hence there exist n eigenvalues, counting multiplicities, denoted by $\mu_i(\lambda)$ and eigenvectors denoted by $u_i(\lambda)$ such that

$$F(\lambda) u_i(\lambda) = \mu_i(\lambda) u_i(\lambda), \qquad \text{for } i = 1, 2, ..., n.$$

Note that $\mu_i(\lambda)$ is a scalar function in the parameter λ while $u_i(\lambda)$ is a vector-valued function in λ.

A matrix polynomial of degree d and order n has $d \cdot n$ *polynomial eigenvalues* λ_i counting multiplicities and $d \cdot n$ corresponding *polynomial eigenvectors* x_i that are linearly dependent in the space $\mathbb{R}^n$ if $d > 1$:

$$F(\lambda_i)x_i = 0, \qquad \text{for } i = 1, 2, ..., d \cdot n, \tag{1.7}$$

and we will call the eigenpair (λ_i, x_i) *polynomial eigenpair.*

Theorem 1.3.1. *If ($\mu(\tilde{\lambda})$,$u(\tilde{\lambda})$) is an eigenpair of the matrix $F(\tilde{\lambda})$ for some $\lambda = \tilde{\lambda}$ and $\mu(\tilde{\lambda}) = 0$,then ($\tilde{\lambda}$,$u(\tilde{\lambda})$) is a polynomial eigenpair of the matrix polynomial $F(\lambda)$.*

Proof. $F(\tilde{\lambda})u(\tilde{\lambda}) = \mu(\tilde{\lambda})u(\tilde{\lambda}) = 0.$ □

Theorem 1.3.2. *$\mu_i(\lambda)$ is a continuous function in λ for $i = 1, 2, ..., n$.*

Proof. Eigenvalues of a matrix are continuous functions of the matrix elements from Proposition 4.4 [3]. But from (1.5), the elements of a matrix polynomial are polynomials in λ and thus eigenvalues of a matrix polynomial are continuous in λ. □

Theorem 1.3.3. *Let $u_i(\lambda)$ be an eigenvector of a symmetric matrix polynomial $F(\lambda)$ and assume its corresponding eigenvalue $\mu_i(\lambda)$ is simple in an interval $\mathcal{I} : \alpha < \lambda < \beta$. Then $u_i(\lambda)$ is a continuous function in λ in the interval $\mathcal{I}$.*

Proof. Since $F(\lambda)$ is a compact, self-adjoint linear operator in $\mathbb{R}^n$, and $\mu_i(\lambda)$ is simple in $\mathcal{I}$, from [4] $u_i(\lambda)$ is continuous in $\mathcal{I}$. □

In the next theorem we will show that $\mu_i(\lambda)$ is not only a continuous scalar function in λ but also a smooth function for almost every value of λ.

Theorem 1.3.4. *Consider a symmetric matrix polynomial $F(\lambda)$. Let $\mu(\lambda)$ be a simple eigenvalue of $F(\lambda)$ in an interval $\mathcal{I} : \alpha < \lambda < \beta$, with corresponding eigenvector $u(\lambda)$ with $\|u\|_2 = 1$. Then*

$$\mu'(\lambda) = u^T(\lambda)F'(\lambda)u(\lambda). \tag{1.8}$$

Proof. Since $\mu(\lambda)$ is a continuous function in λ from Theorem 1.3.2 and using the Taylor series of (1.6) we get

$$F(\lambda + t) = F(\lambda)u(\lambda + t) + tF'(\lambda)u(\lambda + t) + t^2H(\lambda, t),$$

where H is a matrix polynomial in the parameters λ and t, and $\lambda + t \in \mathcal{I}$. For a sufficiently small t, $F(\lambda + t)$ has a simple eigenvalue $\mu(\lambda + t)$ with corresponding eigenvector $u(\lambda + t)$. We get

$$F(\lambda + t)u(\lambda + t) = \mu(\lambda + t)u(\lambda + t), \tag{1.9}$$

but also we have

$$\begin{aligned}
&\mu(\lambda + t)u^T(\lambda)u(\lambda + t) \\
= \;& u^T(\lambda)F(\lambda + t)u(\lambda + t) \\
= \;& u^T(\lambda)\Big(F(\lambda) + tF'(\lambda) + t^2H(\lambda, t)\Big)u(\lambda + t) \\
= \;& \mu(\lambda)u^T(\lambda)u(\lambda + t) + tu^T(\lambda)F'(\lambda)u(\lambda + t) + t^2u^T(\lambda)H(\lambda, t)u(\lambda + t).
\end{aligned}$$

Collecting μ on the left side and dividing by t we get

$$\frac{\mu(\lambda + t) - \mu(\lambda)}{t} = \frac{u^T(\lambda)F'(\lambda)u(\lambda + t)}{u^T(\lambda)u(\lambda + t)} + tH(\lambda, t).$$

Taking the limit as $t \to 0$ we get

$$\mu'(\lambda) = \frac{u^T(\lambda)F'(\lambda)u(\lambda)}{u^T(\lambda)u(\lambda)} = u^T(\lambda)F'(\lambda)u(\lambda).$$

□

Theorems 1.3.2 - 1.3.4 are the special cases of more general results on matrix functions [5, 6, 7].

Remark 1. We could have taken the first derivative of $\mu(\lambda)$ directly to get the same result as follows. We have

$$\|u\|_2 = 1 \implies u^T(\lambda)u(\lambda) = 1 \implies \frac{d}{d\lambda}(u^T(\lambda)u(\lambda)) = u'^T(\lambda)u(\lambda) + u^T(\lambda)u'(\lambda) = 0.$$

Now by taking the derivative of $\mu(\lambda) = u^T(\lambda)F(\lambda)u(\lambda)$ directly we may get the same result. But then we would have to deal with the issue of the differentiability of the eigenvector $u(\lambda)$ which was avoided in the proof.

1.4 Eigenvalue Function $\mu(\lambda)$

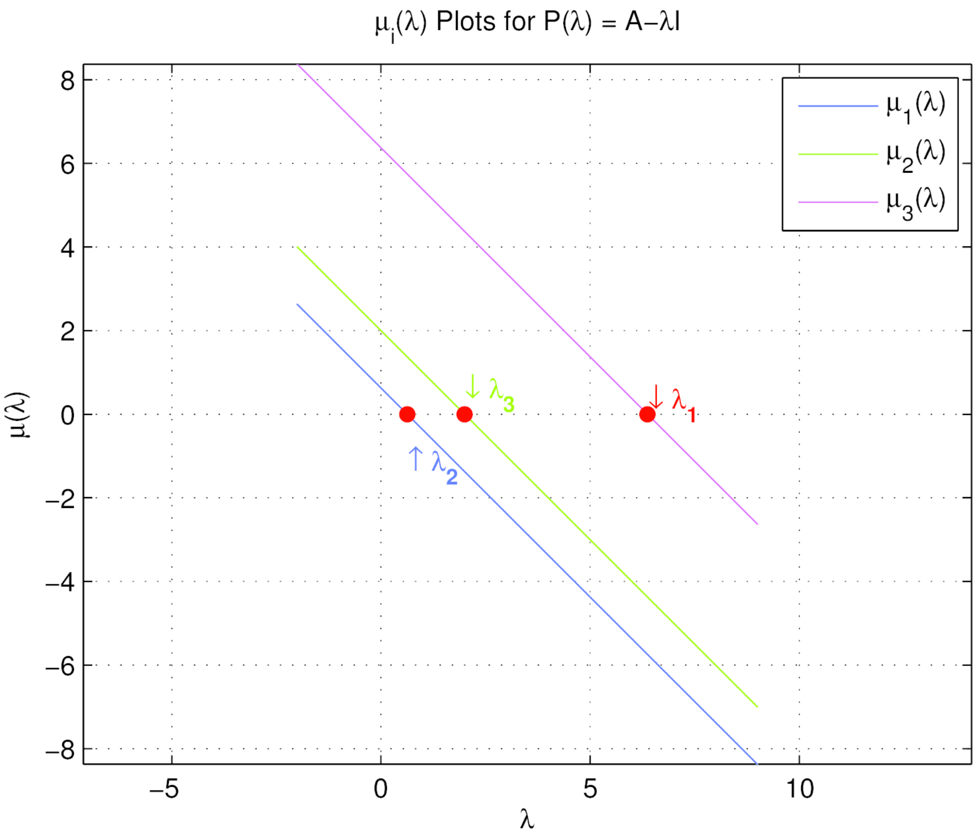
Let A be a square symmetric matrix of size n with simple eigenvalues λ_i, and corresponding eigenvector x_i where $i = 1, 2, \cdots, n$. Then to find the eigenvalues we need to solve the characteristic equation $\det(A - \lambda I) = 0$ or in other words we need to solve the eigenvalue problem for the linear pencil (matrix polynomial of degree 1),

$$P(\lambda) = A - \lambda I.$$

Using Matlab we can plot for a fixed value of λ, the points $(\lambda, \mu_i(\lambda))$ where $\mu_i(\lambda)$ are the eigenvalues of the matrix $P(\lambda)$ for $i = 1, 2, \cdots, n$. If we repeat this step while changing the value of λ using some convenient step and keep plotting the eigenvalues, we will get the plot of all the eigenvalues of the pencil $P(\lambda)$ as curves like in Figure 1.2 which shows the eigenvalue functions $\mu(\lambda)$ of the matrix polynomial $P(\lambda) = A - \lambda I$ where

$$A = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix}.$$

Figure 1.2. Plot of Eigenvalue Function $\mu(\lambda)$.



The matrix A has the eigenvalues $(0.6277, 2.0000, 6.3723)$ and these are exactly the roots of the corresponding eigenvalue curves $\mu(\lambda)$ (the red dots on the λ-axis). Furthermore, all the $\mu(\lambda)$ curves have the same slope as a direct application of Theorem 1.3.4,

$$\mu'(\lambda) = u^T(\lambda)P'(\lambda)u(\lambda) = -1.$$

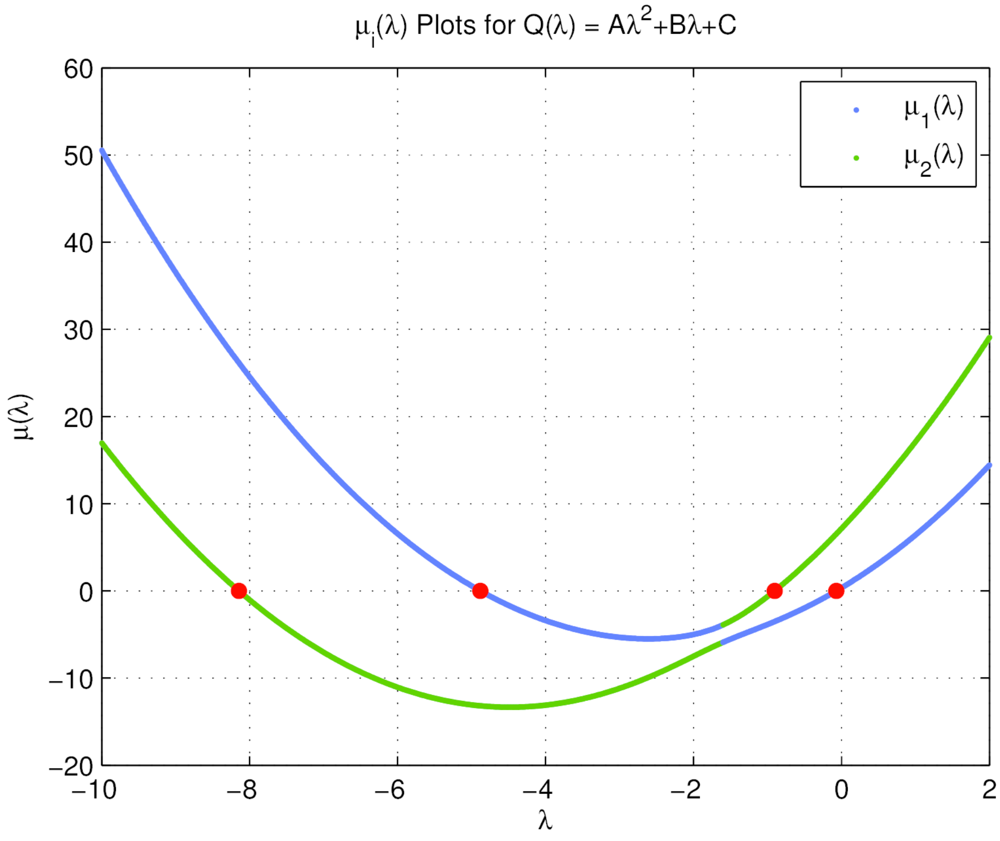
The previous example may appear very trivial, but the same technique will be used extensively in the next section where we will study the properties of HQEPs and in general QEPs.

Example 1.4.1. Now consider the HQEP defined by

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 5 & 0 \\ 0 & 9 \end{bmatrix}, C = \begin{bmatrix} \frac{1}{2} & 1 \\ 1 & 7 \end{bmatrix}.$$

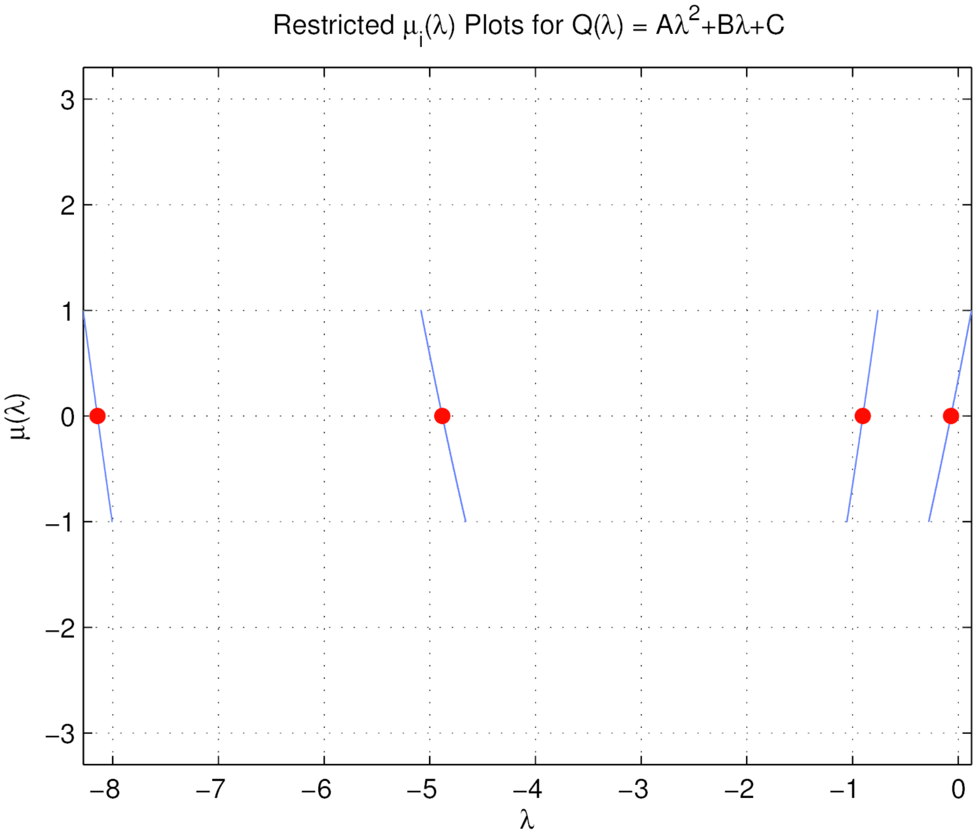
The quadratic eigenvalues of this HQEP are $(-8.1453, -4.8820, -0.9031, -0.0696)$ using Matlab's `polyeig` command. Note that these values are exactly where the $\mu(\lambda)$ curves intersect the λ-axis as in Figure 1.3.

Figure 1.3. Plot of Eigenvalue Function $\mu(\lambda)$ of an HQEP.



The slope of the eigenvalue functions $\mu(\lambda)$ changes with λ, also we are only concerned with the values of the functions around the λ-axis. If we restrict the plot to the values where $|\mu(\lambda_i)| \leq 1$, we may get a more meaningful plot in Figure 1.4 than in Figure 1.3.

Figure 1.4. Restricted Plot of Eigenvalue Function $\mu(\lambda)$ of an HQEP.



Now, we can see where the eigenvalue curves cross the λ-axis, and hence their roots. We can also see the slope of each curve at its roots where we will use in the theory of HQEP in the next chapter.

CHAPTER 2

Properties of HQEP

HQEP has special properties that we will exploit when developing a numerical solution to the problem.

2.1 Properties of HQEP

Theorem 2.1.1. *The quadratic eigenvalues of an HQEP are real.*

Proof. Consider ($\hat{\lambda}$,$\hat{x}$) a quadratic eigenpair of an HQEP such that $Q(\hat{\lambda})\hat{x} = 0$ then we can form the ***Rayleigh Quotient*** for the HQEP to obtain

$$0 = \hat{x}^T Q(\hat{\lambda})\hat{x} = \hat{\lambda}^2 \hat{x}^T A\hat{x} + \hat{\lambda}\hat{x}^T B\hat{x} + \hat{x}^T C\hat{x}.$$

Then by applying the quadratic formula, $\hat{\lambda}$ will get one of the two values:

$$\frac{-\hat{x}^T B\hat{x} \pm \sqrt{(\hat{x}^T B\hat{x})^2 - 4(\hat{x}^T A\hat{x})(\hat{x}^T C\hat{x})}}{2\hat{x}^T A\hat{x}}. \tag{2.1}$$

Hence $\hat{\lambda} \in \mathbb{R}$ by Definition 1.1.1. □

Theorem 2.1.2 ([8, 9, 10])**.** *The HQEP as defined in Definition 1.1.1 is equivalent to the 1st degree symmetric matrix polynomial of order* $2n$

$$L(\lambda) = \underbrace{\begin{bmatrix} B & A \\ A & 0 \end{bmatrix}}_{M} \lambda + \underbrace{\begin{bmatrix} C & 0 \\ 0 & -A \end{bmatrix}}_{N}, \tag{2.2}$$

and if $(\hat{\lambda}, \begin{bmatrix} \hat{x} \\ y \end{bmatrix})$ *is an eigenpair of* (2.2)*, then* $(\hat{\lambda}, \hat{x})$ *is a quadratic eigenpair of the HQEP and* $y = \hat{\lambda}\hat{x}$*. Furthermore,* $L(\lambda)$ *as defined in* (2.2) *is a* **definite pencil** (*symmetric and* $\exists\hat{\lambda} \in \mathbb{R}$ *such that* $L(\hat{\lambda}) \prec 0$).

Proof. We have

$$\underbrace{\begin{bmatrix} I & I\lambda \\ 0 & -A^{-1} \end{bmatrix}}_{E(\lambda)} \underbrace{\begin{bmatrix} B\lambda + C & A\lambda \\ A\lambda & -A \end{bmatrix}}_{L(\lambda)} \underbrace{\begin{bmatrix} I & 0 \\ I\lambda & I \end{bmatrix}}_{F(\lambda)} = \begin{bmatrix} Q(\lambda) & 0 \\ 0 & I \end{bmatrix}. \tag{2.3}$$

Since A is non-singular, $\det(E(\lambda)) \neq 0$. Notice also $\det(F(\lambda)) \neq 0$. Hence

$$\det(Q(\lambda)) = 0 \iff \det(L(\lambda)) = 0,$$

i.e. both have the same zeros. Also,

$$0 = L(\hat{\lambda}) \begin{bmatrix} \hat{x} \\ y \end{bmatrix} = \begin{bmatrix} B\hat{\lambda} + C & A\hat{\lambda} \\ A\hat{\lambda} & -A \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{\lambda}\hat{x} \end{bmatrix} = \begin{bmatrix} A\hat{\lambda}^2\hat{x} + B\hat{\lambda}\hat{x} + C\hat{x} \\ A\hat{\lambda}\hat{x} - A\hat{\lambda}\hat{x} \end{bmatrix} = \begin{bmatrix} Q(\hat{\lambda})\hat{x} \\ 0 \end{bmatrix},$$

and this proves the equivalency. □

Theorem 2.1.3 ([11])**.** *A QEP where A, B, C are $n \times n$ Hermitian matrices and $A \succ 0$ is hyperbolic if and only if $Q(\hat{\lambda}) \prec 0$ for some $\hat{\lambda} \in \mathbb{R}$.*

Proof. ($\Longleftarrow$)[12] Let $M_x \equiv x^T M x$ for any symmetric matrix M and a nonzero vector x. We have

$$\begin{aligned}
& Q(\hat{\lambda}) \prec 0 \text{ for some } \hat{\lambda} = \lambda \in \mathbb{R} \\
\Longrightarrow \quad & \hat{\lambda}^2 A + \hat{\lambda} B + C \prec 0 \\
\Longrightarrow \quad & \hat{\lambda}^2 A_x + \hat{\lambda} B_x + C_x < 0, \ \forall 0 \neq x \in \mathbb{C}^n \\
\Longrightarrow \quad & \hat{\lambda}^2 A_x + C_x < \hat{\lambda} B_x \\
\Longrightarrow \quad & 2\sqrt{\hat{\lambda}^2 A_x C_x} < \hat{\lambda} B_x, \quad (2\sqrt{ab} < a + b, \text{ for } 0 < a, b \in \mathbb{R}) \\
\Longrightarrow \quad & 4\hat{\lambda}^2 A_x C_x < \hat{\lambda}^2 B_x^2 \\
\Longrightarrow \quad & B_x^2 - 4 A_x C_x > 0.
\end{aligned}$$

($\implies$)[13] $Q(\lambda)$ is equivalent to the definite pencil $L(\lambda)$, hence according to Theorem 1.2 in [14] there exist $\alpha, \beta \in \mathbb{R}$ and $\beta > 0$ such that

$$\alpha \begin{bmatrix} B & A \\ A & 0 \end{bmatrix} - \beta \begin{bmatrix} C & 0 \\ 0 & -A \end{bmatrix} \succ 0.$$

But

$$\begin{aligned} \alpha \begin{bmatrix} B & A \\ A & 0 \end{bmatrix} - \beta \begin{bmatrix} C & 0 \\ 0 & -A \end{bmatrix} &= \begin{bmatrix} \alpha B - \beta C & \alpha A \\ \alpha A & \beta A \end{bmatrix} \\ &= \begin{bmatrix} I & \frac{\alpha}{\beta} I \\ 0 & I \end{bmatrix} \begin{bmatrix} \alpha B - \beta C - \frac{\alpha^2}{\beta} A & 0 \\ 0 & \beta A \end{bmatrix} \begin{bmatrix} I & 0 \\ \frac{\alpha}{\beta} I & I \end{bmatrix} \\ &\implies \frac{\alpha^2}{\beta} A - \alpha B + \beta C \prec 0. \end{aligned}$$

By choosing $\hat{\lambda} = -\frac{\alpha}{\beta}$ we get $L(\hat{\lambda}) \prec 0$ which implies that $Q(\hat{\lambda}) \prec 0$. □

Definition 2.1.1. The quadratic eigenpair $(\hat{\lambda}, \hat{x})$ is said to be of a *positive (negative) type* if

$$\hat{x}^T Q'(\hat{\lambda}) \hat{x} > 0 \quad (\hat{x}^T Q'(\hat{\lambda}) \hat{x} < 0), \tag{2.4}$$

where $Q'(\lambda)$ is the first derivative of $Q(\lambda)$ with respect to λ.

Since $Q'(\lambda) = 2\lambda A + B$, from Definition 2.1.1 a quadratic eigenvalue λ is of *positive (negative) type* if

$$\hat{\lambda} > -\frac{\hat{x}^T B \hat{x}}{2\hat{x}^T A \hat{x}} \quad (\hat{\lambda} < -\frac{\hat{x}^T B \hat{x}}{2\hat{x}^T A \hat{x}}).$$

Denote a quadratic eigenvalue of positive (negative) type by λ^+ (λ^-), the corresponding quadratic eigenvectors of positive (negative) type by $\overset{+}{x}$, and $(\overset{-}{x})$ respectively. Since all the quadratic eigenvalues will have the form (2.1), the ***principal*** (positive type) quadratic eigenvalues will have the form

$$\lambda_i^+ = \frac{-\overset{+}{x}_i^T B x_i + \sqrt{(\overset{+}{x}_i^T B \overset{+}{x}_i)^2 - 4(\overset{+}{x}_i^T A \overset{+}{x}_i)(\overset{+}{x}_i^T C \overset{+}{x}_i)}}{2\overset{+}{x}_i^T A \overset{+}{x}_i},$$

while the ***secondary*** (negative type) quadratic eigenvalues will have the form

$$\lambda_i^- = \frac{-\bar{x}_i^T B x_i + \sqrt{(\bar{x}_i^T B \bar{x}_i)^2 - 4(\bar{x}_i^T A \bar{x}_i)(\bar{x}_i^T C \bar{x}_i)}}{2\bar{x}_i^T A \bar{x}_i}.$$

Furthermore, if $\mu_j(\lambda_i^+) = 0$ then $\mu_j'(\lambda_i^+) > 0$, also if $\mu_k(\lambda_i^-) = 0$ then $\mu_k'(\lambda_i^-) < 0$ (see Figure 2.1).

Example 2.1.1. Consider the HQEP defined by

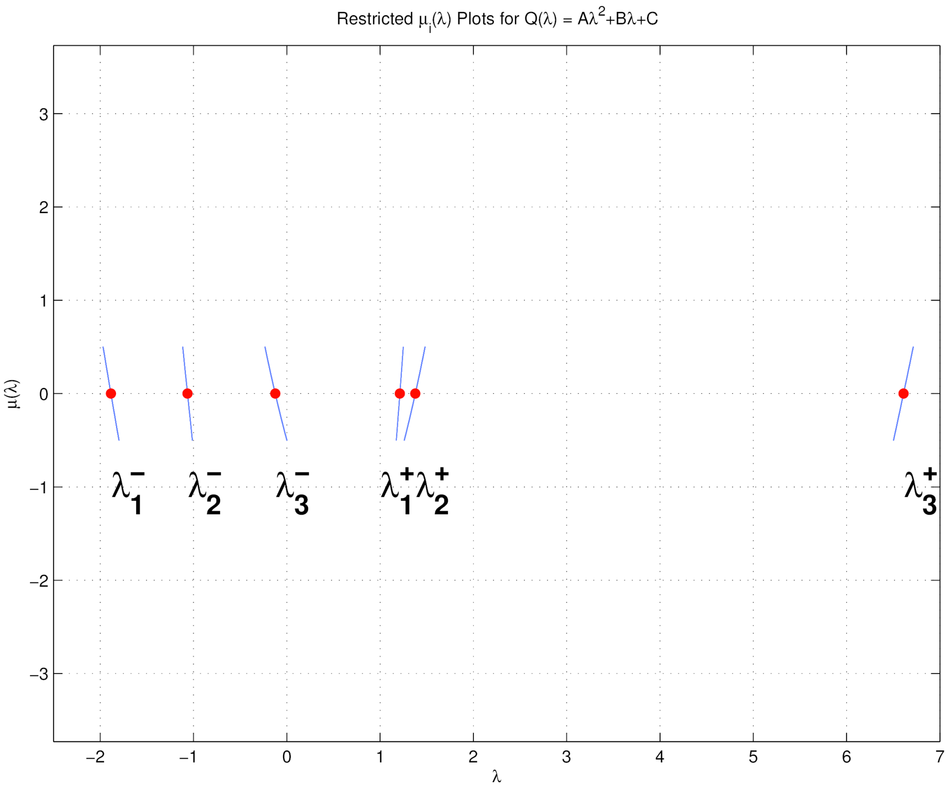
$$A = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix}, B = \begin{bmatrix} -2 & -1 & -1 \\ -1 & -3 & 2 \\ -1 & 2 & -1 \end{bmatrix}, C = \begin{bmatrix} -5 & 1 & -2 \\ 1 & -4 & -3 \\ -2 & -3 & -5 \end{bmatrix},$$

which has quadratic eigenvalues using Matlab's `polyeig`

λ_1^-	-1.8856
λ_2^-	-1.0644
λ_3^-	-0.1242
λ_1^+	1.2117
λ_2^+	1.3772
λ_3^+	6.6104

which agrees with Figure 2.1, and note the slope of $\mu(\lambda)$ curves as they cross the λ-axis.

Figure 2.1. Spectrum of an HQEP of order 3.



For a fixed λ, $Q(\lambda)$ is a symmetric matrix of order n and has n real eigenvalues. From Example 2.1.1 we can see that for a small enough fixed λ, say $\lambda < \lambda_1^-$, we get $Q(\lambda) \succ 0$ and hence all the eigenvalues of such symmetric matrix will be positive. If we choose a large enough fixed λ, say $\lambda > \lambda_3^+$, all the eigenvalues of the symmetric matrix $Q(\lambda)$ will also be positive. For $Q(\lambda)$ to be negative definite for some $\lambda = \hat{\lambda}$, we must have all the eigenvalues of the symmetric matrix $Q(\hat{\lambda})$ negative. This implies that positive eigenvalues of the matrix $Q(\lambda)$ must decrease from the left of λ_1^- until we reach $\hat{\lambda}$ when all the eigenvalues of the matrix $Q(\lambda)$ are negative, and then increase again until all the eigenvalues are positive on the right of λ_3^+. If the quadratic eigenvalue $\tilde{\lambda}$ is of a negative type then the matrix $Q(\lambda)$ will have one less positive eigenvalue passing through $\tilde{\lambda}$ from left to right, and the reverse happen if it is of the positive type. This implies that all the secondary quadratic eigenvalues of the negative type must be on the left of $\hat{\lambda}$, and all the principle quadratic eigenvalues of the positive type must be on the right of $\hat{\lambda}$ [1]. In fact the symmetric matrix $Q(\lambda)$ is negative definite for all values of λ such that $\lambda_i^- < \lambda < \lambda_i^+$ for $i = 1 \ldots n$.

Theorem 2.1.4 ([11]). *All the quadratic eigenvalues of the HQEP are non-defective.*

2.2 Inertia of $Q(\lambda)$

Here we will study how the inertia of $Q(\lambda)$ changes as a function of λ but first we need to define the inertia of a symmetric matrix which will be applied to the symmetric matrix $Q(\lambda)$ for a fixed λ. Then we will see that the inertia of a symmetric matrix is invariant under congruence transformation.

Definition 2.2.1 (Inertia of a Symmetric Matrix). The inertia of a symmetric matrix A is the triple integers i_-, i_0, i_+, where i_- is the number of negative eigenvalues of A, i_0 is the number of zero eigenvalues of A, and i_+ is the number of positive eigenvalues of A. It will be denoted as Inertia(A)=(i_-, i_0, i_+).

Theorem 2.2.1 (Sylvester's Inertia Theorem). *Let A be symmetric and X be nonsingular. Then A and X^TAX have the same inertia.*

Proof. [3] Let A be $n \times n$ matrix. Let $\mathcal{N}$ be the negative eigenspace of A spanned by the eigenvectors corresponging to the negative eigenvalues of A. dim($\mathcal{N}$) is the number of negative eigenvalues of A. Let $\mathcal{P}$ be the nonnegative eigenspace of X^TAX where $y^TX^TAXy \geq 0$, $\forall y \in \mathcal{P}$. Let $X\mathcal{P}$ be the space of all vectors $x = Xy$ where

$y \in \mathcal{P}$. Since X is nonsingular, $\dim(X\mathcal{P}) = \dim(\mathcal{P})$. Assume the dimension of the negative eigenspace of X^TAX is less than $\dim(\mathcal{N})$ then $\dim(\mathcal{N}) + \dim(X\mathcal{P}) > n$, and $\exists x \in \mathcal{N} \cap X\mathcal{P}$, $x \neq 0$. But $x \in \mathcal{N} \implies x^TAx < 0$, and $x \in X\mathcal{P} \implies x^TAx = y^TX^TAXy \geq 0$ which is a contradiction. Hence $\dim(\mathcal{N}) = n - \dim(\mathcal{P})$. Therefore, the number of negative eigenvalues of A and X^TAX are the same. The same argument can be made for the number of positive eigenvalues and therefore must have the same number of zero eigenvalues. □

The next theorem gives the inertia of $Q(\lambda)$.

Theorem 2.2.2 ([15]). *Denote the $2n$ eigenvalues of an HQEP of order n by $\lambda_i^{\pm}$ and arrange in order such that:*

$$\underbrace{\lambda_1^- \leq \ldots \leq \lambda_n^-}_{Secondary} < \underbrace{\lambda_1^+ \leq \ldots \leq \lambda_n^+}_{Primary}. \tag{2.5}$$

Then:

(a) $Q(\lambda) \prec 0 \ \forall \ \lambda \in (\lambda_n^-, \lambda_1^+)$;

(b) $Q(\lambda) \succ 0 \ \forall \ \lambda \in (-\infty, \lambda_1^-) \cup (\lambda_n^+, +\infty)$;

(c) Inertia($Q(\lambda)$)=$(n-k, 0, k)$ for $\lambda \in (\lambda_k^+, \lambda_{k+1}^+)$ or $\lambda \in (\lambda_{n-k}^-, \lambda_{n-k+1}^-)$ for $k = 1, \ldots, n$.

The interval $(\lambda_n^-, \lambda_1^+)$ is the ***gap*** between the principal and the secondary quadratic eigenvalues.

2.3 Decomposition of HQEP

Since HQEP is quadratic, we may be able to factor it into two linear eigenvalue problems the same as in quadratic polynomials in one variable. Indeed, $Q(\lambda)$ can be factorized.

Definition 2.3.1. A matrix $U \in \mathbb{R}^{n\times n}$ is a *right (left) solvent* of $Q(\lambda)$ if $Q(U) = 0$ ($Q^T(U) = 0$).

Theorem 2.3.1. *$Q(\lambda)$ is right divisible by $I\lambda - U$ if and only if U is a right solvent of $Q(\lambda)$.*

Proof. $Q(\lambda)$ is right divisible by $I\lambda - U \implies Q(\lambda) = D(\lambda)(I\lambda - U) \implies Q(U) = 0$. On the other hand,

$$\begin{aligned} Q(U) = 0 \implies Q(\lambda) &= Q(\lambda) - Q(U) \\ &= \lambda^2 A + \lambda B + C - (AU^2 + BU + C) \\ &= A(\lambda^2 I - U^2) + B(\lambda I - U) \\ &= A(\lambda I + U)(\lambda I - U) + B(\lambda I - U) \\ &= [A(\lambda I + U) + B](\lambda I - U) \\ &= (\lambda A + AU + B)(\lambda I - U), \end{aligned}$$

implying $Q(\lambda)$ is right divisible by $I\lambda - U$. □

Note that $Q^T(\lambda) = (I\lambda - U^T)D^T(\lambda)$ and hence if U is a right solvent of $Q(\lambda)$, then U^T is a left solvent of $Q(\lambda)$.

Theorem 2.3.2. *Let $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ and $X = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix}$ where (λ_i, x_i) for $i = 1, \ldots, n$, is a quadratic eigenpair of the HQEP of order n for $Q(\lambda)$, and X is non-singular. Then the matrix $S = X\Lambda X^{-1}$ is a right solvent of $Q(\lambda)$.*

Proof. We have

$$\begin{aligned} Q(S) &= AS^2 + BS + C \\ &= A(X\Lambda X^{-1})^2 + B(X\Lambda X^{-1}) + C \\ &= AX\Lambda^2 X^{-1} + BX\Lambda X^{-1} + CXX^{-1} \\ &= (AX\Lambda^2 + BX\Lambda + CX)X^{-1} \\ &= \Big(\begin{bmatrix} Ax_1 & Ax_2 & \ldots & Ax_n \end{bmatrix} \Lambda^2 + \begin{bmatrix} Bx_1 & Bx_2 & \ldots & Bx_n \end{bmatrix} \Lambda \end{aligned}$$

$$
\begin{aligned}
&+ \begin{bmatrix} Cx_1 & Cx_2 & \dots & Cx_n \end{bmatrix}\Big) X^{-1} \\
&= \Big(\begin{bmatrix} \lambda_1^2 A x_1 & \lambda_2^2 A x_2 & \dots & \lambda_n^2 A x_n \end{bmatrix} + \begin{bmatrix} \lambda_1 B x_1 & \lambda_2 B x_2 & \dots & \lambda_n B x_n \end{bmatrix} \\
&+ \begin{bmatrix} Cx_1 & Cx_2 & \dots & Cx_n \end{bmatrix}\Big) X^{-1} \\
&= \begin{bmatrix} Q(\lambda_1)x_1 & Q(\lambda_2)x_2 & \dots & Q(\lambda_n)x_n \end{bmatrix} X^{-1} = 0.
\end{aligned}
$$

Hence S as defined is a solvent of $Q(\lambda)$. □

There are in fact two very special solvents namely the ***primary solvent*** and the ***secondary solvent***.

Definition 2.3.2 (Primary Solvent)**.** Let $\Lambda_+ = \mathrm{diag}(\lambda_1^+, \lambda_2^+, \dots, \lambda_n^+)$ and $X_+ = \begin{bmatrix} \overset{+}{x}_1 & \overset{+}{x}_2 & \dots & \overset{+}{x}_n \end{bmatrix}$ where $\overset{+}{x}_i$ are the quadratic eigenvectors respectively. Then the matrix $U = X_+\Lambda_+X_+^{-1}$ is called the *primary solvent* of $Q(\lambda)$.

Definition 2.3.3 (Secondary Solvent)**.** Let $\Lambda_- = \mathrm{diag}(\lambda_1^-, \lambda_2^-, \dots, \lambda_n^-)$ and $X_- = \begin{bmatrix} \overset{-}{x}_1 & \overset{-}{x}_2 & \dots & \overset{-}{x}_n \end{bmatrix}$ where $\overset{-}{x}_i$ are the quadratic eigenvectors respectively. Then the matrix $V = X_-\Lambda_-X_-^{-1}$ is called the *secondary solvent* of $Q(\lambda)$.

Theorem 2.3.3 ([15])**.** *The matrices X_+ and X_- are non-singular.*

Theorem 2.3.4. *If U is a right solvent of $Q(\lambda)$, then $V = A^{-1}U^{-T}C$ is also a right solvent of $Q(\lambda)$, where U^{-T} is the inverse transpose of U.*

Proof. We have

$$AU^2 + BU + C = 0 \implies A^{-1}C = -U^2 - A^{-1}BU \implies CA^{-1} = -(U^T)^2 - U^TBA^{-1}.$$

Thus

$$
\begin{aligned}
Q(V) &= AV^2 + BV + C \\
&= AA^{-1}U^{-T}CA^{-1}U^{-T}C + BA^{-1}U^{-T}C + C \\
&= U^{-T}CA^{-1}U^{-T}C + BA^{-1}U^{-T}C + C
\end{aligned}
$$

$$\begin{aligned} &= U^{-T}(-(U^T)^2 - U^T BA^{-1})U^{-T}C + BA^{-1}U^{-T}C + C \\ &= -C - BA^{-1}U^{-T}C + BA^{-1}U^{-T}C + C = 0, \end{aligned}$$

as was to be shown. $\square$

Theorem 2.3.5 (***Decomposition of*** $Q(\lambda)$[16])**.** *If matrix* U *is the right solvent of* $Q(\lambda)$, *then*

$$Q(\lambda) = (I\lambda - V^T)A(I\lambda - U), \tag{2.6}$$

where $V^T = CU^{-1}A^{-1}$.

Proof. From Theorem 2.3.1 We have

$$\begin{aligned} Q(\lambda) &= (\lambda A + AU + B)(\lambda I - U) \\ &= (\lambda A - CU^{-1})(\lambda I - U) \\ &= (\lambda I - CU^{-1}A^{-1})A(\lambda I - U), \end{aligned}$$

as expected. $\square$

Example 2.3.1. Consider the HQEP from Example 2.1.1. We have $\Lambda_+ = \text{diag}(1.2117, 1.3772, 6.6104)$, $\Lambda_- = \text{diag}(-1.8856, -1.0644, -0.1242)$,

$$X_+ = \begin{bmatrix} -0.2327 & -0.7718 & -0.3960 \\ -0.6939 & -0.2056 & 0.7646 \\ -0.6814 & 0.6017 & -0.5084 \end{bmatrix}, \text{ and } X_- = \begin{bmatrix} -0.7422 & 0.5926 & 0.4079 \\ 0.5935 & -0.1087 & 0.6795 \\ 0.3115 & 0.7981 & -0.6098 \end{bmatrix}.$$

Now we can construct the primary and secondary solvents for the HQEP (shown with up to 4 decimal places):

$$U = \begin{bmatrix} 2.5448 & -1.4359 & 1.0070 \\ -2.3232 & 4.0278 & -2.0745 \\ 1.4759 & -1.8844 & 2.6266 \end{bmatrix}, \text{ and } V = \begin{bmatrix} -1.1705 & 0.7961 & 0.1871 \\ 0.6179 & -0.8849 & -0.4344 \\ -0.1880 & -0.6900 & -1.0188 \end{bmatrix}.$$

We get $\|AU^2 + BU + C\|_2 = 8.3061e - 14$, and $\|AV^2 + BV + C\|_2 = 1.8178e - 14$, which implies that U and V are solvents of the HQEP.

2.4 Hyperbolicity of HQEP

The hyperbolicity of a QEP is only a ***sufficient*** condition that guarantees all the quadratic eigenvalues are real. In the next example we will see a QEP that is not hyperbolic but still have all real quadratic eigenvalues.

Example 2.4.1. Consider the QEP with

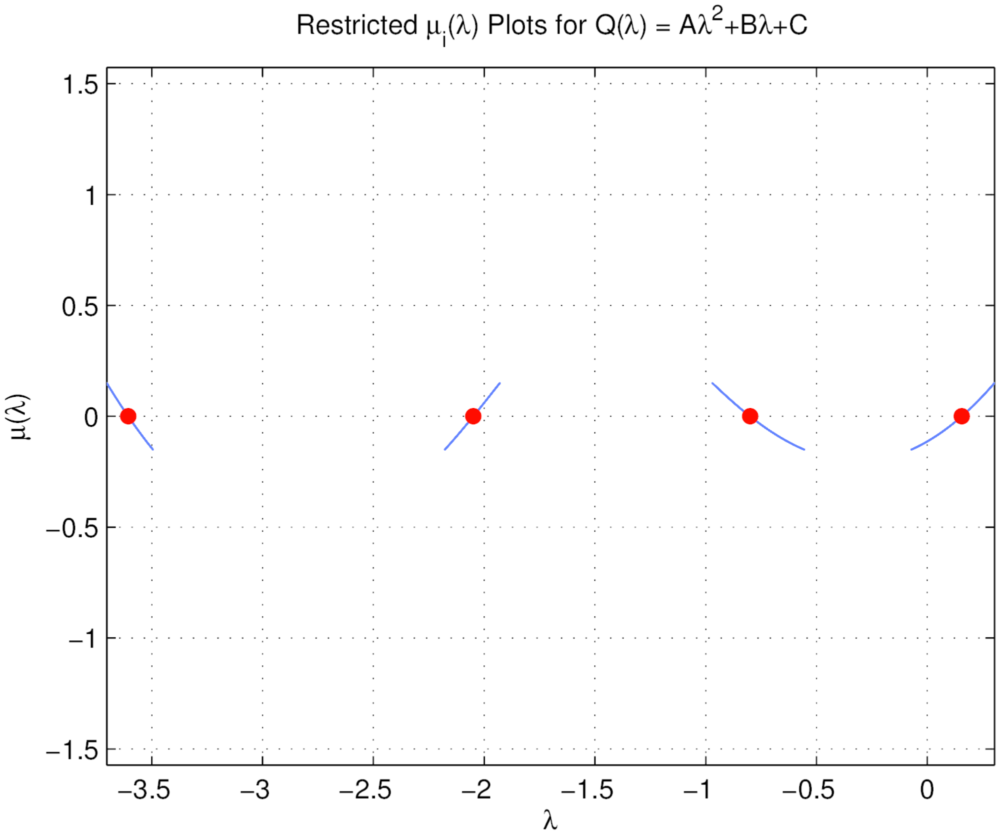
$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0.5 & 0 \\ 0 & 5.8 \end{bmatrix}, C = \begin{bmatrix} 0.01 & 1 \\ 1 & 8 \end{bmatrix}.$$

Matlab `polyeig`(C, B, A) reveals all the quadratic eigenvalues are real:

$$\lambda_1 = -3.6065, \quad \lambda_2 = -2.0491, \quad \lambda_3 = -0.8000, \quad \lambda_4 = 0.1556$$

ordered as in (2.5). This QEP is not hyperbolic which can easily be tested by checking that $Q(\frac{\lambda_2+\lambda_3}{2}) \succ 0$ which violates the condition of Theorem 2.2.2 (see Figure 2.2). If we label the quadratic eigenvalues of such QEP using Definition 2.1.1 then we will get $\lambda_2^- > \lambda_1^+$ which also contradicts Theorem 2.2.2.

Figure 2.2. Plot of Eigenvalue Function $\mu(\lambda)$ of a non-hyperbolic QEP.



In Example 2.4.1, all the quadratic eigenvalues were real because $(x^TBx)^2 > 4(x^TAx)(x^TCx)$, for any quadratic eigenvector x of the QEP. On the other hand, hyperbolicity of a QEP requires the condition that $\forall\ 0 \neq x \in \mathbb{C}^n$ and hence hyperbolicity is more restrictive.

We can test for the hyperbolicity of a given QEP by using the sufficient condition

$$(\lambda_B^{\min})^2 > 4\lambda_A^{\max}\lambda_C^{\max}, \tag{2.7}$$

where $\lambda_M^{\max}$ ($\lambda_M^{\min}$) is the biggest (smallest) absolute values of the eigenvalues of a given symmetric matrix M. But this condition will not catch all the HQEP problems because it's more restrictive than the hyperbolicity condition of (1.2), as will be seen in the next two examples.

Example 2.4.2. Consider the HQEP with

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 5.5 & 0 \\ 0 & 33 \end{bmatrix}, C = \begin{bmatrix} \frac{1}{2} & 1 \\ 1 & 7 \end{bmatrix},$$

with $\lambda_C^{\max} = 7.1504$ and $\lambda_B^{\min} = 5.5$. It is straightforward using (2.7) to conclude that it is an HQEP.

But on the other hand

Example 2.4.3 (low degree of hyperbolicity HQEP)**.** Consider the HQEP with

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 2 & 0 \\ 0 & 12 \end{bmatrix}, C = \begin{bmatrix} \frac{1}{2} & 1 \\ 1 & 7 \end{bmatrix},$$

which has quadratic eigenvalues:

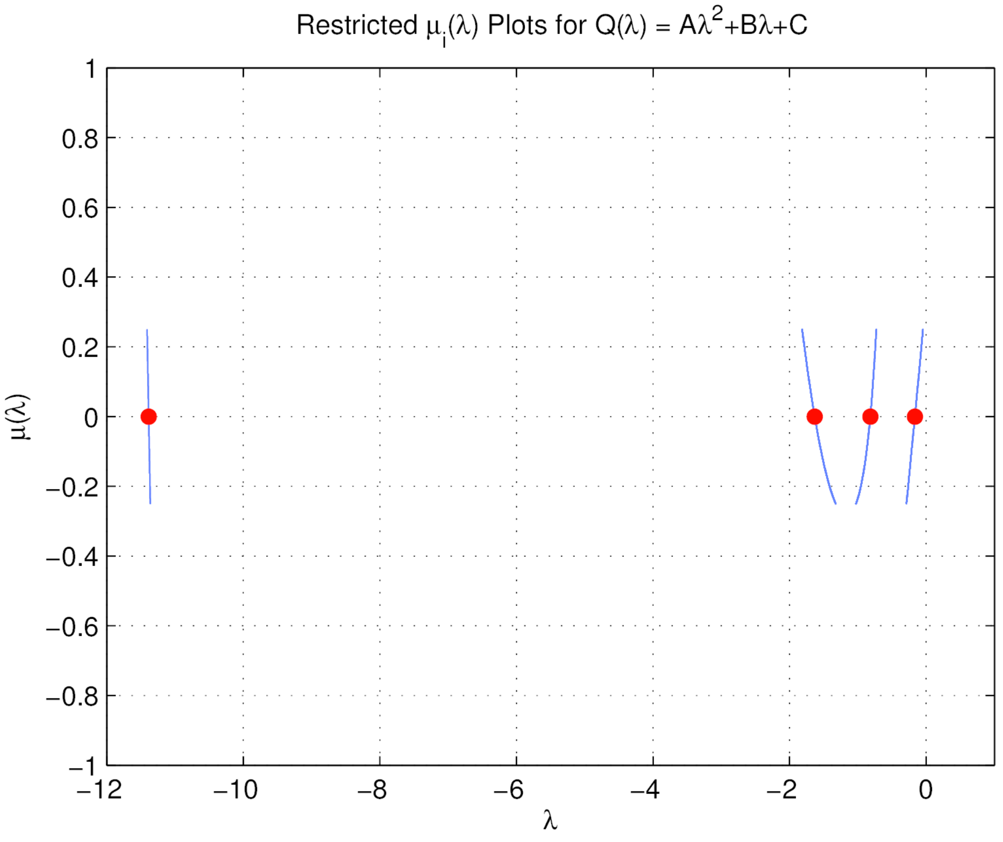
$$\lambda_1^- = -11.3860, \quad \lambda_2^- = -1.6318, \quad \lambda_1^+ = -0.8176, \quad \lambda_2^+ = -0.1646$$

and $Q(\frac{\lambda_2^- + \lambda_1^+}{2}) \prec 0$. But in this case we have $\lambda_C^{\max} = 7.1504$, $\lambda_B^{\min} = 2$ and hence

$$(\lambda_B^{\min})^2 \not> 4\lambda_A^{\max}\lambda_C^{\max}. \tag{2.8}$$

This HQEP failed (2.7) (see Figure 2.3).

Figure 2.3. Plot of Eigenvalue Function $\mu(\lambda)$ of HQEP (low degree of Hyperbolicity).



We now present four procedure to test if a given QEP is hyperbolic:

1. If the QEP pass the sufficient condition (2.7) then it is hyperbolic.
2. Solve the one-dimensional global optimization problem $\min_{\mu}(\max(\lambda_{Q(\mu)}))$, where $\lambda_{Q(\mu)}$ is the set of all eigenvalues of the symmetric matrix $Q(\mu)$. If it is negative then it is hyperbolic.
3. Compute all $2n$ quadratic eigenvalues of the QEP. If all quadratic eigenvalues are real, and $Q(\frac{\lambda_n^- + \lambda_1^+}{2}) \prec 0$ then it is hyperbolic [1].
4. If $L(\lambda)$ in (2.2) is a definite pencil then it is hyperbolic [11].

However, none of these tests is cheap and definitive at the same time. Later we will develop an algorithm that can test for the hyperbolicity of a QEP to a high degree of accuracy using the bisection method without computing any quadratic eigenvalues.

2.5 Overdamped HQEP

Definition 2.5.1. An HQEP is overdamped if $B \succ 0, C \succeq 0$ [1].

An HQEP becomes an overdamped HQEP after the proper shift in λ [17, 1],

$$\begin{aligned} Q(\lambda+\theta) &= \lambda^2 A + \lambda(B + 2\theta A) + (C + \theta B + \theta^2 A) \\ &= \lambda^2 \tilde{A} + \lambda \tilde{B} + \tilde{C} \\ &=: \tilde{Q}(\lambda). \end{aligned}$$

By choosing a large enough shift θ, we will have

$$\tilde{B} = B + 2\theta A \succ 0, \qquad \tilde{C} = C + \theta B + \theta^2 A \succeq 0. \tag{2.9}$$

Theorem 2.5.1. *$Q(\lambda)$ is overdamped $\iff$ $\lambda_n^+ \leq 0$.*

Proof. Let $M_x \equiv x^T M x$ for any symmetric matrix M and a nonzero vector x.

($\Longrightarrow$) Assume $Q(\lambda)$ is overdamped, i.e. , $B \succ 0, C \succeq 0$. Then we have

$$\frac{-B_x - \sqrt{B_x^2 - 4A_xC_x}}{2A_x} < 0.$$

and

$$\begin{aligned} -4A_xC_x \leq 0 &\implies B_x^2 - 4A_xC_x \leq B_x^2 \\ &\implies \sqrt{B_x^2 - 4A_xC_x} \leq (B_x) \\ &\implies -B_x + \sqrt{B_x^2 - 4A_xC_x} \leq 0 \\ &\implies \frac{-B_x + \sqrt{B_x^2 - 4A_xC_x}}{2A_x} \leq 0. \end{aligned}$$

Hence

$$\frac{-B_x \pm \sqrt{B_x^2 - 4A_xC_x}}{2A_x} \leq 0.$$

($\Longleftarrow$) Since $\lambda_n^+ \leq 0$, $Q(0) = C \succeq 0$. We also have

$$\begin{aligned} \lambda_n^+ \leq 0 \quad &\Longrightarrow \quad -B_x + \sqrt{B_x^2 - 4A_xC_x} \leq 0 \\ &\Longrightarrow \quad B_x \geq \sqrt{B_x^2 - 4A_xC_x} > 0 \\ &\Longrightarrow \quad B_x > 0. \end{aligned}$$

i.e., $B \succ 0$. $\square$

2.6 Solving the HQEP

In this section, we survey methods and techniques to solve HQEPs and QEPs in general.

2.6.1 Linearization

The HQEP in (1.1) can be solved by solving $L(\lambda)$ as defined in (2.2) using the QZ algorithm. Also by applying iterative algorithms like the subspace iteration, the Arnoldi method, and the unsymmetric Lanczos method to the matrix $M^{-1}N$ from (2.2) we may solve the underlying HQEP [18].

The same can be done with any equivalent generalized eigenvalue problem which has the same eigenvalues as (1.1), for example

$$\lambda \begin{bmatrix} I_n & 0 \\ 0 & A \end{bmatrix} - \begin{bmatrix} 0 & I_n \\ -C & -B \end{bmatrix}. \tag{2.10}$$

The HQEP can also be converted into the companion block-matrix

$$\mathcal{C} = \begin{bmatrix} -A^{-1}B & -A^{-1}C \\ I_n & 0 \end{bmatrix} \tag{2.11}$$

which is analogous to changing a high-order ODE to a system of first-order linear ODEs. Applying the QR algorithm to such matrix should reveal the spectrum of the HQEP.

Solving $L(\lambda)$ has the advantage of keeping the symmetry of the problem which is destroyed by the other two approaches, but $L(\lambda)$ may destroy the banded structure of the HQEP which we will take advantage of in our algorithm later. Also solving (2.11) may lead to loss of accuracy and to instability if A is ill-conditioned .

2.6.2 Finding a Solvent Matrix

We can find a solvent matrix S as defined in Theorem 2.3.2 using an efficient algorithm introduced in [2] based on the Block Cyclic Reduction [1] that recursively finds a solvent matrix of a given HQEP. The algorithm should converge to the primary solvent matrix U if the HQEP is overdamped according to Lemma 6 in [1]. In the case of a not overdamped HQEP, the algorithm will converge to a solvent matrix S which has a subset spectrum of the HQEP. In both cases, another solvent can be found using the relation in Theorem 2.3.4 which has the rest of the HQEP's spectrum.

Algorithm 2.1 Iteratively find a Solvent for an HQEP using BCR [1, 2]

Given matrices A, B, C representing an HQEP, the procedure will return a solvent matrix S.

```
function ReturnSolvent($A, B, C, \epsilon$)
    $\hat{X}_0 \leftarrow 0$
    $\hat{X} \leftarrow B$
    $\hat{A} \leftarrow A$
    $\hat{B} \leftarrow B$
    $\hat{C} \leftarrow C$
    while $\left\|\hat{X}_0 - \hat{X}\right\|_2 < \epsilon$ do
        $\hat{X} \leftarrow \hat{X} - \hat{A}\hat{B}^{-1}\hat{C}$
        $\hat{A}_0 \leftarrow \hat{A}$
        $\hat{A} \leftarrow \hat{A}\hat{B}^{-1}\hat{A}$
        $\hat{B}_0 \leftarrow \hat{B}$
        $\hat{B} \leftarrow \hat{B} - \hat{A}_0\hat{B}^{-1}\hat{C} - \hat{C}\hat{B}^{-1}\hat{A}_0$
        $\hat{C}_0 \leftarrow \hat{C}\hat{B}_0^{-1}\hat{C}$
    end while
    return $-\hat{X}^{-1}C$
end function
```

Example 2.6.1. Applying Algorithm 2.1 to the HQEP in Example 2.4.3 gives us the primary solvent

$$U = \begin{bmatrix} -0.3590 & -0.9825 \\ -0.0908 & -0.6231 \end{bmatrix},$$

which has the eigenvalues -0.1646 and -0.8176. Then we can get the secondary solvent

$$V = A^{-1}U^{-T}C = \begin{bmatrix} -1.6410 & 0.0908 \\ 0.9825 & -11.3769 \end{bmatrix},$$

which has the eigenvalues -11.3860 and -1.6318, agreeing with Matlab's `polyeig` used before.

In the previous example the HQEP was overdamped and the algorithm converges to the primary solver. In the next example we will see that the algorithm will converge to a solvent that is not the primary solvent because the HQEP is not overdamped.

Example 2.6.2. Applying Algorithm 2.1 to the HQEP in Example 2.1.1 gives us a solvent

$$X_1 = \begin{bmatrix} -1.0813 & 0.6674 & 0.1034 \\ -0.5231 & 0.7615 & 0.6370 \\ -1.6380 & 1.4025 & 0.3428 \end{bmatrix},$$

which has the eigenvalues -1.0644, -0.1242 and 1.2117. Then we can get another solvent

$$X_2 = A^{-1}X_1^{-T}C = \begin{bmatrix} 6.8030 & 6.1165 & 9.0490 \\ -7.4281 & -5.0263 & -11.7155 \\ 2.3754 & -0.2891 & 4.3254 \end{bmatrix},$$

which has the eigenvalues -1.8856, 1.3772 and 6.6104. The union of the spectrums of X_1 and X_2 will give us the whole spectrum of the HQEP as in Example 2.1.1.

2.6.3 Iterative methods with Deflation for a Sparse HQEP

The HQEP as defined in Definition 1.1.1 can be transformed to the linear symmetric eigenvalue problem

$$\beta(\mu)Cx = -(A\mu + B)x \tag{2.12}$$

where $\beta(\mu)$ is an eigenvalue of the matrix pencil $-(A\mu + B) - \lambda C$ for any given μ. It was shown in [18] that $\beta_j(\mu)$, for $j = 1, \cdots, n$, are strictly decreasing. Hence the smallest positive eigenvalue of the HQEP is the smallest positive fixed point of the function $1/\beta_1(\mu)$. In [18], they also introduced three algorithms to compute the smallest positive quadratic eigenvalue namely, Basic, Tangent, and Newton's iterative methods. All the methods use the symmetric Lanczos algorithm to compute the eigenpair in every iteration. Afterwards, a nonequivalence deflation technique is used to produce a new HQEP that has the same spectrum as the original one except that the smallest positive eigenvalue of the original problem becomes zero in the deflated problem. Then they find the smallest positive eigenvalue of the deflated problem and deflate again and so on. This approach can also be used to find all the negative quadratic eigenvalues. The algorithm will converge linearly globally and quadratically locally.

2.6.4 Solving the Characteristic Equation of $Q(\lambda)$

The zeros of the characteristic polynomial of an HQEP are exactly its eigenvalues. **Laguerre's method** and **Ehrlich-Aberth method** can approxiamte the roots of the characteristic polynomial of $Q(\lambda)$ but they require stable and efficient computation of Inertia($Q(\lambda)$), $f_Q(\lambda)$, $f'_Q(\lambda)/f_Q(\lambda)$, and $f''_Q(\lambda)/f_Q(\lambda)$, where $f_Q(\lambda) = \det(Q(\lambda))$ [19]. Although these methods may converge faster than the bisection method, they can only be applied to tri-diagonal HQEPs, and may not be efficient for banded HQEPs beyond tri-diagonal.

CHAPTER 3

Bisection for HQEP and Hyperbolicity Test

3.1 The Symmetric Eigenvalue Problem case

In this section we will introduce the algorithms in [3, pp.228-232] which we will use latter in the thesis when dealing with HQEP but with some modifications. Let A be a square symmetric matrix of size n with simple eigenvalues λ_i, and corresponding eigenvector x_i for $i = 1, 2, ..., n$. Then from Theorem 1.3.1 to find the eigenvalues of A is equivalent to finding the eigenvalues of the matrix polynomial of degree 1:

$$P(\lambda) = A - \lambda I.$$

This means we will have n $\mu_i(\lambda)$ curves that we need to find their roots assuming all eigenvalues of the matrix A are simple. We know that the matrix A has at most n eigenvalues and hence we know that each $\mu_i(\lambda)$ curve will have only one real root. Now since we need to find the roots of n $\mu_i(\lambda)$ curves, we need to have a mechanism to show if a given $\mu_i(\lambda)$ curve changes sign between 2 different values of λ and therefore a root exist between those 2 values. Using Sylvester's Inertia Theorem we can get the inertia of $P(\lambda)$ at different values of λ. If there is a change in say the number of the negative eigenvalues of $P(\lambda)$ between those two values, then we can deduce that there are some roots for some $\mu_i(\lambda)$ curves within those two values, and the number of roots is equal to the difference in the number of negative eigenvalues.

One of the readily available matrix decompositions for symmetric matrices and which gives us the inertia is the LDL^T decomposition at a cost of $\mathcal{O}(n^3)$ using Matlab's `ldl`. The procedure `Negcount` is a procedure that returns the number of eigenvalues less than λ of a symmetric matrix A as a function in λ. Note that

Matlab's `ldl` on line 4 of Algorithm 3.1 returns a block diagonal matrix D which has 1×1 and 2×2 blocks on its diagonal. In case of 2×2 blocks, we need to compute the eigenvalues of the 2×2 blocks. In all our examples, D is a diagonal matrix and computing its inertia is trivial.

Algorithm 3.1 Return the number of eigenvalues less than λ of a symmetric matrix A

```
1: function NEGCOUNT($A, \lambda$)
2:     $P(\lambda) \leftarrow A - \lambda I$
3:     $num \leftarrow 0$
4:     $D \leftarrow$ ldl($P(\lambda)$)
5:     for $i = 1$ to $n$ do
6:         if $D(i,i) < 0$ then
7:             $num \leftarrow num + 1$
8:         end if
9:     end for
10:    return $num$
11: end function
```

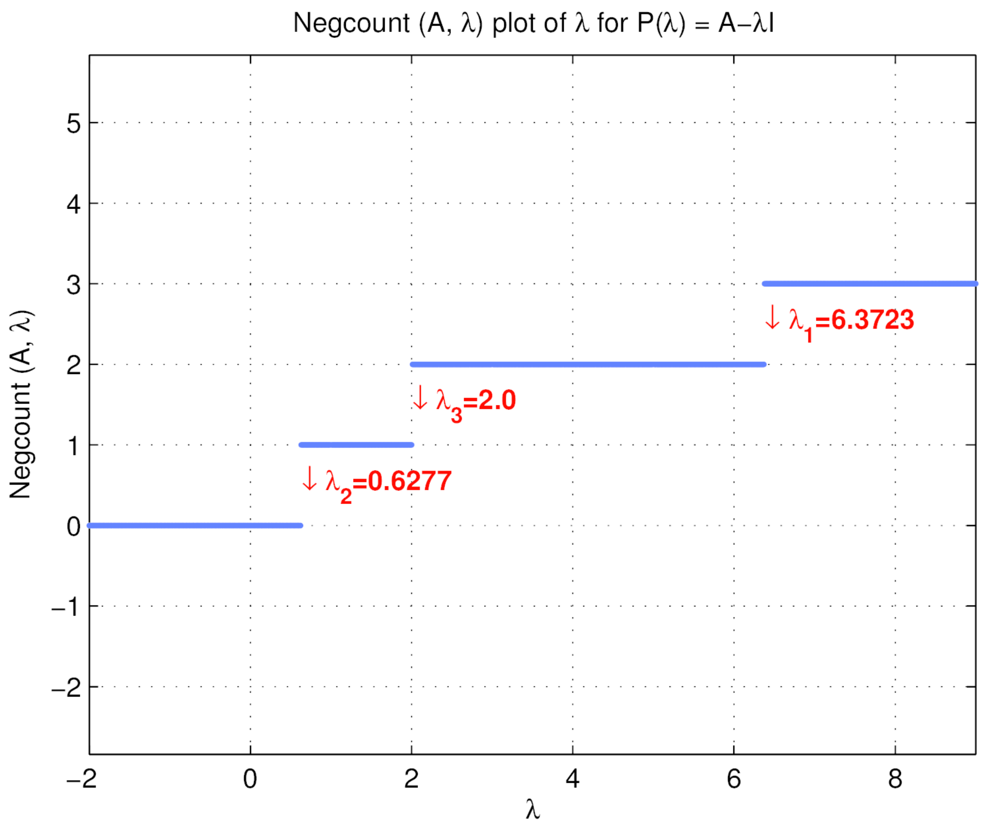


Figure 3.1. Plot of the number of eigenvalues less than λ of a symmetric matrix A as a step function.

Figure 3.1 shows the eigenvalues of the matrix

$$A = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix},$$

which are (0.6277, 2.0000, 6.3723), and note the change of `Negcount` as λ varies.

3.2 Fast Inertia Calculation

`Negcount` will help us decide if a section of the interval where we want to find the eigenvalues has any eigenvalues or none (see Figure 3.1). The issue with such algorithm is that we will need to apply `Negcount` a factor of n times making the total cost of the procedure $\mathcal{O}(n^4)$ flops. This is a factor of n times more expensive than the QR algorithm which has a cost of only $\mathcal{O}(n^3)$ flops. But if we compute the tri-diagonal (Hessenberg) form of A to get T only once using Householder's transformations at a cost of $\mathcal{O}(n^3)$, we get $\widetilde{P}(\lambda) = T - \lambda I$ which has the same spectrum as $P(\lambda)$:

$$\begin{aligned} \widetilde{P}(\lambda) &= \begin{bmatrix} a_1-\lambda & b_1 & & \\ b_1 & a_2-\lambda & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n-\lambda \end{bmatrix} = LD(\lambda)L^T \\ &= L \begin{bmatrix} d_1(\lambda) & & & \\ & d_2(\lambda) & & \\ & & \ddots & \\ & & & d_n(\lambda) \end{bmatrix} L^T \end{aligned} \tag{3.1}$$

[3, p.230], where

$$d_1(\lambda) = a_1 - \lambda, \quad d_i(\lambda) = (a_i - \lambda) - \frac{b_{i-1}^2}{d_{i-1}} \quad i = 2, ..., n. \tag{3.2}$$

Now we have a linear recursion of the diagonal entries of the matrix $D(\lambda)$ and therefore `Negcount` will only cost $\mathcal{O}(n)$. We will call the new procedure `FastNegcount` which takes T (the tri-diagonal form of A) and λ as inputs and outputs the same values as the original `Negcount` procedure.

Algorithm 3.2 Return the number of eigenvalues less than λ of a tri-diagonal symmetric matrix T in linear time

```
function FastNegcount($T, \lambda$)
    $num \leftarrow 0$
    $d \leftarrow T(1,1) - \lambda$
    if $d < 0$ then
        $num \leftarrow num + 1$
    end if
    for $i = 2$ to $n$ do
        $d \leftarrow T(1,i) - \lambda - \frac{T(2,i-1)^2}{d}$
        if $d < 0$ then
            $num \leftarrow num + 1$
        end if
    end for
    return $num$
end function
```

Algorithm 3.2 which is based on (3.2) was shown to be stable without any pivoting in Lemma 5.3 in [3]. We will end this section by implementing the procedure `Bisect`. The idea is to bisect the interval on-hand and to discard any sections where the procedure `FastNegcount` doesn't change, and keep bisecting sections that have one or more eigenvalues. Until we get n sections with only one eigenvalue in each and have a length within some tolerance. At that moment we may choose the midpoint of a section as an approximation to the eigenvalue we need to find.

Algorithm 3.3 Return the approximated eigenvalues of a symmetric matrix A using Bisection

Given a symmetric matrix A, the procedure will return all the eigenvalues of A in the interval (a, b) as a vector Eig in a descending order.

```
function BISECT($A, a, b, tol$)
    Declare stack $Interval$
    Declare queue $Eig$
    $T \leftarrow$ ToTriDiag($A$)
    put.$Interval(a, b)$
    while $Interval$ not empty do
        $(a, b) \leftarrow$ pop from $Interval$
        $c \leftarrow$ FastNegcount$(T, b) -$ FastNegcount$(T, a)$
        if $c > 1$ or $(c = 1$ and $b - a > tol * \text{abs}(\frac{a+b}{2}))$ then
            put $(a, \frac{a+b}{2})$ into $Interval$
            put $(\frac{a+b}{2}, b)$ into $Interval$
        else if $c = 1$ and $b - a < tol * \text{abs}(\frac{a+b}{2})$ then
            put $(\frac{a+b}{2})$ into $Eig$
        end if
    end while
    return $Eig$
end function
```

3.3 Analysis of the HQEP case

We will begin by introducing the procedure `NegcountHQEP` that counts the number of negative eigenvalues of the symmetric matrix $Q(\lambda)$ for a given fixed λ.

Algorithm 3.4 Return the number of negative eigenvalues of $Q(\lambda)$ for a specific λ

```
1: function NegcountHQEP($A, B, C, \lambda$)
2:     $Q(\lambda) \leftarrow A\lambda^2 + B\lambda + C$
3:     $num \leftarrow 0$
4:     $D \leftarrow$ ldl($Q(\lambda)$)
5:     for $i = 1$ to $n$ do
6:         if $D(i,i) < 0$ then
7:             $num \leftarrow num + 1$
8:         end if
9:     end for
10:    return $num$
11: end function
```

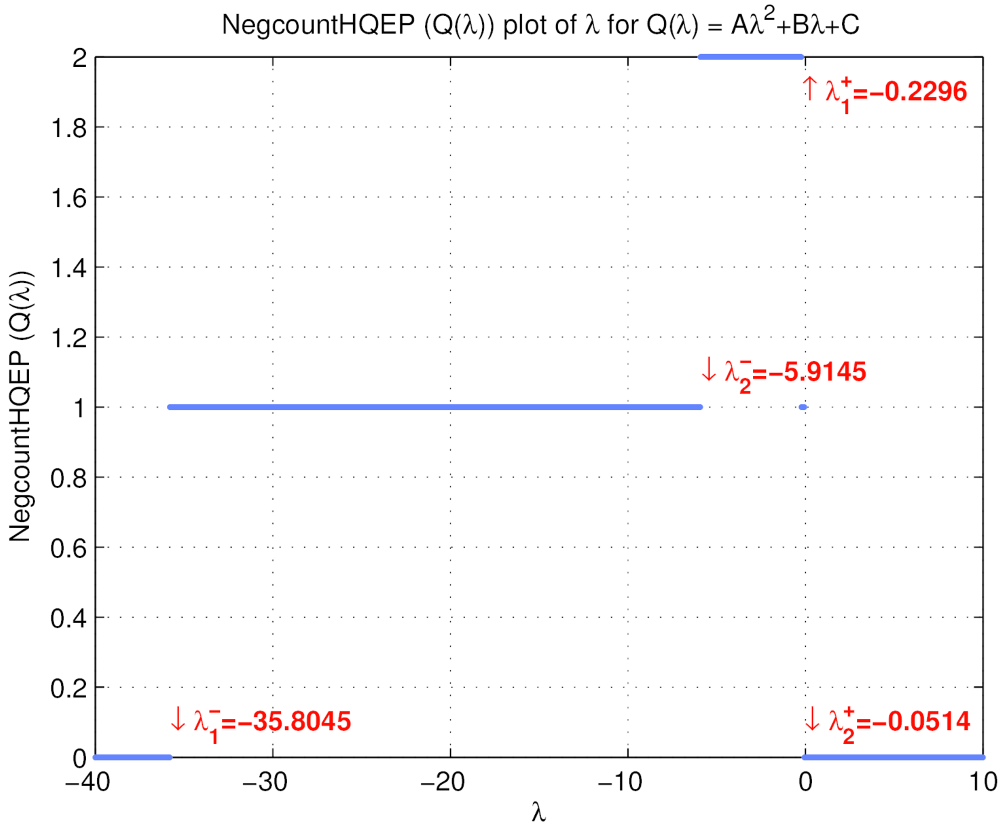
Now using Theorem 2.2.2 and the algorithm `NegcountHQEP` we may be able to find the eigenvalues of the HQEP. The `Bisect` algorithm won't succeed directly in this case, and the reason is that `NegcountHQEP`($Q(\lambda)$) is not monotonic with λ.

Example 3.3.1. Consider the HQEP with

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 6 & 0 \\ 0 & 36 \end{bmatrix}, C = \begin{bmatrix} \frac{1}{2} & 1 \\ 1 & 7 \end{bmatrix}.$$

Matlab's `polyeig`(C, B, A) reveals the quadratic eigenvalues of the HQEP: $(-35.8045, -5.9145, -0.2296, -0.0514)$ ordered as in (2.5) (see Figure 3.2).

Figure 3.2. Plot of the number of negative eigenvalues of $Q(\lambda)$ as a step function.



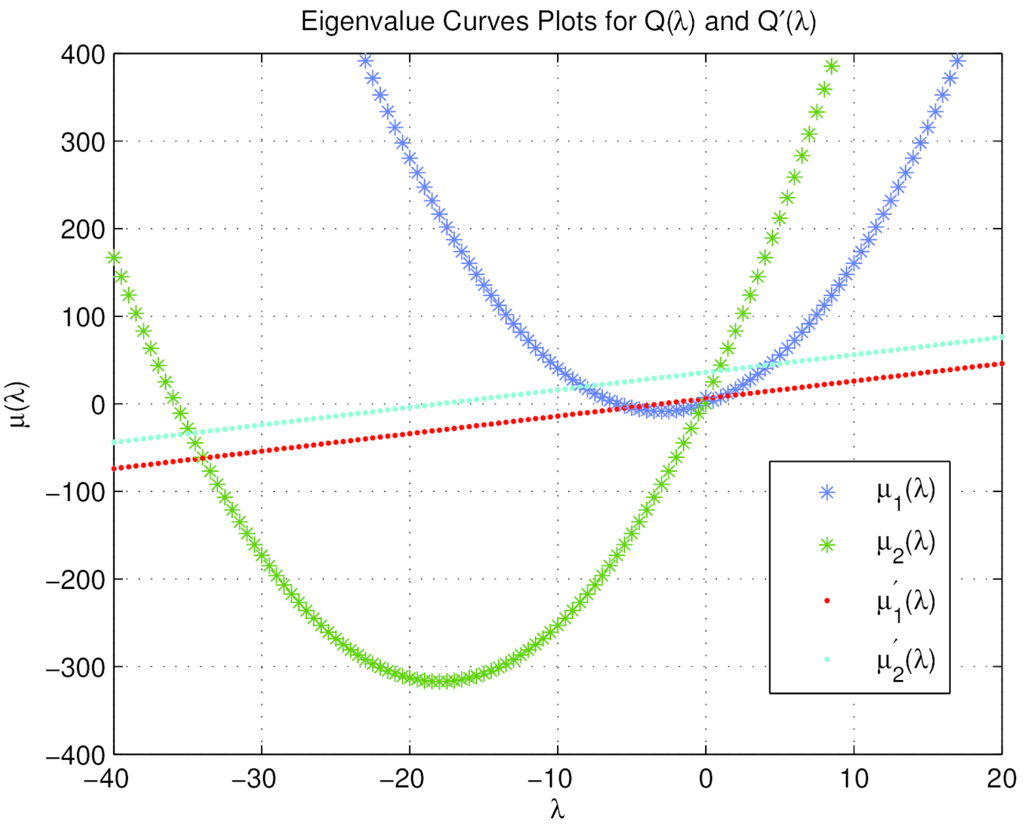
From Figure 3.2 we can see that `NegcountHQEP`$(Q(\lambda)) = 0$ when $\lambda < \lambda_1^-$ or $\lambda > \lambda_n^+$. Then it increases from λ_1^- to λ_n^-. Between λ_n^- and λ_1^+ `NegcountHQEP`$(Q(\lambda)) = n$, it decreases from λ_1^+ to λ_n^+ to 0. For example, if $\lambda_\alpha < \lambda_1^-$ and $\lambda_\beta > \lambda_n^+$ then `NegcountHQEP`$(Q(\lambda_\beta)) =$`NegcountHQEP`$(Q(\lambda_\alpha)) = 0$ and the `Bisect` algorithm will discard the interval $[\lambda_\alpha,\lambda_\beta]$ thinking that there are no eigenvalues in that interval even though all the $2n$ eigenvalues of the HQEP are in that interval.

We may be able to avoid such difficulty if we can partition the interval $[\lambda_\alpha,\lambda_\beta]$ into $[\lambda_\alpha,\lambda_c]$ and $[\lambda_c,\lambda_\beta]$ where $\lambda_n^- < \lambda_c < \lambda_1^+$, then apply the `Bisect` algorithm to each partition separately. Later we introduce the procedure `CenterHQEP` to find λ_c.

It will use Negcount with bisection and is guaranteed to converge if $[\lambda_n^-, \lambda_1^+] \cap [a, b]$ is not empty, otherwise it may not converge. Therefore, we need to be able to find an interval for `CenterHQEP` to converge.

Example 3.3.2. Consider again Example 3.3.1. Here we will see an example where the eigenvalue curves $\mu(\lambda)$ of $Q(\lambda)$ are parabolic-like curves facing up, while the eigenvalue curves $\mu'(\lambda)$ of $Q'(\lambda)$ are nearly linear.

Figure 3.3. Eigenvalue curves of $Q(\lambda)$ and $Q'(\lambda)$.



In fact, if we apply Theorem 1.3.4 to $Q(\lambda) = A\lambda^2 + B\lambda + C$ we get

$$\mu'(\lambda) = u^T(\lambda)Q'(\lambda)u(\lambda) = 2\lambda u^T(\lambda)Au(\lambda) + u^T(\lambda)Bu(\lambda), \tag{3.3}$$

and if we assume $u^T(\lambda)Au(\lambda)$ and $u^T(\lambda)Bu(\lambda)$ are almost constant for small changes in λ then the slope of the eigenvalue curve is almost linear in λ and hence parabolic-like function in λ. In case $A \succ 0$ and assume $u(\lambda)$ is almost constant for small changes of λ then the derivative of $\mu'(\lambda)$ with respect to λ is positive and hence the curve of $\mu(\lambda)$ is facing up.

Observation 3.3.1. *The eigenvalue curve $\mu(\lambda)$ of $Q(\lambda)$ is parabolic-like curve and its vertex occurs near some $\lambda = \omega$, where ω is a Rayleigh quotient to the pencil $Q'(\lambda)$.*

In fact,

$$\begin{aligned} 0 = \mu'(\omega) &= u^T(\omega)Q'(\omega)u(\omega) \\ &= 2\omega u^T(\omega)Au(\omega) + u^T(\omega)Bu(\omega) \end{aligned}$$

then

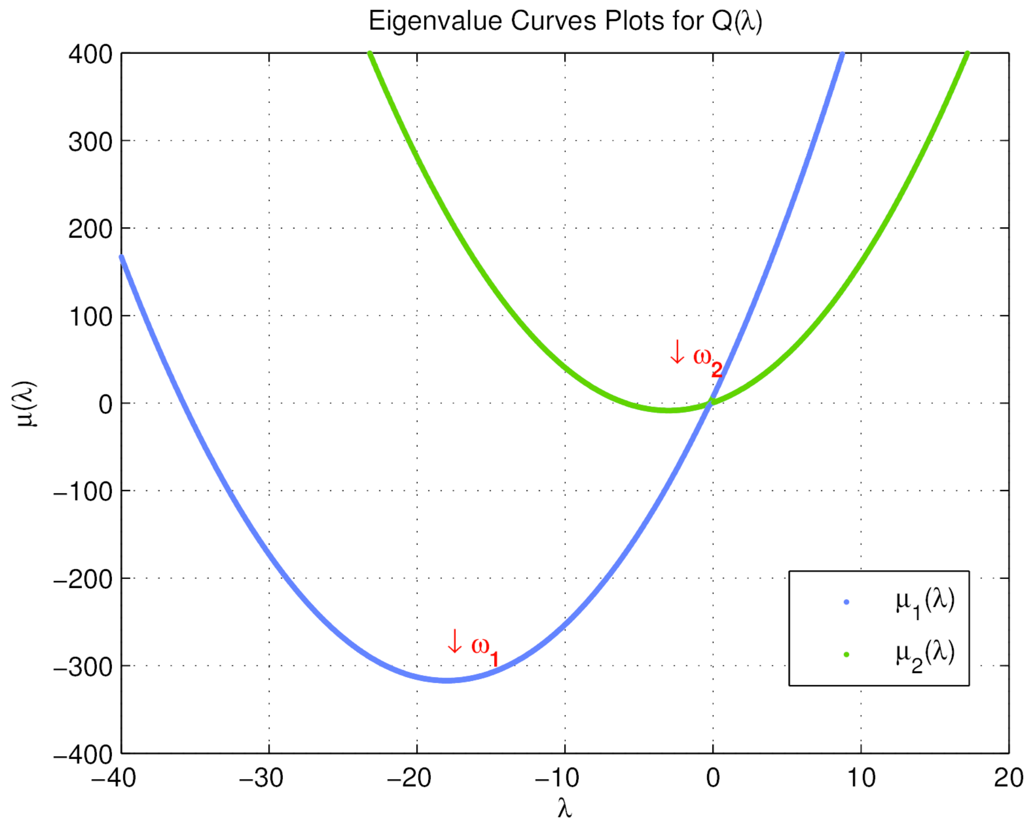
$$\omega = \frac{-u^T(\omega)Bu(\omega)}{2u^T(\omega)Au(\omega)}. \tag{3.4}$$

Example 3.3.3. Consider the HQEP from Example 3.3.1. We have

$$Q'(\lambda) = 2\lambda A + B,$$

and by using `eig(B,-2A)`, we get the eigenvalues of $Q'(\lambda)$: $\omega_1 = -18$ and $\omega_1 = -3$. Note that ω_i is where the vertex occurs on the λ-axis of the parabolic-like curve $\mu_i(\lambda)$ which has $\lambda_i^{\pm}$ as its roots (see Figure 3.4).

Figure 3.4. Plot of eigenvalue curves of $Q(\lambda)$ with eigenvalues of $Q'(\lambda)$.



3.4 Hyperpolicity Test and Finding the Gap of the HQEP

The `CenterHQEP` algorithm to be introduced is essential to finding the eigenvalues of an HQEP using bisection as well as a test for the QEP hyperbolicity. The algorithm is guaranteed to converge to some value in the gap of $Q(\lambda)$ if the QEP is hyperbolic or in other words the interval $(\lambda_n^-, \lambda_1^+)$ is not empty. Therefore if

$$[\lambda_n^-, \lambda_1^+] \cap [a, b] \neq \emptyset,$$

the algorithm will find a value in the gap of the HQEP in the interval $[a, b]$. The next theorem will give the range of the quadratic eigenvalues of the HQEP which will contain the interval $(\lambda_n^-, \lambda_1^+)$.

Theorem 3.4.1. *All the quadratic eigenvalues of an HQEP is contained in the interval,*

$$(-\frac{\lambda_B^{\max}}{\lambda_A^{\min}}, \frac{\lambda_B^{\max}}{\lambda_A^{\min}}). \tag{3.5}$$

Proof. Define $M_x = x^T M x$ for any symmetric matrix M and a nonzero vector x. Then we have

$$0 \leq \sqrt{B_x^2 - 4A_xC_x} \leq |B_x| \leq \lambda_B^{\max},$$

and $A_x > 0$. Therefore we get

$$\begin{aligned} \lambda &= \frac{-B_x \pm \sqrt{B_x^2 - 4A_xC_x}}{2A_x} \\ &\leq \frac{-B_x + \sqrt{B_x^2 - 4A_xC_x}}{2A_x} \\ &\leq \frac{-B_x + \lambda_B^{\max}}{2\lambda_A^{\min}} \\ &\leq \frac{2\lambda_B^{\max}}{2\lambda_A^{\min}} = \frac{\lambda_B^{\max}}{\lambda_A^{\min}}. \end{aligned}$$

The same can be applied to the lower bound and hence we get

$$-\frac{\lambda_B^{\max}}{\lambda_A^{\min}} \leq \lambda \leq \frac{\lambda_B^{\max}}{\lambda_A^{\min}},$$

where $\lambda_B^{\max}$ is the maximum absolute value of the eigenvalues of the symmetric matrix B, while $\lambda_A^{\min}$ is the minimum eigenvalue of the symmetric matrix A. $\square$

We may use Theorem 3.4.1 to determine the interval that the procedure `CenterHQEP` uses to find a value in the gap of HQEP but a tighter range may be found to speedup the procedure.

Observation 3.4.2. *Consider a hyperbolic $Q(\lambda)$ and the eigenvalues ω_i of its first derivative $Q'(\lambda) = 2\lambda A + B$ are ordered as follows:*

$$\omega_1 \leq \omega_2 \leq \ldots \leq \omega_{n-1} \leq \omega_n. \tag{3.6}$$

Then usually

$$[\lambda_n^-, \lambda_1^+] \cap [\omega_1, \omega_n] \neq \emptyset. \tag{3.7}$$

In fact, ω_1 is close to the vertex of the eigenvalue curve $\mu_1(\lambda)$ which has left (right) roots at λ_1^- (λ_1^+). Also, ω_n is close to the vertex of $\mu_n(\lambda)$ with left (right) roots at λ_n^- (λ_n^+) (see Figure 3.4).

Then we have

$$\omega_1 < \lambda_1^+ \text{ and } \lambda_n^- \leq \lambda_1^+ \implies [\lambda_n^-, \lambda_1^+] \cap [\omega_1, \lambda_1^+] \neq \emptyset,$$

also

$$\omega_n > \lambda_n^- \text{ and } \lambda_1^+ \geq \lambda_n^- \implies [\lambda_n^-, \lambda_1^+] \cap [\lambda_n^-, \omega_n] \neq \emptyset.$$

So

$$[\omega_1, \lambda_1^+] \cup [\lambda_n^-, \omega_n] = [\omega_1, \omega_n],$$

and hence

$$[\omega_1, \omega_n] \cap [\lambda_n^-, \lambda_1^+] \neq \emptyset.$$

The procedure will begin by solving the positive definite pencil $2\lambda A + B$ which is equivalent to a symmetric eigenvalue problem (SEP) since $A_x > 0$ for all nonzero vectors x. Taking the smallest (largest) pencil eigenvalues as the left (right) range to find the center, the procedure will use a level mechanism to find the gap by choosing the first value where the negative inertia of $Q(\lambda)$ equal to n.

Algorithm 3.5 Return a value in the gap interval of an HQEP

Given a $Q(\lambda)$ representing an HQEP as in (1.1), the procedure will return some λ_c where $\lambda_n^- < \lambda_c < \lambda_1^+$.

```
1: function CenterHQEP($A, B, C$)
2:     $a \leftarrow$ smallest eigenvalue of the pencil $2A\lambda + B$
3:     $b \leftarrow$ largest eigenvalue of the pencil $2A\lambda + B$
4:     $N_a \leftarrow$ NegcountHQEP($A, B, C, a$)
5:     if $N_a = n$ then
6:         return $a$
7:     end if
8:     $N_b \leftarrow$ NegcountHQEP($A, B, C, b$)
9:     if $N_b = n$ then
10:        return $b$
11:    end if
```

```
12:     $Level \leftarrow \max(N_a, N_b)$
13:     Declare Stack $Interval$
14:     while $Interval$ not empty do
15:         for index=size.$Interval$ : 1 do
16:             $(a, b) \leftarrow$ pop from $Interval$
17:             $N_{mid} \leftarrow$ NegcountHQEP($A, B, C, \frac{a+b}{2}$)
18:             if $N_{mid} = n$ then
19:                 return $\frac{a+b}{2}$
20:             else if $N_{mid} > Level$ then
21:                 $Level \leftarrow N_{mid}$
22:                 empty $Interval$
23:             end if
24:             put $(a, \frac{a+b}{2})$ into $Interval$
25:             put $(\frac{a+b}{2}, b)$ into $Interval$
26:         end for
27:     end while
28: end function
```

In the following example we will test the sensitivity of the algorithm to an HQEP with low degree of hyperbolicity.

Example 3.4.1. Consider the HQEP with

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B = \epsilon \begin{bmatrix} 1 & 0 \\ 0 & 6 \end{bmatrix}, C = \begin{bmatrix} \frac{1}{2} & 1 \\ 1 & 7 \end{bmatrix}. \tag{3.8}$$

For $\epsilon \geq 1.797789047$ the HQEP is overdamped as was shown in [1], and hence has $2n$ real nonpositive eigenvalues. The closest value of ϵ to 1.797789047 that the algorithm was able to converge was $\epsilon = 1.79779$ which has 2 inner eigenvalues of $\lambda_2^- = -1.152510604112638$ and $\lambda_1^+ = -1.150828354812121$, and hence the hyperbolicity degree $|\lambda_2^- - \lambda_1^+| = 0.001682249300517$. `CenterHQEP` returned $\lambda_c = -1.150979047272038$.

3.5 Implementing The Bisection Method for the HQEP

Now the `Bisect` algorithm can be modified for the HQEP case. The new algorithm will be called `BisectHQEP`. It will begin by calling `CenterHQEP` to find λ_c for a $Q(\lambda)$. If λ_c is contained in the interval then the interval will be initially partitioned by λ_c. Once that step is done, the procedure should behave more or less like `Bisect`.

Algorithm 3.6 Return the approximated quadratic eigenvalues of an HQEP using Bisection Method

Given a symmetric matrices A, B, C representing an HQEP, the procedure will return all its eigenvalues in the interval (a, b) as a vector Eig in a descending order.

```
1: function BisectHQEP($A, B, C, a, b$)
2:     Declare stack $Interval$
3:     Declare queue $Eig$
4:     $\lambda_c \leftarrow$ CenterHQEP($A, B, C$)
5:     if $a > \lambda_c$ and $b < \lambda_c$ then
6:         put $(a, \lambda_c)$ into $Interval$
7:         put $(\lambda_c, b)$ into $Interval$
8:     else
9:         put $(a, b)$ into $Interval$
10:    end if
11:    while $Interval$ not empty do
12:        $(a, b) \leftarrow$ pop from $Interval$
13:        $m \leftarrow |\texttt{NegcountHQEP}(A, B, C, a) - \texttt{NegcountHQEP}(A, B, C, b)|$
14:        if $m > 1$ or $(m = 1$ and $b - a > tol * \left|\frac{a+b}{2}\right|)$ then
15:            put $(a, \frac{a+b}{2})$ into $Interval$
16:            put $(\frac{a+b}{2}, b)$ into $Interval$
17:        else if $m = 1$ and $b - a < tol * \left|\frac{a+b}{2}\right|$ then
18:            put $(\frac{a+b}{2})$ into $Eig$
19:        end if
20:    end while
21:    return $Eig$
22: end function
```

Example 3.5.1. Consider the HQEP with

$$A = I_n,\ B = \begin{bmatrix} 20 & -10 & & & \\ -10 & 30 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 30 & -10 \\ & & & -10 & 20 \end{bmatrix}_{n\times n},\ C = \begin{bmatrix} 15 & -5 & & & \\ -5 & 15 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 15 & -5 \\ & & & -5 & 15 \end{bmatrix}_{n\times n}.$$

We will set $tol = 1.0e - 014$ and run `BisectHQEP` with different values of n and measure the runtime sensitivity as n grows.

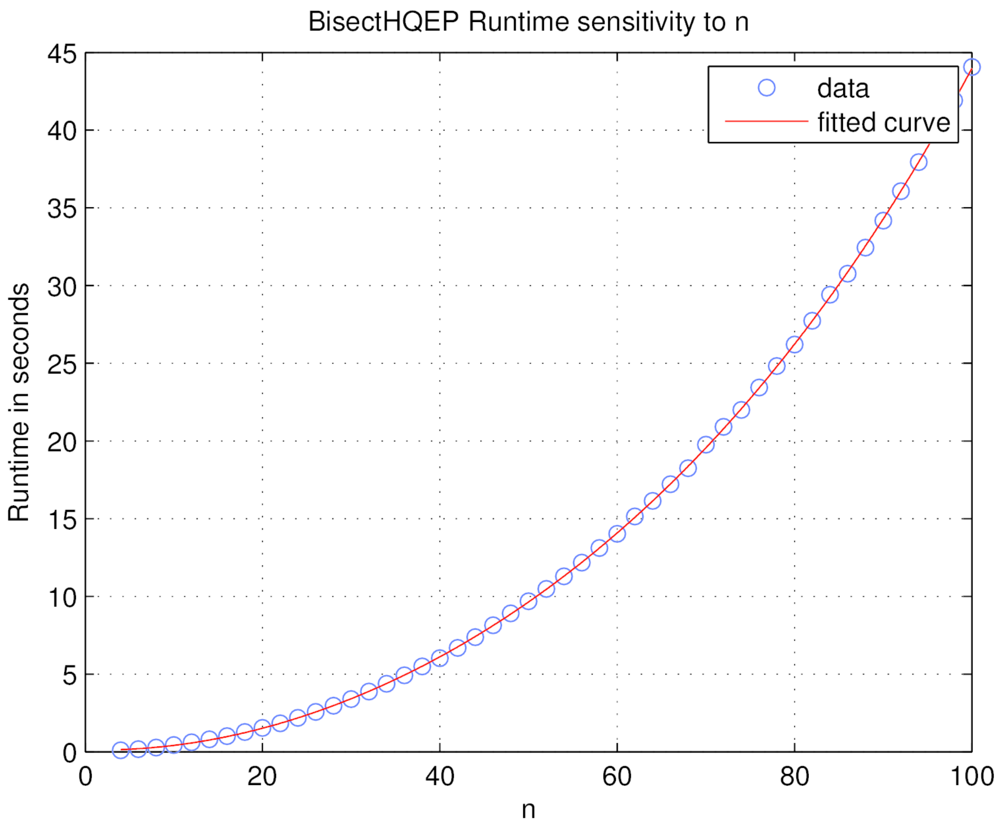


Figure 3.5. `BisectHQEP` Runtime Plot.

The fitted curve in Figure 3.5 is

$$T(n) = (1.647 \times 10^{-07})n^4 - (1.638 \times 10^{-05})n^3 + (0.004543)n^2 - (0.01681)n + 0.1399. \tag{3.9}$$

$T(n)$: the runtime of `BisectHQEP` acts quadratically for small n while for very large n has a cost of $\mathcal{O}(n^4)$. The reason that `BisectHQEP` is so time expensive is because of the procedure `NegcountHQEP` which cost $\mathcal{O}(n^3)$ and has to be executed an order of n times.

Our goal is to apply `BisectHQEP` to large HQEP as in Example 3.5.1. For $n = 1000$ the runtime estimate using $T(n)$ would be

$$T(1000) \approx 1.5 \times 10^5 \text{ seconds } \approx 42 \text{ hours}, \tag{3.10}$$

for $n = 10000$, the algorithm will take around 53 years to finish. These are unacceptable time costs as $n = 1000$ and larger are very common in HQEPs. It is hard to make a dense HQEP tridiagonal but as in Example 3.5.1, the HQEP is already tri-diagonal and hence we may be able to take advantage of the banded structure to reduce the cost of `NegcountHQEP`.

CHAPTER 4

Bisection Method for Banded HQEP

In this chapter, we will develop a negative inertia procedure that will take advantage of the banded structure of the banded HQEP by beginning with the tri-diagonal HQEP. The negative inertia algorithm for the tri-diagonal to be introduced is based on Algorithm 3.2 which was shown to be stable without any pivoting in Lemma 5.3 in [3]. Then we will extend the same logic to the banded HQEPs beginning with the penta-diagonal HQEPs and analyse its efficiency and stability.

4.1 The Tri-diagonal HQEP

4.1.1 Fast Inertia Calculation

We can write the tri-diagonal HQEP as a tridiagonal matrix polynomial:

$$Q(\lambda) = \begin{bmatrix} g_1(\lambda) & h_1(\lambda) & & & \\ h_1(\lambda) & g_2(\lambda) & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & g_{n-1}(\lambda) & h_{n-1}(\lambda) \\ & & & h_{n-1}(\lambda) & g_n(\lambda) \end{bmatrix}, \tag{4.1}$$

where

$$\begin{aligned} g_i(\lambda) &= A_{(i,i)}\lambda^2 + B_{(i,i)}\lambda + C_{(i,i)}, \\ h_i(\lambda) &= A_{(i+1,i)}\lambda^2 + B_{(i+1,i)}\lambda + C_{(i+1,i)}. \end{aligned}$$

Then

$$Q(\lambda) = L(\lambda)D(\lambda)L(\lambda)^T = L(\lambda) \begin{bmatrix} d_1(\lambda) & & & \\ & d_2(\lambda) & & \\ & & \ddots & \\ & & & d_n(\lambda) \end{bmatrix} L(\lambda)^T, \tag{4.2}$$

where

$$\begin{aligned} d_1(\lambda) &= g_1(\lambda), \\ d_i(\lambda) &= g_i(\lambda) - \frac{h_{i-1}(\lambda)^2}{d_{i-1}(\lambda)}, \quad i = 2, ..., n. \end{aligned} \tag{4.3}$$

Now the new negative inertia counter procedure `NegcountTHQEP` will be $\mathcal{O}(n)$ in time, making the whole bisection algorithm cost $\mathcal{O}(n^2)$. The new `CenterTHQEP` and `BisectTHQEP` algorithms are exactly the same as the original ones except they will call `NegcountTHQEP` instead of `NegcountHQEP` for the inertia calculation.

Algorithm 4.1 Return the number of negative eigenvalues of a tri-diagonal $Q(\lambda)$ for a specific λ

```
1: function NEGCOUNTTHQEP($A, B, C, \lambda$)
2:     $num \leftarrow 0$
3:     $d \leftarrow (A_{(i,i)}\lambda + B_{(i,i)})\lambda + C_{(i,i)}$
4:     if $d < 0$ then
5:         $num \leftarrow num + 1$
6:     end if
7:     for $i = 2$ to $n$ do
8:         $h \leftarrow (A_{(i,i-1)}\lambda + B_{(i,i-1)})\lambda + C_{(i,i-1)}$
9:         $g \leftarrow (A_{(i,i)}\lambda + B_{(i,i)})\lambda + C_{(i,i)}$
10:        $d \leftarrow g - \frac{h^2}{d}$
11:        if $d < 0$ then
12:            $num \leftarrow num + 1$
13:        end if
14:    end for
15:    return $num$
16: end function
```

4.1.2 A Numerical Example

Example 4.1.1. In this example we will see that `BisectTHQEP` will have a cost of $\mathcal{O}(n^2)$ flops. Consider the tri-diagonal HQEP from Example 3.5.1. We set $tol = 1.0e - 014$ and run `BisectTHQEP` with different values of n to test for time cost.

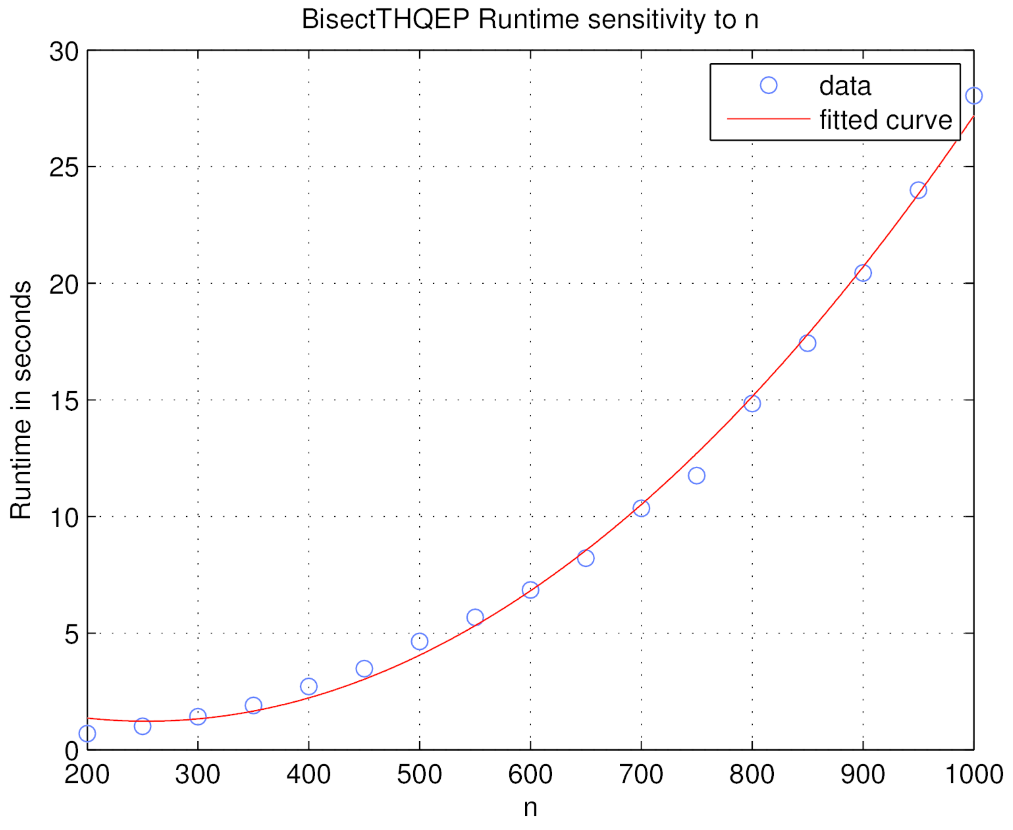


Figure 4.1. `BisectTHQEP` Runtime Plot for Large Tri-diagonal HQEP.

The fitted curve in Figure 4.1 is

$$T(n) = (4.66 \times 10^{-05})n^2 - (0.02363)n + 4.219. \tag{4.4}$$

4.1.3 Stability Analysis

Algorithm 4.2.1 basically computes the inertia of the symmetric tri-diagonal matrix $Q(\lambda)$ for a fixed value of λ by computing the diagonal entries of the diagonal matrix D in the LDL^T decomposition. In the process the algorithm counts the number of negative entries of the diagonal of D to determine the negative inertia of the symmetric matrix $Q(\lambda)$. The concern here is about the error that may be embedded in the entries of D and hence changing the sign of one or more entries making the inertia calculation inaccurate. Therefore, some kind of pivoting may be needed to compute the LDL^T decomposition which is based on Gaussian Elimination. Algorithm 4.2.1 is almost identical to algorithm 3.2 that was shown to be stable without pivoting in Lemma 5.3 [3]. The other concern may be of computing the entries of the symmetric matrix $Q(\lambda)$ given a fixed value of λ in Algorithm 4.2.1 in lines 3, 8,and 9. If we use Horner's algorithm to compute the required values of $g_i(\lambda)$ and $h_i(\lambda)$ then the overall method will be backward stable [20].

4.1.4 Cost Analysis

A single call of `NegcountTHQEP` will cost $11n$ flops and hence to find k quadratic eigenvalues using `BisectTHQEP` we will need $\mathcal{O}(kn)$ flops. `BisectTHQEP` will cost $\mathcal{O}(n^2)$ flops to find all the quadratic eigenvalues *vs* $\mathcal{O}(n^4)$ of the original bisection algorithm `BisectHQEP` using `NegcountHQEP`. In the case of memory requirement, the tri-diagonal matrices of the underlying $Q(\lambda)$ can be stored in $(3(2(n-1)+1)) \approx 6n$ memory locations and $2n$ memory locations to store the computed quadratic eigenvalues.

4.1.5 Convergence

If the QEP is hyperbolic and the gap between the primary and the secondary quadratic eigenvalues is large enough, then `CenterTHQEP` will converge and will find a value in the gap of the HQEP as was seen in Example 3.4.1. Otherwise, either the QEP is not hyperbolic or has a very low degree of hyperbolicity. `BisectTHQEP` will always return all the quadratic eigenvalues of an HQEP within some interval. In case that two or more quadratic eigenvalues are so close and hence they fall inside a partition with length less than the tolerance ϵ then the algorithm will return all the quadratic eigenvalues approximated in a uniform distribution fashion within the partition on hand. The convergence of `BisectTHQEP` to a quadratic eigenvalue is linear. To accelerate convergence we may use the Newton's method and the roots of the characteristic polynomial as in [19].

4.1.6 Quadratic Eigenvectors and Residuals

Once a quadratic eigenvalue λ is computed then we can find the associated quadratic eigenvector x by solving the following equation for x using the `null` procedure of Matlab which returns the non-zero kernel of a matrix,

$$Q(\lambda)x = 0.$$

If λ is a computed quadratic eigenvalue and x is the computed quadratic eigenvector then the relative residual

$$R(\lambda) = \frac{\|(\lambda^2 A + \lambda B + C)x\|}{(\lambda^2 \|A\| + |\lambda| \|B\| + \|C\|) \|x\|}$$

can be used to measure the quality of the approximate quadratic eigenpair (λ, x). In case that λ and x are exact, we should get a residual zero but as long as the residual is small, we can accept the eigenpair as a good approximation. The closer the residual is to zero the better the approximate eigenpair. Now we will compare the residuals for the calculated quadratic eigenpairs using `BisectTHQEP`, Matlab's `Polyeig`, and the Solvent approach procedure `ReturnSolvent` from Section 2.6.2 for the HQEP in Example 3.5.1 for $n = 100$ (see Figure 4.2).

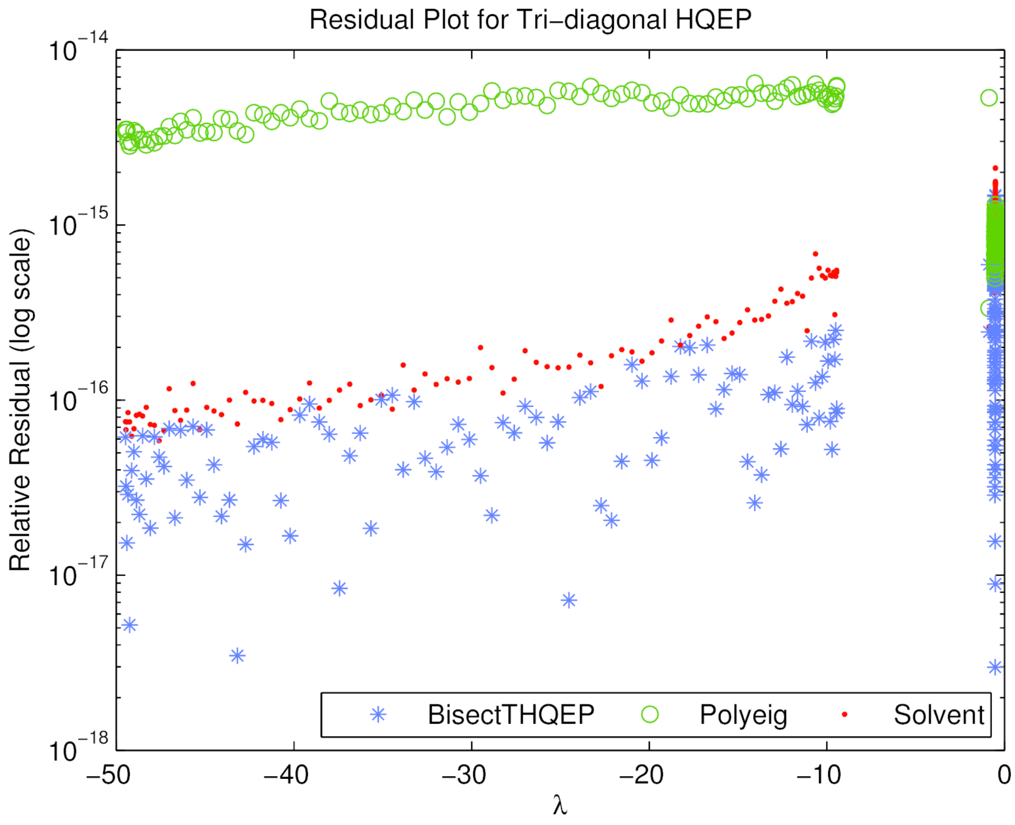


Figure 4.2. Comparison of Relative Residual for a tri-diagonal HQEP solved by `BisectTHQEP`, `Polyeig`, and `ReturnSolvent`.

The implemented bisection algorithm is about 6 bits more accurate than Matlab's `Polyeig` while more accurate by about 3 bits compared to the solvent approach. This difference in accuracy is more visible if the size of the problem gets bigger as will be seen later. Looking at the right of Figure 4.2, as the eigenvalues get very close in value, the bisection method shows even more superiority over the compared approaches.

4.2 The Penta-diagonal HQEP

Now that we have a procedure to compute the quadratic eigenvalues of a tri-diagonal HQEP for $Q(\lambda) = A\lambda^2 + B\lambda + C$, we will take the next step in developing our solution for penta-diagonal HQEPs and further.

4.2.1 Fast Inertia Calculation

In the case of Penta-diagonal HQEP, the bisection procedure won't change. We only need to adapt the inertia calculation to take into consideration the extra diagonals in the problem. Now consider the penta-diagonal symmetric matrix

$$\begin{bmatrix} a_1 & b_1 & c_1 & 0 & \cdots & 0 \\ b_1 & a_2 & b_2 & c_2 & \ddots & \vdots \\ c_1 & b_2 & a_3 & b_3 & \ddots & 0 \\ 0 & c_2 & b_3 & a_4 & \ddots & c_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & b_{n-1} \\ 0 & \cdots & 0 & c_{n-2} & b_{n-1} & a_n \end{bmatrix} = LDL^T$$

where

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ l_1 & 1 & 0 & 0 & \ddots & \vdots \\ m_1 & l_2 & 1 & 0 & \ddots & 0 \\ 0 & m_2 & l_3 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & m_{n-2} & l_{n-1} & 1 \end{bmatrix}, \text{ and } D = \mathrm{diag}(d_1, d_2, \cdots, d_n).$$

So we get

$$\begin{aligned} d_1 &= a_1 \\ d_2 &= a_2 - d_1 l_1^2 \end{aligned}$$

$$\begin{aligned} &\vdots \\ d_i &= a_i - \frac{c_{i-2}^2}{d_{i-1}} - d_{i-1}l_{i-1}^2,\ i = 3, \cdots, n, \end{aligned} \tag{4.5}$$

where

$$\begin{aligned} l_1 &= \frac{b_1}{d_1} \\ &\vdots \\ l_i &= \frac{b_i - l_{i-1}c_{i-2}}{d_i},\ i = 2, \cdots, n-1. \end{aligned}$$

Define

$$\begin{aligned} a_i &= A_{(i,i)}\lambda^2 + B_{(i,i)}\lambda + C_{(i,i)}, \\ b_i &= A_{(i+1,i)}\lambda^2 + B_{(i+1,i)}\lambda + C_{(i+1,i)}, \\ c_i &= A_{(i+2,i)}\lambda^2 + B_{(i+2,i)}\lambda + C_{(i+2,i)}. \end{aligned}$$

Now we can implement the procedure `NegcountPentaHQEP` to compute the inertia of $Q(\lambda)$.

Algorithm 4.2 Return the number of negative eigenvalues of a penta-diagonal $Q(\lambda)$ for a specific λ

```
1: function BisectPentaHQEP($A, B, C, \lambda$)
2:     $num \leftarrow 0$
3:     $a_i := (A_{(i,i)}\lambda + B_{(i,i)})\lambda + C_{(i,i)}$
4:     $b_i := (A_{(i+1,i)}\lambda + B_{(i+1,i)})\lambda + C_{(i+1,i)}$
5:     $c_i := (A_{(i+2,i)}\lambda + B_{(i+2,i)})\lambda + C_{(i+2,i)}$
6:     $d_1 \leftarrow a_1$
7:     if $d_1 < 0$ then
8:         $num \leftarrow num + 1$
9:     end if
10:    $l_1 = \frac{b_1}{d_1}$
11:    $d_2 = a_2 - d_2 L(1)^2$
12:    if $d_2 < 0$ then
13:        $num \leftarrow num + 1$
14:    end if
15:    for $i = 3$ to $n$ do
16:        $l_{i-1} = \frac{b_{i-1} - l_{i-2} c_{i-2}}{d_{i-1}}$
17:        $d_i = a_i - \frac{c_{i-2}^2}{d_{i-2}} - d_{i-1} l_{i-1}^2$
18:        if $d_i < 0$ then
19:            $num \leftarrow num + 1$
20:        end if
21:    end for
22:    return $num$
23: end function
```

4.2.2 A Numerical Example

Example 4.2.1. Consider a penta-diagonal HQEP with $n = 100$

$$A = I_n,\ B = \begin{bmatrix} 20 & -10 & -3 & & \\ -10 & 30 & \ddots & \ddots & \\ -3 & \ddots & \ddots & \ddots & -3 \\ & \ddots & \ddots & 30 & -10 \\ & & -3 & -10 & 20 \end{bmatrix}_{n\times n},\ C = \begin{bmatrix} 15 & -5 & -1 & & \\ -5 & 15 & \ddots & \ddots & \\ -1 & \ddots & \ddots & \ddots & -1 \\ & \ddots & \ddots & 15 & -5 \\ & & -1 & -5 & 15 \end{bmatrix}_{n\times n}.$$

We will compute the relative residual of the bisection procedure `BisectPentaHQEP` compared to the other two algorithms as in Figure 4.3.

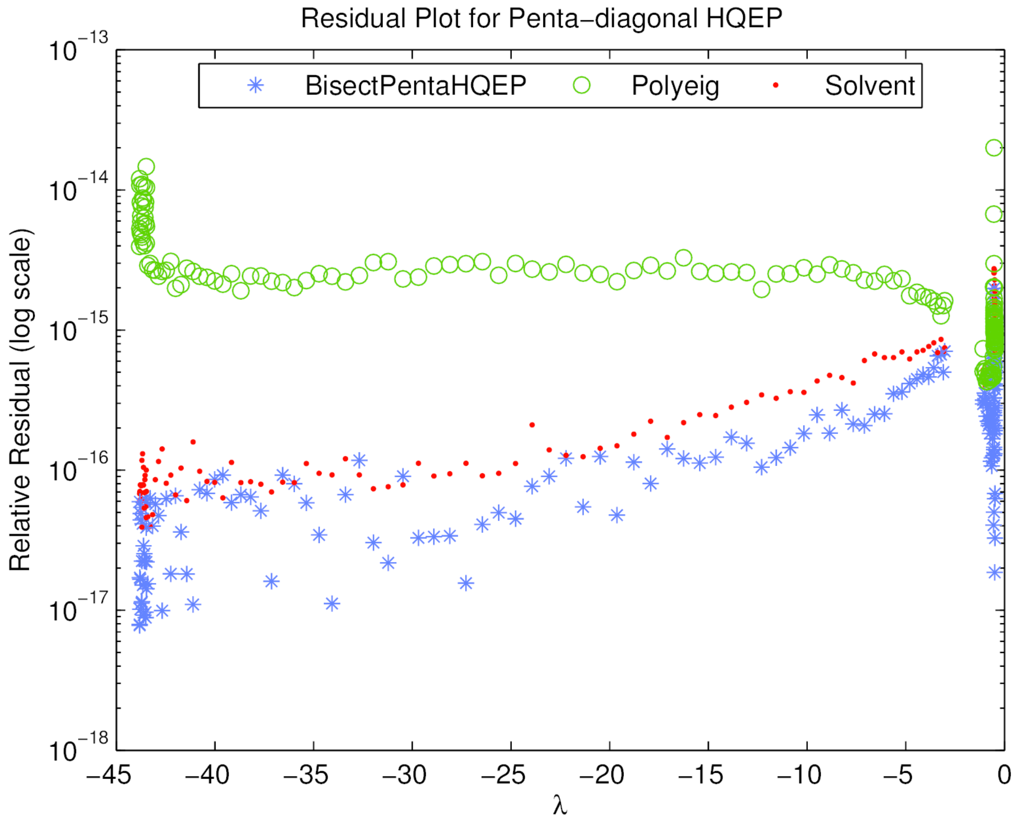


Figure 4.3. Comparison of Relative Residual for a Penta-diagonal HQEP solved by `BisectTHQEP`, `Polyeig`, and `ReturnSolvent`.

4.2.3 Stability Analysis

We saw in Section 4.1.3 that calculating the inertia of a tri-diagonal symmetric matrix is backward stable even without any pivoting. In the case of the Penta-diagonal, we can't make such claim and hence calculating the LDL^T decomposition using (4.6) without pivoting may be unstable. The issue with pivoting is that it doesn't conserve the congruency and therefore changes the inertia of the matrix. In [21], a method was introduced to apply pivoting while conserving congruency at the same time as follow. If M is a symmetric matrix and Π is a permutation matrix then we get

$$\Pi M \Pi^T = \begin{bmatrix} E & C^T \\ C & B \end{bmatrix} = \begin{bmatrix} I_s & 0 \\ CE^{-1} & I_{n-s} \end{bmatrix} \underbrace{\begin{bmatrix} E & 0 \\ 0 & B - CE^{-1}C^T \end{bmatrix}}_{\hat{M}} \begin{bmatrix} I_s & 0 \\ CE^{-1} & I_{n-s} \end{bmatrix}^T,$$

where the inertia of $\hat{M}$ and M are the same by Theorem 2.2.1, and the matrix E is 1×1 or 2×2. The choice of the permutation matrix Π depends on the rows and columns to be permuted according to the pivot location, size, and the pivoting technique to be used. Although calculating the inertia of $\hat{M}$ is stable compared to calculating the inertia of M directly, the permutation doesn't conserve the banded structure of M. If the matrix M is penta-diagonal then there is no guarantee that the matrix $B - CE^{-1}C^T$ is still penta-diagonal.

Since calculating the LDL^T decomposition of the tri-diagonal matrix and hence inertia is inherently stable even without any pivoting, reducing the penta-diagonal matrix to a tri-diagonal in a stable algorithm and then computing the inertia of the resulting matrix will make the procedure overall stable. In the next chapter we will implement such approach for the penta-diagonal matrix and the banded matrix, in general.

CHAPTER 5

Bisection for the Banded HQEP by Tridiagonalization

Applying (4.6) to a penta-diagonal HQEP to calculate the negative inertia is unstable without some pivoting, but pivoting will destroy the banded structure of the problem and with it the advantage of the bisection procedure that we implemented in this study. Another approach was suggested in Schwarz [22], and that to reduce the band of a banded matrix to a tri-diagonal matrix. Once this is done then we can use the tri-diagonal negative inertia calculation Algorithm 4.2.1 to compute the inertia. The reduction must have these characteristics to be a viable solution to the problem, namely:

1. Must conserve congruency.
2. Must be stable.
3. Must be efficient.

5.1 The Givens Rotations

We will begin by applying tridaigonalization to the penta-diagonal case. Later in the section we will introduce an algorithm for the general banded case. The tridaigonalization will be implemented using the Givens rotation.

Givens Rotations are used in the triangularization process where a symmetric matrix is reduced to an upper triangular matrix to compute the QR-factorization. They are also used in the tridiagonalization of a symmetric matrix to a tri-diagonal form for the purpose of computing the eigenvalues using the QR algorithm. The purpose of Givens rotations are to annihilate elements in the outer bands of the matrix beginning from the outermost band and working inward until it annihilates all bands except the first, leaving the matrix in a tri-diagonal form [22].

A single Givens rotation consists of a similarity transformation:

$$\hat{M} = U^T M U, \tag{5.1}$$

where M is a symmetric matrix and U is an orthogonal matrix different from the identity matrix in the elements

$$u_{pp} = u_{qq} = \cos\theta, \quad u_{pq} = -u_{qp} = \sin\theta,$$

and θ is the angle of rotation in the pq-plane.

Example 5.1.1. Consider the symmetric matrix $M \in \mathbb{R}^{3\times 3}$ which happens also to be penta-diagonal

$$M = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 5 & 6 \\ 4 & 6 & 7 \end{bmatrix}.$$

To eliminate the elements $M_{1,3} = M_{3,1} = 4$, we need to have

$$M_{1,3}\cos\theta + M_{1,2}\sin\theta = 0,$$

giving

$$\theta = \tan^{-1}(-\frac{M_{1,3}}{M_{1,2}}),$$

giving

$$\cos\theta = \frac{-M_{1,3}}{r}, \quad \sin\theta = \frac{M_{1,2}}{r},$$

where $r = \sqrt{M_{1,3}^2 + M_{1,2}^2}$. Therefore we get

$$U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.4160 & -0.5547 \\ 0 & 0.5547 & 0.4160 \end{bmatrix}$$

and

$$\hat{M} = U^T M U = \begin{bmatrix} 2 & 3.4669 & 0 \\ 3.4669 & 5.7885 & -0.3462 \\ 0 & -0.3462 & -0.0192 \end{bmatrix}.$$

In the previous example, to eliminate the elements $M_{1,3}(M_{3,1})$ we chose $p = 2$ and $q = 3$. In general to eliminate the elements $M_{i,j}(M_{j,i})$ with $j > i$, we choose $p = j - 1$ and $q = j$. Note that the only rows (columns) that changed are the 2nd and the 3rd, while the element $M_{1,1}$ retained its value. In general, the transformation (5.1) changes only the rows (columns) of M with the indices p and q, while leaving the other elements unchanged. In the case that the matrix is large enough (for example if the matrix is penta-diagonal with size $n \geq 5$) the transformation (5.1) will introduce non-zero elements outside the band in locations where elements were initially zero.

Example 5.1.2. Consider the penta-diagonal matrix $M \in \mathbb{R}^{5\times 5}$

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

To eliminate the elements $M_{1,3}(M_{3,1})$ we form the rotation matrix

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.4160 & -0.5547 & 0 & 0 \\ 0 & 0.5547 & 0.4160 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and get

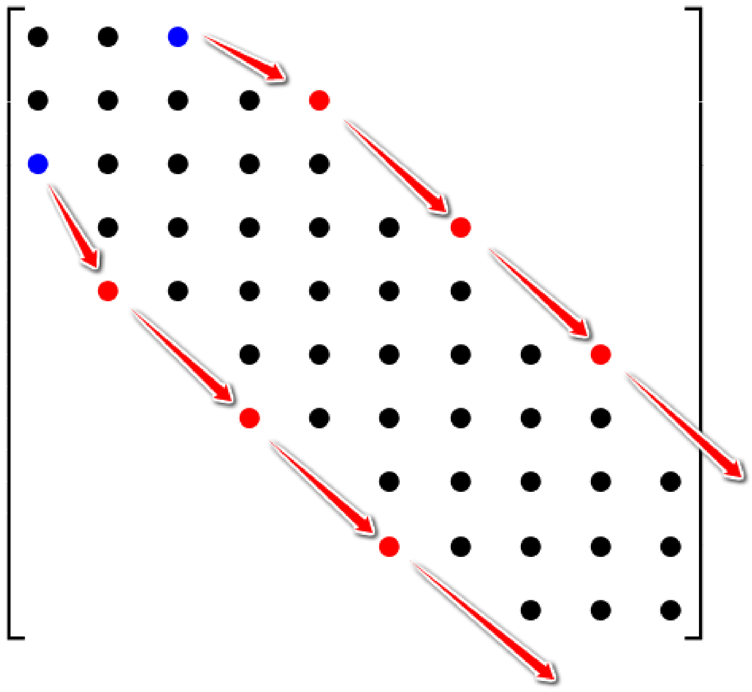
$$\hat{M} = U^TMU = \begin{bmatrix} 2 & 3.4669 & 0 & 0 & 0 \\ 3.4669 & 5.7885 & -0.3462 & 7.3498 & 0.5547 \\ 0 & -0.3462 & -0.0192 & -0.5547 & 0.4160 \\ 0 & 7.3498 & -0.5547 & 3 & 9 \\ 0 & 0.5547 & 0.4160 & 9 & 1 \end{bmatrix}.$$

The transformation did indeed eliminate $M_{1,3}(M_{3,1})$ but in the process it changed $M_{2,5}$ and symmetrically $M_{5,2}$ to a nonzero value outside the 2nd band, as can be seen in $\hat{M}_{2,5} = \hat{M}_{5,2} = 0.5547$, and hence another rotation is needed to eliminate $\hat{M}_{2,5}(\hat{M}_{5,2})$.

In general, the transformation while eliminating the elements $M_{i,j}(M_{j,i})$ with $j = i + 2$ on the 2nd band of the penta-diagonal matrix, will change the elements $M_{j-1,j+2}(M_{j+2,j-1})$ to a nonzero value, assuming $j + 2 \leq n$. The same happened in Example 5.1.1 except that the elements lied outside the matrix.

The whole process of transforming a penta-diagonal matrix into tri-diagonal will begin by eliminating the first element in the 2nd band symmetrically and then with additional rotations subsequent non-zero elements which appear outside the band will be eliminated, until the nonzero value is beyond the border of the matrix. Then the process will do the same for the 2nd element then the 3rd, until all the elements outside the 2nd band are eliminated (see Figure 5.1).

Figure 5.1. Tridiagonalization of a Penta-diagonal Matrix.



The Givens rotation doesn't only conserve congruency. In fact it is a similarity transformation that also conserve the matrix spectrum (spectrum of $\hat{M}$ is identical to the spectrum of M). Therefore the tri-diagonal matrix formed from a sequence of transformations of a penta-diagonal matrix using Givens rotations is not only having the same inertia as the original penta-diagonal matrix, but it also have identical set of eigenvalues. In addition, the Givens rotation is an orthogonal transformation and applying a sequence of orthogonal transformations is known to be backward stable according to Theorem 3.5 in [3]. For a penta-diagonal symmetric matrix with size n, we have $n-2$ entries in the 2nd band (one on each side of the symmetry) that we need to eliminate. For each element to eliminate we need $\mathcal{O}(n)$ rotations which make the overall cost of transforming a penta-diagonal into tri-diagonal cost $\mathcal{O}(n^2)$.

5.1.1 `BandSymmToTriArr`

Here we will extend the idea of tridiagonalization into band reduction for any banded symmetric matrix and even a dense symmetric matrix. `BandSymmToTriArr` will take the symmetry and the banded structure into consideration when storing the matrix, and when performing the calculations implicitly. We will store the entries of a banded symmetric matrix of k bands in an array storing only the diagonal and the k bands as columns, therefore saving the storage requirement from n^2 memory locations for a matrix of size n into kn memory locations only. The procedure will perform the calculations implicitly on the array and hence saving unnecessary memory storage and access operations.

Example 5.1.3. In this example we will illustrate storing a penta-diagonal symmetric matrix,

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

`BandSymmToTriArr` will take the symmetric banded matrix as an array (Arr) as in Example 5.1.3, the size of the matrix ($Size$) and the band size ($BandSize$). For example a tridiagonal matrix will have $BandSize = 1$, penta-diagonal will have $BandSize = 2$, and so on. Note that

$$0 \leq BandSize \leq Size - 1,$$

and the matrix is diagonal if $BandSize = 0$, and dense if $BandSize = Size - 1$. The output will be an array (Arr) of size ($Size \times (BandSize + 1)$) representing the banded matrix. The procedure to be presented here is due to [22]. We will present it here for reference purposes only. No changes were made to the procedure except that the indexing in the original procedure began with 0 for referencing the columns of the array, our procedure here will adjust the indexing to begin with 1.

Algorithm 5.1 Tridiagonalizing a Symmetric Banded Matrix

```
function BandSymmToTriArr($Arr, Size, BandSize$)
    $g \leftarrow 0$
    for $k = 1$ to $Size - 2$ do
        $maxr \leftarrow \texttt{min}(Size - k, BandSize)$;
        for $r = maxr$ to 2 step -1 do
            for $j = k + r$ to $Size$ step $BandSize$ do
                if $j = k + r$ then
                    if $Arr(k, r) = 0$ then
                        break
                    end if
                    $b \leftarrow -Arr(k, r)/Arr(k, r + 1)$
                    $ugl \leftarrow k$
                else
                    if $g == 0$ then
                        break
                    end if
                    $b \leftarrow Arr(j - BandSize - 1, BandSize + 1)/g$
                    $ugl \leftarrow j - BandSize$
                end if
```

```
20:     $s \leftarrow 1/\sqrt{1+b^2}$
21:     $c \leftarrow b * s$
22:     $c2 \leftarrow c^2$
23:     $s2 \leftarrow s^2$
24:     $cs \leftarrow c * s$
25:     $u \leftarrow c2 * Arr(j-1,1) - 2 * cs * Arr(j-1,2) + s2 * Arr(j,1)$
26:     $u1 \leftarrow s2 * Arr(j-1,1) + 2 * cs * Arr(j-1,2) + c2 * Arr(j,1)$
27:     $Arr(j-1,2) \leftarrow cs*(Arr(j-1,1)-Arr(j,1))+(c2-s2)*Arr(j-1,2)$
28:     $Arr(j-1,1) \leftarrow u$
29:     $Arr(j,1) \leftarrow u1$
30:     **for** $l = ugl$ to $j-2$ **do**
31:         $u \leftarrow c * Arr(l,j-l) - s * Arr(l,j-l+1)$
32:         $Arr(l,j-l+1) \leftarrow s * Arr(l,j-l) + c * Arr(l,j-l+1)$
33:         $Arr(l,j-l) \leftarrow u;$
34:     **end for**
35:     **if** $j \neq k+r$ **then**
36:         $Arr(j-BandSize-1, BandSize+1)$
37:             $\leftarrow c * Arr(j-BandSize-1, BandSize+1) - s * g$
38:     **end if**
```

```
39:                 $maxl \leftarrow \min(Size - j, BandSize - 1)$
40:                 **for** $l = 1$ to $maxl$ **do**
41:                     $u \leftarrow c * Arr(j-1, l+2) - s * Arr(j, l+1)$
42:                     $Arr(j, l+1) \leftarrow s * Arr(j-1, l+2) + c * Arr(j, l+1)$
43:                     $Arr(j-1, l+2) \leftarrow u$
44:                 **end for**
45:                 **if** $j + BandSize \leq BandSize + 1$ **then**
46:                     $g \leftarrow -s * Arr(j, BandSize + 1)$
47:                     $Arr(j, BandSize + 1) = c * Arr(j, BandSize + 1)$
48:                 **end if**
49:             **end for**
50:         **end for**
51:     **end for**
52:     $Arr \leftarrow Arr(1 : Size, 1 : 2)$
53:     **return** $Arr$
54: **end function**
```

5.2 Implementing Bisection for Banded HQEP

Here we introduce the complete solution for approximating the quadratic eigenvalues for the HQEP using the bisection method. Given three symmetric banded matrices A, B, C of size n and $A \succ 0$, we construct the quadratic matrix polynomial in λ

$$Q(\lambda) = A\lambda^2 + B\lambda + C.$$

The procedure that will check for the hyperbolicity `CenterBandedHQEP` will simultaneously return a value in the gap of the HQEP (center) that will partition the interval of the search into two partitions; the primary partition on the right of (center) where the n primary quadratic eigenvalues lie and the secondary partition on the left where the n secondary quadratic eigenvalues lie. The procedure `NegcountBandedHQEP` will return the number of negative eigenvalues of the symmetric matrix $Q(\lambda)$ for a given λ, which will be used by the main procedure `BisectBandedHQEP` to search for the roots of the eigenvalue curves and hence finding the quadratic eigenvalues. `CenterBandedHQEP` will also use `NegcountBandedHQEP` to find the (center). `NegcountBandedHQEP` will check if the matrix needs band reduction ($BandSize > 1$) by calling `BandSymmToTriArr` before using an implicit LDL^T decomposition on the matrix array to calculate the number of negative eigenvalues of the matrix. All the procedures in this solution will be applied to an array version of the matrices A, B, C, as was seen in Example 5.1.3 to save memory storage and access operations for the solution. The quadratic eigenvectors will be calculated using Matlab's procedure `null`.

5.2.1 `NegcountBandedHQEP`

The final version of the negative inertia counter algorithm will be suitable for any banded HQEP. If the HQEP is not tri-diagonal, the procedure `BandSymmToTriArr` will first reduce it to tri-diagonal and return the array back to `NegcountBandedHQEP`, otherwise `NegcountBandedHQEP` will calculate the number of negative eigenvalues right away. It will be using array versions of A, B, C and will calculate the matrix $Q(\lambda)$ as an array as well. The process of tridiagonalization will cost $\mathcal{O}(kn^2)$ where $k = BandSize$, while calculating the number of negative eigenvalues will cost $\mathcal{O}(n)$, overall the calling of `NegcountBandedHQEP` will cost at most $\mathcal{O}(n^2)$ if $k \ll n$. `NegcountBandedHQEP` will accept as inputs the matrices A, B, C as arrays, λ, the size of the matrix ($Size$) and the band size ($BandSize$).

Algorithm 5.2 Return the number of negative eigenvalues of a Banded $Q(\lambda)$

```
1: function NegcountBandedHQEP($A_{arr}, B_{arr}, C_{arr}, \lambda, Size, BandSize$)
2:     $Q_{arr} \leftarrow (A_{arr} * \lambda + B_{arr}) * \lambda + C_{arr}$
3:     if $BandSize > 1$ then
4:         $Q_{arr} \leftarrow$ BandSymmToTriArr($Q_{arr}, Size, BandSize$)
5:     end if
6:     $num \leftarrow 0$
7:     $d \leftarrow Q_{arr}(1,1)$
8:     if $d < 0$ then
9:         $num \leftarrow num + 1$
10:    end if
11:    for $i = 2$ to $Size$ do
12:        $d \leftarrow Q_{arr}(i,1) - \frac{Q_{arr}(i-1,2)^2}{d}$
13:        if $d < 0$ then
14:            $num \leftarrow num + 1$
15:        end if
16:    end for
17:    return $num$
18: end function
```

5.2.2 `CenterBandedHQEP`

The procedure `CenterBandedHQEP` is of great importance in this work. It checks for the hyperbolicity of a QEP which is not a straightforward task as was seen in Section 2.4. `CenterBandedHQEP` uses the concept of levels to narrow the search intervals using bisection until a value λ is found such that $Q(\lambda) \prec 0$. If

CenterBandedHQEP converges to such λ then the QEP is actually an HQEP and the hyperbolicity condition was met. It also returns that λ value to the main bisection procedure to partition the interval of search into 2 partitions, namely the primary and the secondary partitions. In each partition the procedure NegcountBandedHQEP will be monotonic as was seen in Section 3.3. CenterBandedHQEP will accept as inputs the matrices A, B, C as arrays, the beginning of the search interval (a), the end of the search interval (b), the size of the matrix $(Size)$ and the band size $(BandSize)$. a and b will be chosen using Observation 3.4.2, and will be passed on from the main bisection procedure. For CenterBandedHQEP to be effective in detecting hyperbolicity, we must have a stop mechanism to stop the procedure before the partitions gets too small and further partitioning won't be successful. If the partition size is too small then the machine will treat a and b as the same number and NegcountBandedHQEP may enter an infinite loop. We know that after n bisections the partition size will reach a length of $\frac{b-a}{2^n}$. Therefore the maximum number of partitions should not exceed

$$MaxCounter = \left\lfloor \log_2 \frac{b-a}{minPartSize} \right\rfloor,$$

where $minPartSize$ is the minimum partition size that terminates NegcountBanded-HQEP if a value in the gap in not found prior to that. From Example 3.4.1, CenterHQEP didn't converge for partitions smaller than $0.001682249300517 \approx 0.0017$ even if the problem was an HQEP. We will use $minPartSize = 0.0017$ in our numerical example as a convenient minimum partition size.

Finally, CenterBandedHQEP will be modified to simulate the stack mechanism used in procedure CenterHQEP, to decrease memory space and computation requirements.

Algorithm 5.3 Check for hyperbolicity and return the center of an HQEP

```
1: function CenterBandedHQEP($A_{arr}, B_{arr}, C_{arr}, a, b, Size, BandSize$)
2:     Select $minPartSize$
3:     $N_a \leftarrow$ NegcountBandedHQEP($A_{arr}, B_{arr}, C_{arr}, a, Size, BandSize$)
4:     if $N_a = Size$ then
5:         return $a$
6:     end if
7:     $N_b \leftarrow$ NegcountBandedHQEP($A_{arr}, B_{arr}, C_{arr}, b, Size, BandSize$)
8:     if $N_b = Size$ then
9:         return $b$
10:    end if
11:    $Level \leftarrow$ max($N_a, N_b$)
12:    Declare $Interval$ ($2 \times 2$) Array
13:    $Imarker \leftarrow 0$
14:    $Imarker \leftarrow Imarker + 1$
15:    $MaxCounter \leftarrow \left\lfloor \log_2 \frac{b-a}{minPartSize} \right\rfloor$
```

```
16:     for $i = 1$ to $MaxCounter$ do
17:         for $index = Imarker$ to 1 step $-1$ do
18:             $a \leftarrow Interval(1, index)$
19:             $b \leftarrow Interval(2, index)$
20:             $N_{mid} \leftarrow$ NegcountBandedHQEP$(A_{arr}, B_{arr}, C_{arr}, \frac{a+b}{2}, Size, BandSize)$
21:             if $N_{mid} = Size$ then
22:                 return $\frac{a+b}{2}$
23:             end if
24:             if $N_{mid} > Level$ then
25:                 $Imarker \leftarrow 0$
26:                 $Level \leftarrow N_{mid}$
27:             end if
28:             $Imarker \leftarrow Imarker + 1$
29:             $Interval(:, Imarker) \leftarrow [a\ \frac{a+b}{2}]$
30:             $Imarker \leftarrow Imarker + 1$
31:             $Interval(:, Imarker) \leftarrow [\frac{a+b}{2}\ b]$
32:         end for
33:     end for
34:     Print 'Not hyperbolic or weakly hyperbolic'
35:     Return NaN
36: end function
```

5.2.3 BisectBandedHQEP

The final version of the main bisection procedure `BisectBandedHQEP` accepts as inputs the matrices A, B, C, the beginning of the interval of the search (a), the ending of the interval of the search (b), the search tolerance which dictate how small a partition may get (tol), the size of the problem ($Size$) and the band size ($BandSize$). This version is modified from the previous version `BisectHQEP` by simulating the stack mechanism to decrease memory space and computation requirements. The procedure begins by converting all the matrices into array form using algorithm `MatToArr`. `BisectBandedHQEP` will call `CenterBandedHQEP` to find the center of the HQEP, if the center is included in the interval $[a, b]$, then it will partition it into two partitions namely $[a, center]$ and $[center, b]$ and add these two partition into the stack of partitions $Interval$, otherwise it will add $[a, b]$ to the stack $Interval$. The number of quadratic eigenvalues in a partition will be equal to the absolute difference in the output of the procedure `NegcountBandedHQEP` at the beginning and the ending of a partition [line 25]. It will keep discarding any partitions where there are no quadratic eigenvalues, and keep partitioning the remaining partitions until we get up to $2 * Size$ partitions of length within tol and each partition has at least one quadratic eigenvalue. In case that a partition is within tolerance and hence no further partitioning can be made and at the same time the number of quadratic eigenvalues are greater than one, the algorithm will return approximated quadratic eigenvalues within the partition by dividing the partition length and returning the approximated quadratic eigenvalues in a uniform distribution fashion [line 31].

Algorithm 5.4 Return the approximated quadratic eigenvalues of a Banded HQEP using Bisection Method

Given symmetric matrices A, B, C representing an HQEP, the procedure will return all its eigenvalues in the interval (a, b) as a vector Eig in a descending order.

```
1: function BisectBandedHQEP($A, B, C, a, b, tol, Size, BandSize$)
2:     $w_{min} \leftarrow$ smallest eigenvalue of the pencil $2A\lambda + B$
3:     $w_{max} \leftarrow$ largest eigenvalue of the pencil $2A\lambda + B$
4:     $A_{arr} \leftarrow$ MatToArr($A, BandSize$)
5:     $B_{arr} \leftarrow$ MatToArr($B, BandSize$)
6:     $C_{arr} \leftarrow$ MatToArr($C, BandSize$)
7:     $center \leftarrow$ CenterBandedHQEP($A_{arr}, B_{arr}, C_{arr}, w_{min}, w_{max}, Size, BandSize$)
8:     Declare $Interval$ $(2 \times 2 * Size)$ Array
9:     Declare $Eig$ $(2 * Size)$ Vector
10:    $Imarker \leftarrow 0$
11:    $Emarker \leftarrow 0$
12:    if $a < center$ and $center < b$ then
13:        $Imarker \leftarrow Imarker + 1$
14:        $Interval(:, Imarker) \leftarrow [a\, center]$
15:        $Imarker \leftarrow Imarker + 1$
16:        $Interval(:, Imarker) \leftarrow [center\, b]$
17:    else
18:        $Imarker \leftarrow Imarker + 1$
19:        $Interval(:, Imarker) \leftarrow [a\, b]$
20:    end if
```

```
21:     while $Imarker > 0$ do
22:         $a \leftarrow Interval(1, Imarker)$
23:         $b \leftarrow Interval(2, Imarker)$
24:         $Imarker \leftarrow Imarker + 1$
25:         $m \leftarrow$ NegcountBandedHQEP$(A_{arr}, B_{arr}, C_{arr}, a, Size, BandSize)$
    $-$NegcountBandedHQEP$(A_{arr}, B_{arr}, C_{arr}, b, Size, BandSize)$
26:         $m \leftarrow |m|$
27:         if $m \neq 0$ then
28:             if $b - a < tol * \left|\frac{a+b}{2}\right|$ then
29:                 for $j = 1$ to $m$ do
30:                     $Emarker \leftarrow Emarker + 1$
31:                     $Eig(Emarker) \leftarrow a + j * \frac{b-a}{c+1}$
32:                 end for
33:             else
34:                 $Imarker \leftarrow Imarker + 1$
35:                 $Interval(:, Imarker) \leftarrow [a\ \frac{a+b}{2}]$
36:                 $Imarker \leftarrow Imarker + 1$
37:                 $Interval(:, Imarker) \leftarrow [\frac{a+b}{2}\ b]$
38:             end if
39:         end if
40:     end while
41:     return $Eig$
42: end function
```

Although `BisectBandedHQEP` expects ($Size$) and ($BandSize$) as inputs, the procedure can be modified to compute these values at the beginning of the algorithm. $Size$ can be computed using Matlab's `size` function as follows,

```
    $Size \leftarrow$ size$(A, 1)$
```

while $BandSize$ can also be computed using,

```
    $Bands \leftarrow$ [returnBandSize$(A)$ returnBandSize$(B)$ returnBandSize$(C)$]
    $BandSize \leftarrow \max(Bands)$;
```

and the algorithm

Algorithm 5.5 Return the band size of a symmetric matrix M

```
1: function RETURNBANDSIZE($M$)
2:     $Size \leftarrow$ size$(M, 1)$
3:     for $j = 1$ to $Size - 1$ do
4:         if norm(diag$(M, j-1), 1) < \epsilon$ then
5:             Break
6:         end if
7:     end for
8:     Return $j - 2$
9: end function
```

Now we will introduce the algorithm `MatToArr`

Algorithm 5.6 Return the array form of a symmetric matrix M

```
1: function MatToArr($M, BandSize$)
2:     $Size \leftarrow$size$(M, 1)$
3:     Declare $Arr$ $(Size \times (BandSize + 1))$ Array
4:     for $j = 1$ to $BandSize + 1$ do
5:         $Arr(1 : Size - j + 1, j) \leftarrow$diag$(M, j - 1)$
6:     end for
7:     Return $Arr$
8: end function
```

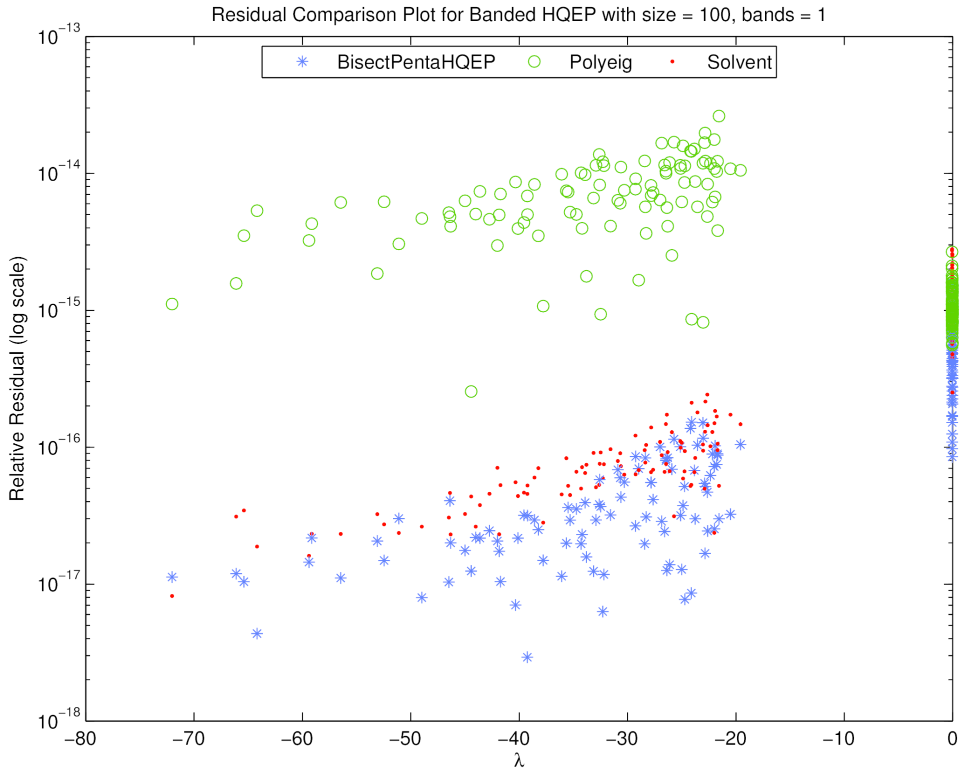
5.3 Numerical Examples

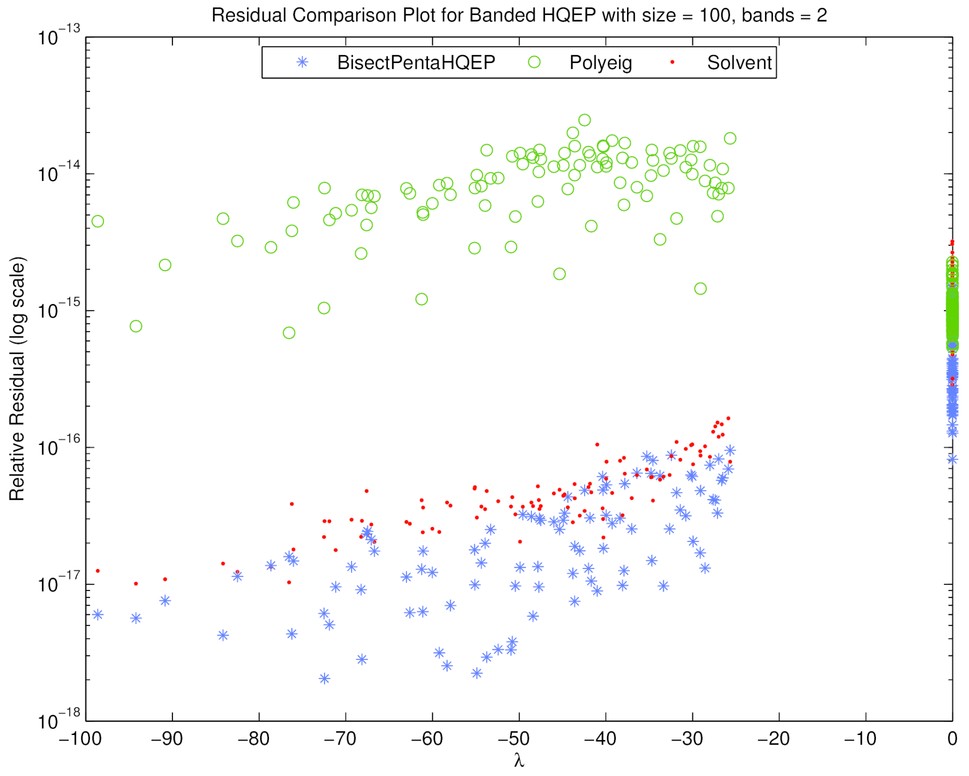
If (λ, x) is a computed quadratic eigenpair of the HQEP then the relative residual

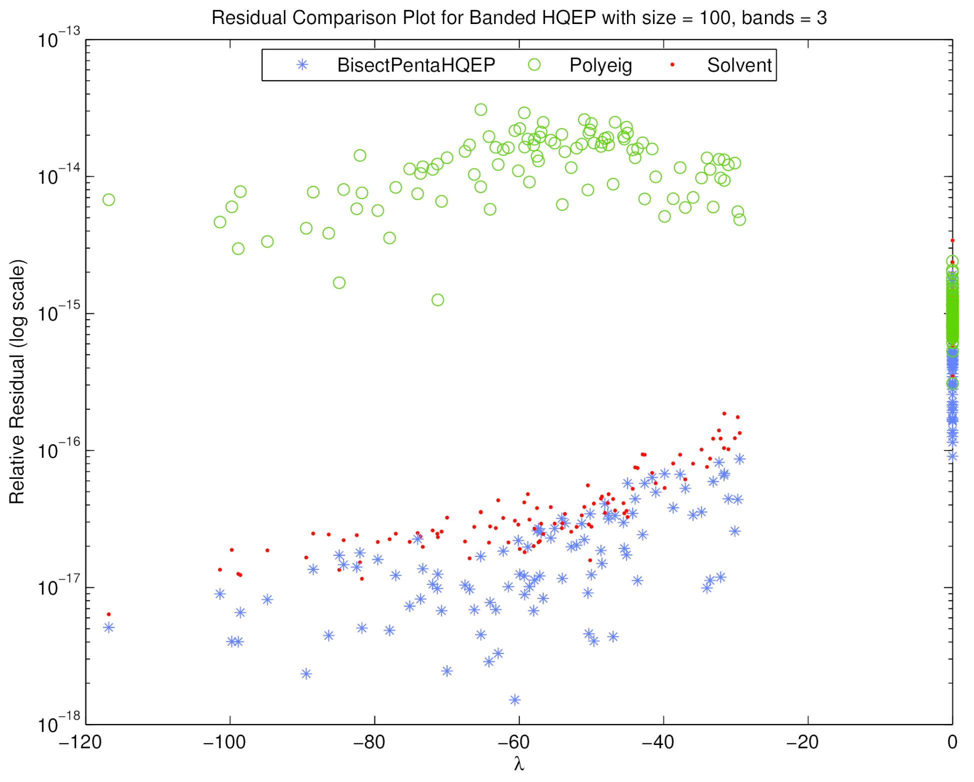
$$R(\lambda) = \frac{\|(\lambda^2 A + \lambda B + C)x\|}{(\lambda^2 \|A\| + |\lambda| \|B\| + \|C\|) \|x\|}.$$

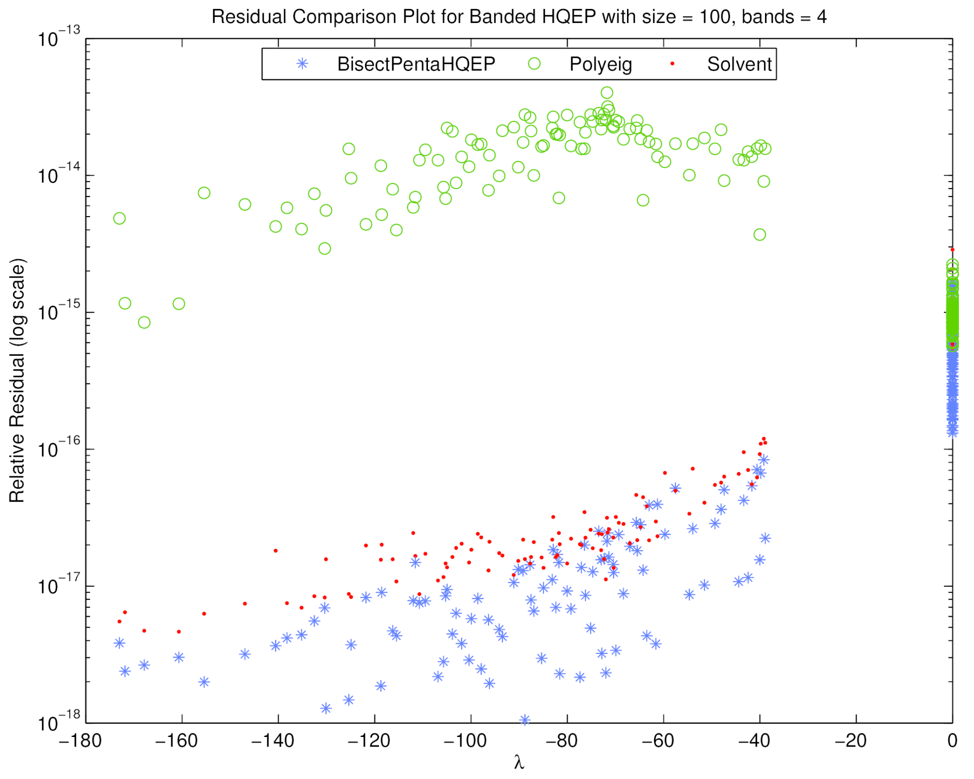
In this section we will compare the residuals for the calculated quadratic eigenpairs using `BisectBandedTHQEP`, Matlab's `Polyeig`, and the Solvent approach procedure `ReturnSolvent` on HQEPs of different sizes to study the effect of size on the residual of different approaches and specially `BisectBandedTHQEP`. The band structure of the introduced HQEPs will vary from tri-diagonal and up, and the purpose is to study the effect of introducing more bands on the residual. The HQEPs will be produced randomly such that the entries of the symmetric A, B, C are uniformly distributed random numbers in the interval $(0, 1)$ using Matlab's `rand` procedure. We will make sure that A, B, C will form an HQEP according to Definition 1.1.1 by performing eigenvalue shifts on them such that A is positive definite and the sufficient condition (2.7) is met.

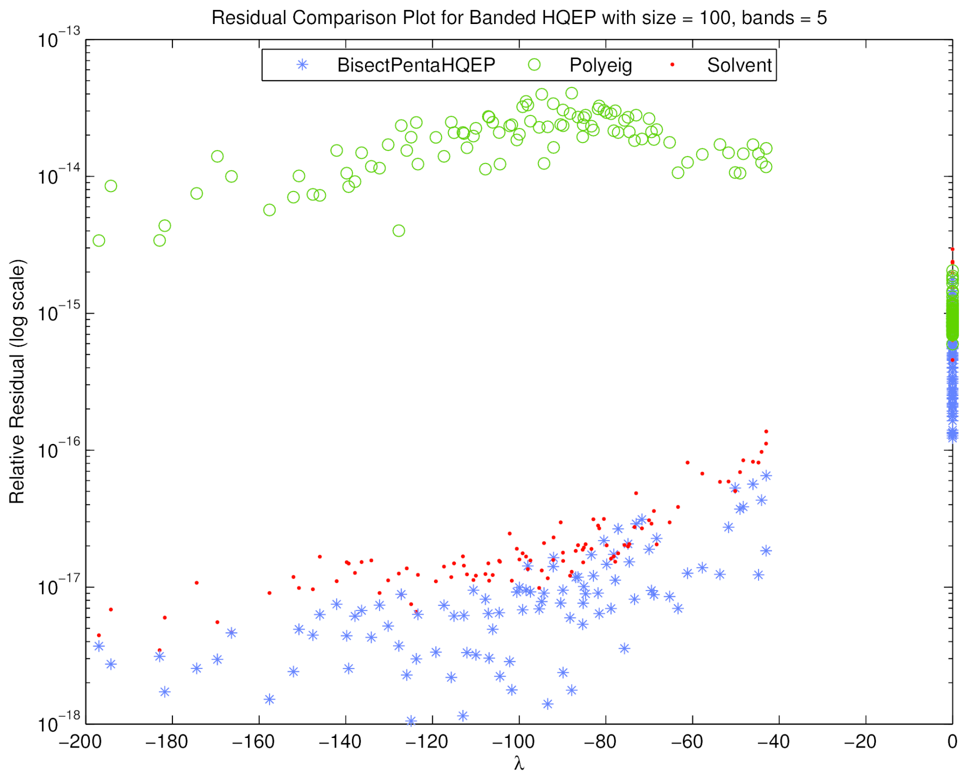
5.3.1 Banded HQEPs of size 100



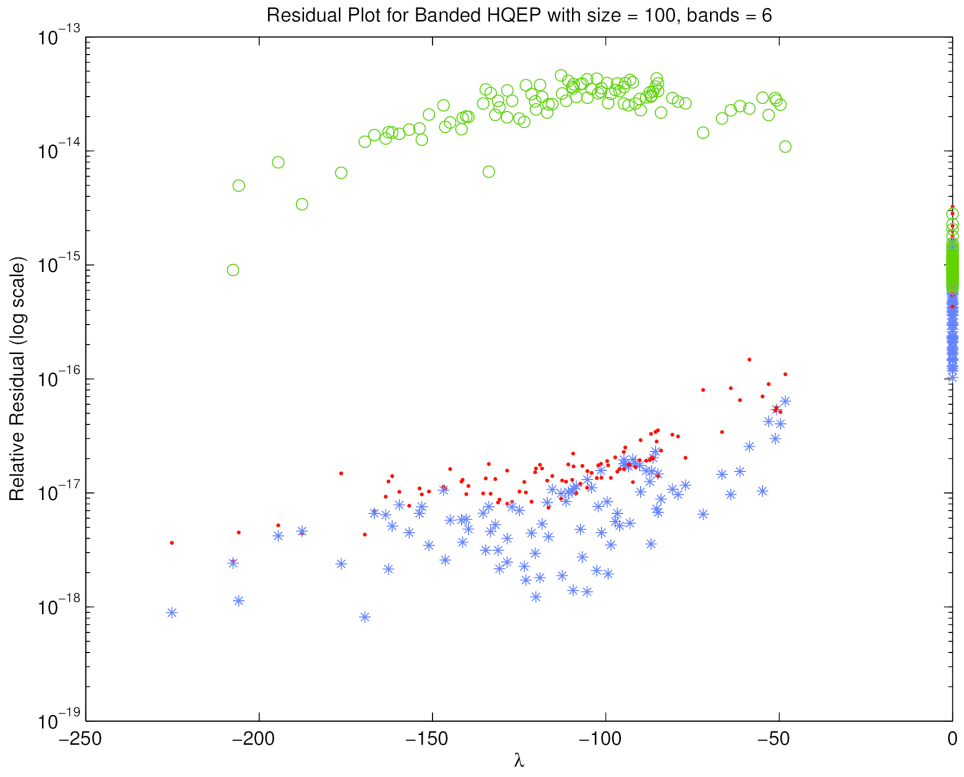



Residual Comparison Plot for Banded HQEP with size = 100, bands = 3
BisectPentaHQEP
Polyeig
Solvent
Relative Residual (log scale)
10^{-13}
10^{-14}
10^{-15}
10^{-16}
10^{-17}
10^{-18}
−120
−100
−80
−60
−40
−20
0
λ


Residual Comparison Plot for Banded HQEP with size = 100, bands = 4
BisectPentaHQEP
Polyeig
Solvent
Relative Residual (log scale)
10^{-13}
10^{-14}
10^{-15}
10^{-16}
10^{-17}
10^{-18}
−180
−160
−140
−120
−100
−80
−60
−40
−20
0
λ

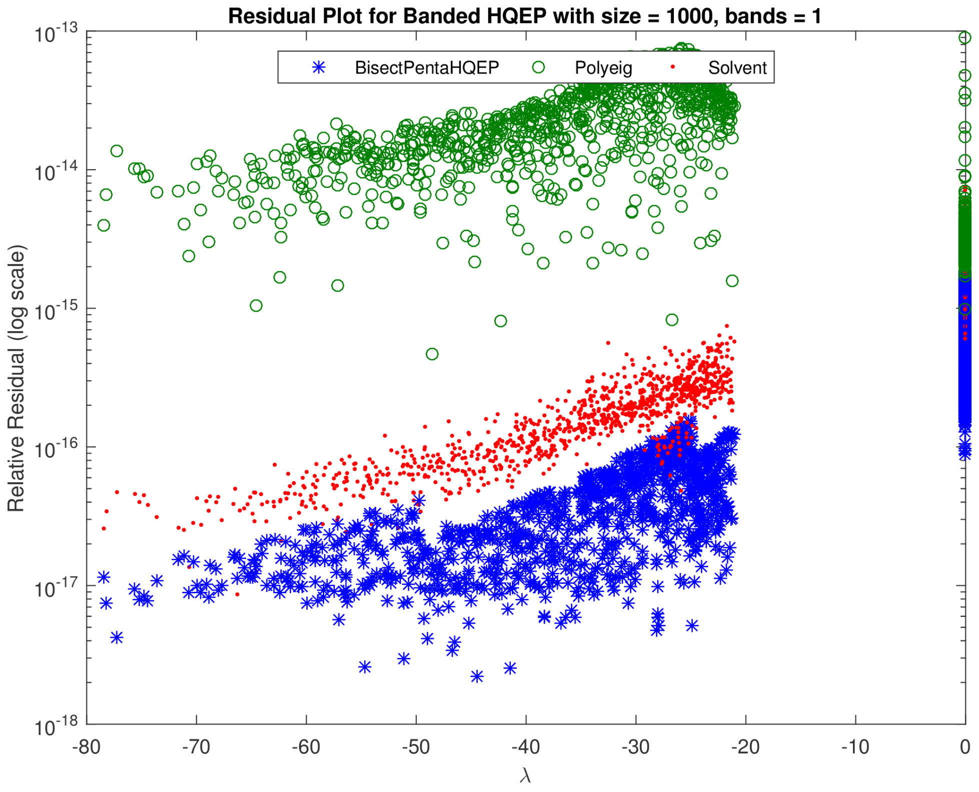


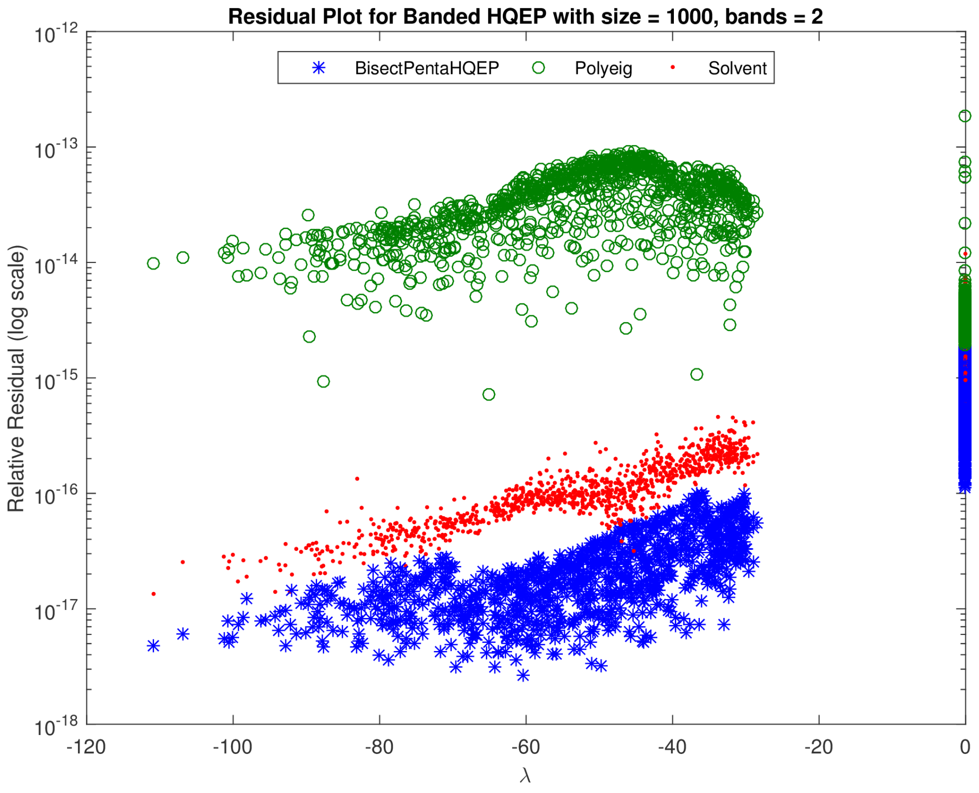
Residual Comparison Plot for Banded HQEP with size = 100, bands = 5
BisectPentaHQEP
Polyeig
Solvent
Relative Residual (log scale)
λ
10^{-13}
10^{-14}
10^{-15}
10^{-16}
10^{-17}
10^{-18}
−200
−180
−160
−140
−120
−100
−80
−60
−40
−20
0

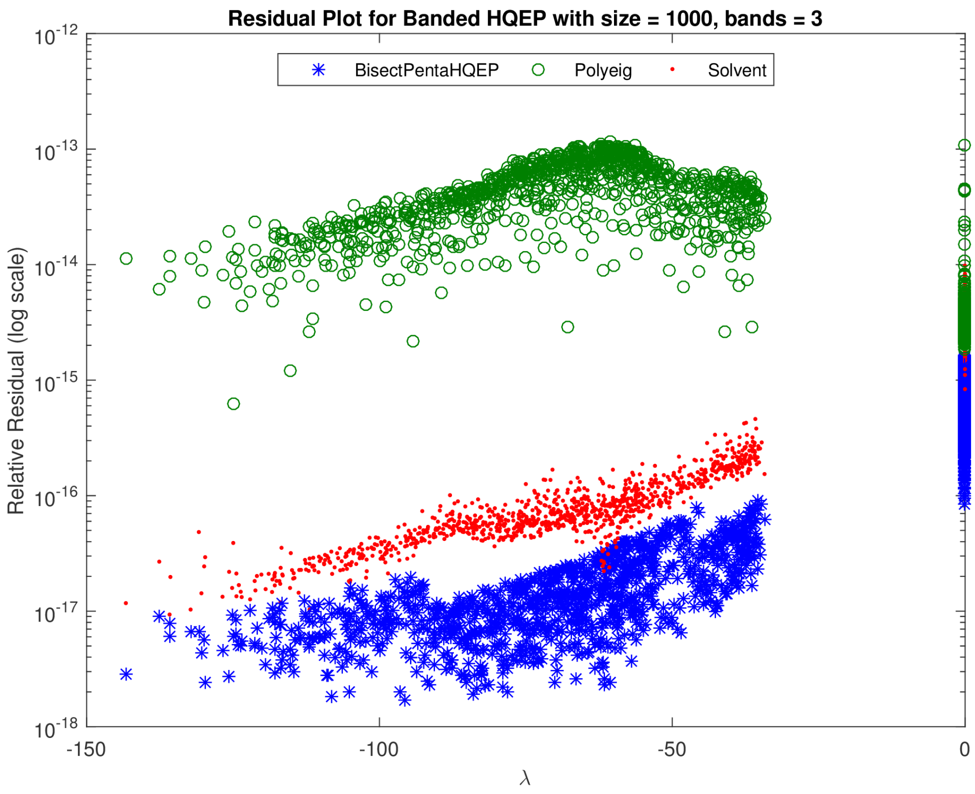


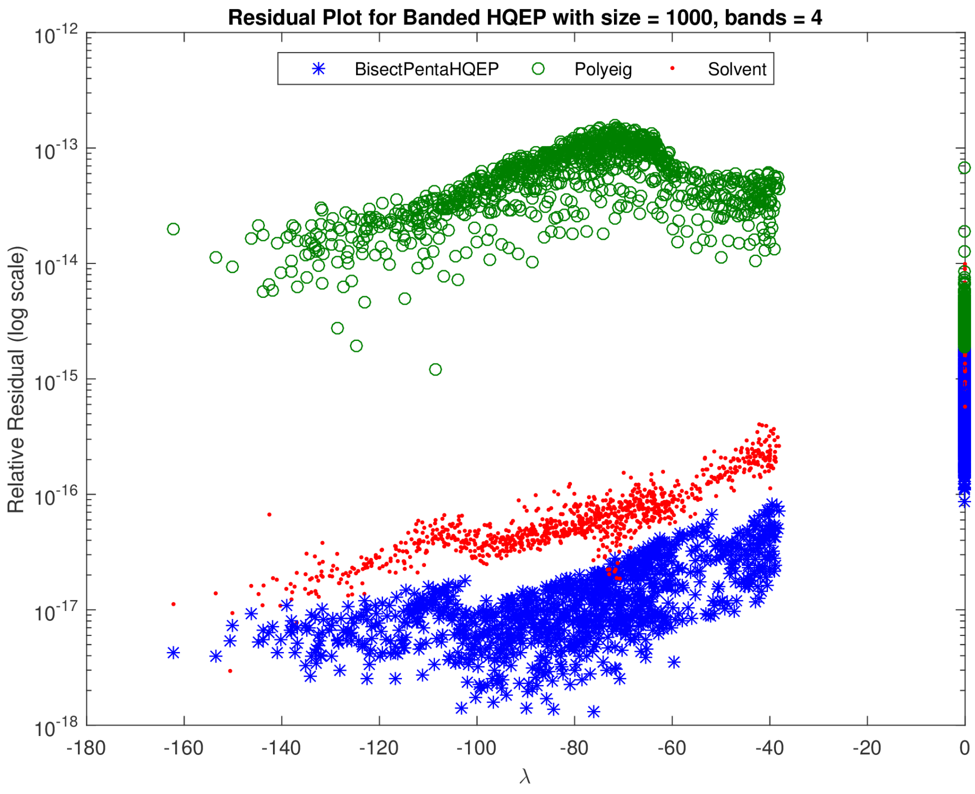
Residual Plot for Banded HQEP with size = 100, bands = 6
Relative Residual (log scale)
λ
10^{-13}
10^{-14}
10^{-15}
10^{-16}
10^{-17}
10^{-18}
10^{-19}
−250
−200
−150
−100
−50
0

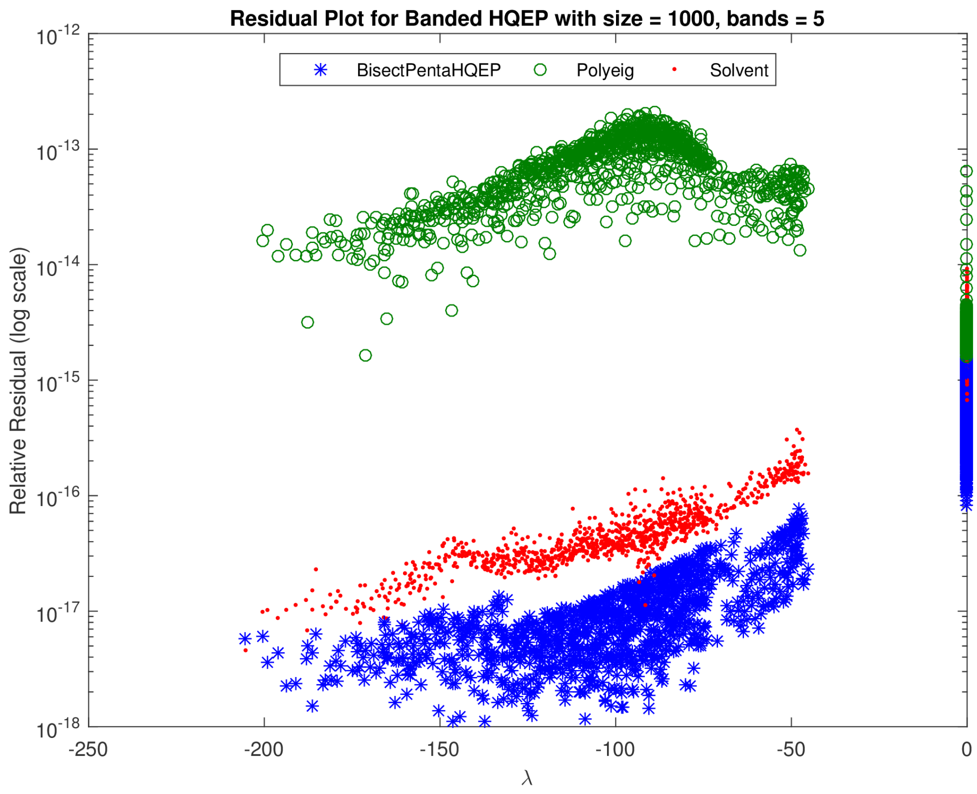
5.3.2 Banded HQEPs of size 1000






Residual Plot for Banded HQEP with size = 1000, bands = 3
BisectPentaHQEP
Polyeig
Solvent
Relative Residual (log scale)
10^{-12}
10^{-13}
10^{-14}
10^{-15}
10^{-16}
10^{-17}
10^{-18}
-150
-100
-50
0
λ

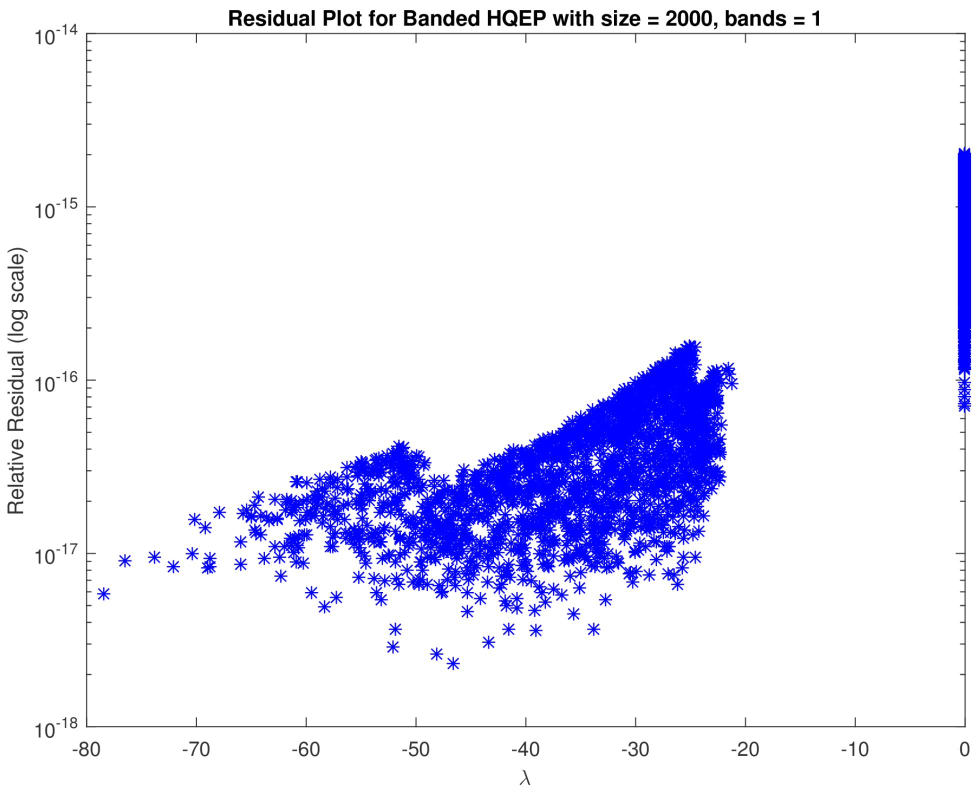
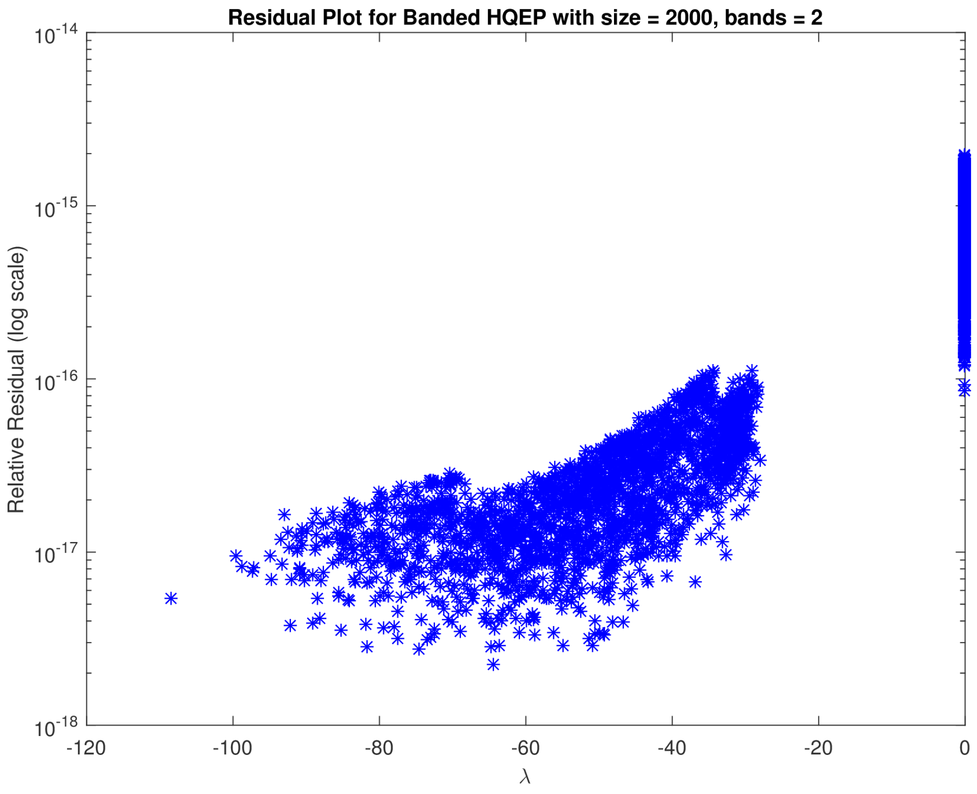

Residual Plot for Banded HQEP with size = 1000, bands = 4
BisectPentaHQEP
Polyeig
Solvent
Relative Residual (log scale)
10^{-12}
10^{-13}
10^{-14}
10^{-15}
10^{-16}
10^{-17}
10^{-18}
-180
-160
-140
-120
-100
-80
-60
-40
-20
0
λ



As can be seen the bisection approach is more accurate and stable for large banded HQEPs even if when the number of bands is growing.

5.3.3 Banded HQEPs of size 2000

In this case we only plot the residual of `BisectBandedTHQEP` since the size of the problem prevented the other 2 approaches from executing due to lack of computer memory. We are trying to show here that due to the procedure's conservation of memory requirements, it can solve large banded HQEPs where the other approaches may fail to execute, and the solutions are still accurate. In fact the residual doesn't grow as the size of the problem grows.


Residual Plot for Banded HQEP with size = 2000, bands = 1
Relative Residual (log scale)
10^-14
10^-15
10^-16
10^-17
10^-18
-80
-70
-60
-50
-40
-30
-20
-10
0
λ

Residual Plot for Banded HQEP with size = 2000, bands = 2
Relative Residual (log scale)
10^-14
10^-15
10^-16
10^-17
10^-18
-120
-100
-80
-60
-40
-20
0
λ

CHAPTER 6

Conclusion

We have shown in this study that the bisection method has many advantages over the other approaches in solving hyperbolic quadratic eigenvalue problems especially when the problem is banded. The bisection approach is more accurate and stable when it comes to the size of the problem and the number of bands in the banded HQEPs. It requires less memory storage to produce the approximated quadratic eigenvalues. The method worked on the problem directly and without introducing a linear pencil with double size. The method can be easily parallelized for parallel computing.

REFERENCES

[1] C.-H. Guo and P. Lancaster, “Algorithms for hyperbolic eigenvalue problems,” *Mathematics Of Computation*, vol. 74, no. 252, pp. 1777–1791, 2005.

[2] B. Meini, “Efficient computation of the extreme solutions of $X + AX^{-1}A = Q$ and $X - AX^{-1}A = Q$,” *Mathematics of Computation*, vol. 71, no. 239, pp. 1189–1204, 2002.

[3] J. W. Demmel, *Applied Numerical Linear Algebra*, 1st ed. SIAM, 1997.

[4] A. F. ACKER, “Absolute continuity of eigenvectors of time-varying operators,” *American Mathematical Society*, vol. 42, no. 1, pp. 198–201, January 1974.

[5] H. Baumgärtel, *Analytical Perturbation Theory for Matrices and Operators*. Basel: Birkhäuser, 1985.

[6] K. R. Parthasarathy, “Eigenvalues of matrix-valued analytic maps.”

[7] T. Kato, *Perturbation Theory for Linear Operators*, 2nd ed. Berlin: Springer-Verlag, 1970.

[8] P. L. I. Gohberg and L. Rodman, *Matrix Polynomials*, 1982.

[9] N. J. Higham, D. S. Mackey, and F. Tisseur, “Definite matrix polynomials and their linearization by definite pencils,” *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 2, pp. 478–502, 2009.

[10] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed. New York: Academic Press, 1985.

[11] A. S. Markus, *Introduction to the Spectral Theory of Polynomial Operator Pencils*. AMS, 1988.

[12] A. M. Farah, "Generalized and Quadratic Eigenvalue Problems with Hermitian Matrices," Master's thesis, School of Mathematics and Statistics The University of Birmingham, Birmingham B15 2TT U.K., 2 2012, e-theses repository.

[13] Barkwell and Lancaster, "Overdamped and gyroscopic vibrating systems," *Journal of Applied Mechanics*, vol. 59, no. 1, pp. 176–181, 1992.

[14] P. Lancaster and Q. Ye, "Variational properties and rayleigh quotient algorithms for symmetric matrix pencils," in *The Gohberg Anniversary Collection*, ser. Operator Theory: Advances and Applications, H. Dym, S. Goldberg, M. Kaashoek, and P. Lancaster, Eds. Birkhäuser Basel, 1989, vol. 40/41, pp. 247–278.

[15] X. LIANG and R.-C. LI, "The hyperbolic quadratic eigenvalue problem," *Forum of Mathematics, Sigma*, vol. 3, p. e13 (93 pages), 2015.

[16] P. Lancaster, *Lambda-matrices and Vibrating Systems.* Mineola, New York: Dover Publications, Inc., 2002.

[17] N. J. Higham, F. Tisseur, and P. M. V. Dooren, "Detecting a definite hermitian pair and a hyperbolic or elliptic quadratic eigenvalue problem, and associated nearness problems," *LINEAR ALGEBRA AND ITS APPLICATIONS*, vol. 351, 2002.

[18] "Numerical solutions for large sparse quadratic eigenvalue problems," *Linear Algebra and its Applications*, vol. 225, pp. 57 – 89, 1993.

[19] B. Plestenjak, "Numerical methods for the tridiagonal hyperbolic quadratic eigenvalue problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 4, pp. 1157–1172, 2006.

[20] C. S. Burrus, J. W. Fox, G. A. Sitton, and S. Treitel, "Horner's method for evaluating and deflating polynomials," *DSP Software Notes, Rice University, Nov*, vol. 26, 2003.

[21] N. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed. Society for Industrial and Applied Mathematics, 2002, ch. 11, pp. 213–229.

[22] H. R. Schwarz, *Linear Algebra*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1971, ch. Tridiagonalization of a Symmetric Band Matrix, pp. 273–283.

BIOGRAPHICAL STATEMENT

Ahmed T, Ali was born in Alexandria, Egypt, in 1976. He received his B.S. , M.A. in Computer Science, and M.A. in Economics from Brooklyn College in 2003, 2005, and 2009 respectively, and Ph.D. degrees from The University of Texas at Arlington in 2016 in Mathematics.