Eigenvalue approximations of the wave equation with local Kelvin-Voigt damping

A PROJECT
SUBMITTED TO THE FACULTY OF THE DEPARTMENT OF
MATHEMATICS & STATISTICS
OF THE UNIVERSITY OF MINNESOTA DULUTH
BY

Christopher Boamah-Mensah

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Advisor: **Steven A. Trogdon**

June, 2015

Acknowledgements

I would like to thank my advisor for all his help.

Abstract

Eigenvalue approximations of the wave equation with local Kelvin-Voigt damping are presented using the well known Chebyshev-Tau spectral method. The problem is formulated in two ways: the first is on one spatial domain while the second is on two spatial domains. Several eigenvalue problems for each method were solved and compared. In general, low frequency eigenvalues were the same for both methods. A brief discussion of inaccurate eigenvalue approximations is also given.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A presentation of the motivation and some known results from theoretical analysis of the problem is given in this chapter.

1.1 Motivation

We will begin with the one-dimensional elastic wave equation with Kelvin-Voigt damping:

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} = \dfrac{\partial}{\partial x}\left[\dfrac{\partial u}{\partial x} + \beta(x)\dfrac{\partial^2 u}{\partial x \partial t}\right] & t > 0, \quad -1 < x < 1, \\ u(t,-1) = u(t,1) = 0 & t \geq 0, \\ u(0,x) = u_0(x),\ \partial u(0,x)/\partial t = u_1(x) & -1 \leq x \leq 1. \end{cases} \tag{1.1}$$

$\beta(x) \in C^1[-1,1]$. Additionally,

$$\beta(x) = \begin{cases} 0, & x \in [-1,0], \\ a(x), & x \in (0,1], \end{cases} \tag{1.2}$$

where $a(x) > 0$ for $x > 0$ and $a(x) \in C[0,1]$. (1.1) generally models the longitudinal motion of a beam made with a non-uniform material or the transverse motion of an string made of two different materials. Engineers at Ford have created a damping patch made of an aluminum foil layer, a viscoelastic layer and an adhesive layer which they place on an elastic plate to analyze natural frequencies and other frequency modes [1]. The solution of the eigenvalue problem of (1.1) may provide analytic results that could be verified experimentally by the engineers. Another application is the use of piezoceramic

patches as sensors on structures [1]. The presence of these patches introduce discontinuities in material properties of the structures being analyzed. The displacements and frequencies of such structure systems can be theoretically analyzed with (1.1).

1.2 Known results

The main goal of this project is to approximate the eigenvalues associated with (1.1) which will give insight to the dynamic stability of the problem. We present some well known results about stability and the eigenvalues of (1.1). The energy ($E(t)$) of (1.1) is defined as

$$E(t) = \frac{1}{2}\int_0^1 \left[\left|\frac{\partial u}{\partial t}\right|^2 + \left|\frac{\partial u}{\partial x}\right|^2\right] \mathrm{d}x. \tag{1.3}$$

For exponential stability of (1.1), $a(\cdot)$ is assumed to satisfy the following [2]:

(A) $a(\cdot) \in C^1[0,1]$, $a(0) = a'(0) = 0$, $a(x) > 0$ for all $x \in (0,1]$.

(B) $\exists a_0 > 0$ such that $\displaystyle\int_0^x \frac{|a'(s)|^2}{a(s)}\mathrm{d}s \leq a_0|a'(x)|$ for all $x \in [0,1]$.

Theorem 1.2.1 (Zhang [2])**.** *Suppose that function $a(\cdot)$ satisfies (A) and (B). Then the energy of system (1.1) is exponentially stable, i.e., $\exists C, \delta > 0$ such that*

$$E(t) \leq Ce^{-\delta t}E(0), \quad \forall t \geq 0.$$

The eigenvalue problem associated with (1.1) can be obtained by seeking a product solution of the form $e^{\lambda t}u(x)$ to give

$$\lambda^2 u = u'' + \lambda(\beta(x)u')' \tag{1.4}$$

with boundary conditions

$$u(-1) = u(1) = 0. \tag{1.5}$$

It is also assumed [3] that at $x = 0$

$$\lim_{x\to 0^+} \frac{\beta'(x)}{x^p} = k > 0$$

for some $p > 0$.

Theorem 1.2.2 (Renardy [3])**.** *Let λ_n, u_n be eigenvalues and corresponding eigenfunctions for (1.4), (1.5). Assume that $|\lambda_n| \to \infty$. Then* $\mathrm{Re}\lambda_n \to -\infty$.

Chapter 2

Formulation

In this chapter we discretize (1.1) using Chebyshev polynomials and seek approximate solutions to the eigenvalues using the tau method. The problem is formulated in two ways. The first is to consider the problem on a single spatial domain. The second is to break the problem into two related problems on two spatial domains.

2.1 Method I

To solve a discrete eigenvalue problem, we start by transforming (1.1) to a system of first order equations as follows:

$$\begin{cases} \dfrac{\partial u}{\partial t} &= v, \\ \\ \dfrac{\partial v}{\partial t} &= \dfrac{\partial}{\partial x}\left[\dfrac{\partial u}{\partial x} + \beta(x)\dfrac{\partial v}{\partial x}\right]. \end{cases} \tag{2.1}$$

Next we expand u, $\beta(x)$, and v using a truncated series of Chebyshev polynomials as

$$u^N = \sum_{k=0}^{N} a_k(t)T_k(x) \quad \beta^N = \sum_{\ell=0}^{N} b_\ell T_\ell(x) \quad v^N = \sum_{k=0}^{N} e_k(t)T_k(x) \tag{2.2}$$

respectively. Substituting (2.2) into (2.1) yields

$$\begin{cases} \dfrac{\partial u^N}{\partial t} &= v^N, \\ \\ \dfrac{\partial v^N}{\partial t} &= \dfrac{\partial}{\partial x}\left[\dfrac{\partial u^N}{\partial x} + \beta^N \dfrac{\partial v^N}{\partial x}\right]. \end{cases} \tag{2.3}$$

Multiplying (2.3) by $T_\ell(x)/\sqrt{1-x^2}$ and integrating from -1 to 1 with respect to x results in the weak form of (2.1) as follows

$$\begin{cases} \displaystyle\int_{-1}^{1} \frac{\partial u^N}{\partial t}\frac{T_\ell(x)}{\sqrt{1-x^2}}\mathrm{d}x &= \displaystyle\int_{-1}^{1} v^N \frac{T_\ell(x)}{\sqrt{1-x^2}}\mathrm{d}x, \\ \\ \displaystyle\int_{-1}^{1} \frac{\partial v^N}{\partial t}\frac{T_\ell(x)}{\sqrt{1-x^2}}\mathrm{d}x &= \displaystyle\int_{-1}^{1} \frac{\partial}{\partial x}\left[\frac{\partial u^N}{\partial x} + \beta^N \frac{\partial v^N}{\partial x}\right]\frac{T_\ell(x)}{\sqrt{1-x^2}}\mathrm{d}x. \end{cases} \tag{2.4}$$

Using (A.2) and (A.9), (2.4) reduces to

$$\begin{cases} \dfrac{\mathrm{d}a_k}{\mathrm{d}t} &= e_k, \\ \\ \dfrac{\mathrm{d}e_k}{\mathrm{d}t} &= a_k^{(2)} + \dfrac{2}{\pi c_k}\displaystyle\int_{-1}^{1} \frac{\partial}{\partial x}\left(\beta^N \frac{\partial v^N}{\partial x}\right)\frac{T_\ell(x)}{\sqrt{1-x^2}}\mathrm{d}x \end{cases} \qquad k = 0, 1, 2, \ldots, N-2 \tag{2.5}$$

where

$$\begin{aligned} \beta^N \frac{\partial v^N}{\partial x} &= \sum_{\ell=0}^{N} b_\ell T_\ell(x) \frac{\partial}{\partial x}\sum_{k=0}^{N} e_k(t) T_k(x) \\ &= \sum_{\ell=0}^{N} b_\ell T_\ell(x) \sum_{k=0}^{N} e_k^{(1)}(t) T_k(x) \\ &= \frac{1}{2}\sum_{\ell=0}^{N}\sum_{k=0}^{N} b_\ell e_k^{(1)}(t)[T_{k+\ell}(x) + T_{|k-\ell|}(x)] \\ &= \frac{1}{2}\sum_{k=0}^{N} B_{\ell k} e_k^{(1)}(t) T_k(x) \end{aligned} \tag{2.6}$$

after using (A.5), (A.7), and doing some algebra. $B_{\ell k}$ in (2.6) represents the elements of a matrix with coefficients of b_k. The inner product of polynomials with index beyond N and polynomials with indices from 0 to N is zero. Substituting (2.6) into the integral

in (2.5) and switching the integral and derivative we get that

$$\frac{2}{\pi c_k}\int_{-1}^{1}\frac{\partial}{\partial x}\left(\beta^N\frac{\partial v^N}{\partial x}\right)\frac{T_\ell(x)}{\sqrt{1-x^2}}\mathrm{d}x=\left[\frac{1}{2}\sum_{k=0}^{N}B_{\ell k}e_k^{(1)}(t)\right]^{(1)}. \tag{2.7}$$

Let $\mathbf{R}_1 \equiv \begin{bmatrix} T_0(1) \\ T_1(1) \\ T_2(1) \\ \vdots \\ T_N(1) \end{bmatrix}$, $\mathbf{R}_{-1} \equiv \begin{bmatrix} T_0(-1) \\ T_1(-1) \\ T_2(-1) \\ \vdots \\ T_N(-1) \end{bmatrix}$, and $\mathbf{a} \equiv \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}$ (Note that $T_k(\pm 1)$ is given in (A.4)). From u^N in (2.2) and the boundary conditions in (1.1) the following is obtained:

$$\begin{aligned} \mathbf{a}^T\mathbf{R}_1 &= 0, \\ \mathbf{a}^T\mathbf{R}_{-1} &= 0. \end{aligned} \tag{2.8}$$

Equation (2.8) may be partitioned as follows

$$\begin{bmatrix} \mathbf{a}_1^T & \mathbf{a}_2^T \end{bmatrix}\begin{bmatrix} \mathbf{R}_{1,1} & \mathbf{R}_{-1,1} \\ \mathbf{R}_{1,2} & \mathbf{R}_{-1,2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \tag{2.9}$$

where $\mathbf{a}_1$ and $\mathbf{a}_2$ represent column vectors with the first $N-1$ and last two elements of $\mathbf{a}$ respectively. $\mathbf{R}_{1,1}$ and $\mathbf{R}_{1,2}$ represent vectors of the first $N-1$ and last two elements of $\mathbf{R}_1$ respectively. A similar statement can be made for $\mathbf{R}_{-1,1}$ and $\mathbf{R}_{-1,2}$ relative to $\mathbf{R}_{-1}$. Equation (2.9) may be written as

$$\mathbf{a}_1^T\mathbf{Q}_1 + \mathbf{a}_2^T\mathbf{Q}_2 = 0, \tag{2.10}$$

where $\mathbf{Q}_1 \equiv \begin{bmatrix} \mathbf{R}_{1,1} & \mathbf{R}_{-1,1} \end{bmatrix}$ and $\mathbf{Q}_2 \equiv \begin{bmatrix} \mathbf{R}_{1,2} & \mathbf{R}_{-1,2} \end{bmatrix}$. Assuming $\mathbf{Q}_2$ is invertible

$$\mathbf{a}_2 = -[\mathbf{Q}_1\mathbf{Q}_2^{-1}]^T\mathbf{a}_1.$$

Thus, replacing the last two elements of $\mathbf{a}$ with $\mathbf{a}_2$ we get

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_1 \\ -[\mathbf{Q}_1\mathbf{Q}_2^{-1}]^T\mathbf{a}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ -[\mathbf{Q}_1\mathbf{Q}_2^{-1}]^T \end{bmatrix}\mathbf{a}_1$$

where $\mathbf{I}$ is an $(N-1)\times(N-1)$ identity matrix. Letting

$$\mathbf{S} \equiv \begin{bmatrix} \mathbf{I} \\ -[\mathbf{Q}_1\mathbf{Q}_2^{-1}]^T \end{bmatrix}$$

results in

$$\mathbf{a} = \mathbf{S}\mathbf{a}_1. \tag{2.11}$$

Assuming v is sufficiently smooth, we can show from (2.1) that v also has homogeneous Dirichlet boundary conditions. Thus, for v^N in (2.2) we obtain

$$\mathbf{e} = \mathbf{S}\mathbf{e}_1 \tag{2.12}$$

in a similar way as (2.11). The convolution sum of (2.7) in matrix-vector form is

$$\mathbf{M}\left[\left[\begin{array}{cccccccc}
b_0 & & & & & & & \\
b_1 & b_0 & & & & & & \\
b_2 & \frac{b_1}{2} & b_0 & & & & & \\
b_3 & \frac{b_2}{2} & \frac{b_1}{2} & b_0 & & & & \\
\vdots & \frac{b_3}{2} & \frac{b_2}{2} & \frac{b_1}{2} & \ddots & & & \\
 & \vdots & \frac{b_3}{2} & \frac{b_2}{2} & \ddots & & & \\
 & & \vdots & & \ddots & & & \\
b_N & \frac{b_{N-1}}{2} & \frac{b_{N-2}}{2} & \cdots & & \frac{b_2}{2} & \frac{b_1}{2} & b_0
\end{array}\right] + \left[\begin{array}{ccccccc}
0 & \frac{b_1}{2} & \frac{b_2}{2} & \frac{b_3}{2} & \cdots & & \frac{b_N}{2} \\
 & 0 & \frac{b_1}{2} & \frac{b_2}{2} & \frac{b_3}{2} & \cdots & \\
 & & 0 & \frac{b_1}{2} & \frac{b_2}{2} & \frac{b_3}{2} & \cdots \\
 & & & \ddots & \ddots & \ddots & \ddots & \vdots \\
 & & & & & & \frac{b_3}{2} \\
 & & & & & & \frac{b_2}{2} \\
 & & & & & & \frac{b_1}{2} \\
 & & & & & & 0
\end{array}\right] + \right.$$

$$\left.\left[\begin{array}{ccccccc}
0 & 0 & 0 & \cdots & & & 0 \\
0 & \frac{b_2}{2} & \frac{b_3}{2} & \frac{b_4}{2} & \frac{b_5}{2} & \cdots & \frac{b_N}{2} & 0 \\
0 & \frac{b_3}{2} & \frac{b_4}{2} & \frac{b_5}{2} & & \frac{b_N}{2} & & 0 \\
\vdots & \frac{b_4}{2} & \frac{b_5}{2} & & \frac{b_N}{2} & & & \vdots \\
 & \frac{b_5}{2} & & \iddots & & & & \\
 & \vdots & & & & & & \\
 & \frac{b_N}{2} & & & & & & \\
0 & 0 & 0 & \cdots & & & & 0
\end{array}\right]\right]\left[\begin{array}{c}
e_0^{(1)} \\ e_1^{(1)} \\ e_2^{(1)} \\ e_3^{(1)} \\ \vdots \\ \\ \\ e_N^{(1)}
\end{array}\right], \tag{2.13}$$

where $\mathbf{M}$ is given by (A.12). Writing (2.13) compactly by defining $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{B}_3$ as matrices yields

$$\mathbf{M}(\mathbf{B}_1 + \mathbf{B}_2 + \mathbf{B}_3)\mathbf{M}\mathbf{e}. \tag{2.14}$$

Now (2.5) in matrix-vector notation becomes

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\mathbf{a}_1 &= \mathbf{e}_1 \\ \\ \dfrac{\mathrm{d}}{\mathrm{d}t}\mathbf{e}_1 &= \mathbf{M}^2\mathbf{S}\mathbf{a}_1 + \mathbf{M}(\mathbf{B}_1+\mathbf{B}_2+\mathbf{B}_3)\mathbf{M}\mathbf{S}\mathbf{e}_1 \end{cases} \tag{2.15}$$

Finally, the eigenvalue problem for (2.15) is obtained by seeking solutions of the form $e^{\lambda t}\begin{bmatrix} \tilde{\mathbf{a}}_1 \\ \tilde{\mathbf{e}}_1 \end{bmatrix}$ which results in

$$\lambda \begin{bmatrix} \tilde{\mathbf{a}}_1 \\ \tilde{\mathbf{e}}_1 \end{bmatrix} = \mathbf{N} \begin{bmatrix} \tilde{\mathbf{a}}_1 \\ \tilde{\mathbf{e}}_1 \end{bmatrix}, \tag{2.16}$$

where

$$\mathbf{N} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{M}^2\mathbf{S} & \mathbf{M}(\mathbf{B}_1+\mathbf{B}_2+\mathbf{B}_3)\mathbf{M}\mathbf{S} \end{bmatrix}$$

Note that $\mathbf{M}^2$ and $\mathbf{M}(\mathbf{B}_1+\mathbf{B}_2+\mathbf{B}_3)\mathbf{M}$ are $(N-1)\times(N+1)$ matrices.

2.2 Method II

We begin by splitting (1.1) on two spatial domains.

$$\frac{\partial^2 u_-}{\partial t^2} - \frac{\partial^2 u_-}{\partial x^2} = 0 \quad t>0, \quad -1<x<0 \tag{2.17}$$

$$u_-(-1,t)=0, \quad u_-(0,t)=?$$

$$\frac{\partial^2 u_+}{\partial t^2} - \frac{\partial}{\partial x}\left[\frac{\partial u_+}{\partial x} + a(x)\frac{\partial^2 u_+}{\partial x \partial t}\right] = 0 \quad t>0, \quad 0<x<1 \tag{2.18}$$

$$u_+(1,t)=0, \quad u_+(0,t)=?$$

u_- denotes u on $(-1,0)$ while u_+ denotes u on $(0,1)$. Initial conditions have been ignored because they are not needed for the eigenvalue problem. We will also enforce certain boundary conditions at $x=0$ later on. Using a linear map ($\xi = 2x+1$) from $x\in(-1,0)$ to $\xi\in(-1,1)$, $\tau = 2t$, and some calculus, (2.17) becomes

$$\frac{\partial^2 \hat{u}_-}{\partial \tau^2} - \frac{\partial^2 \hat{u}_-}{\partial \xi^2} = 0, \tag{2.19}$$

where $u_-(x,t) \equiv \hat{u}_-(\xi,\tau)$. In a similar way, we use a linear map ($\eta = 2x-1$) from $x \in (0,1)$ to $\eta \in (-1,1)$ and $\tau = 2t$ to transform (2.18) to

$$\frac{\partial^2 \hat{u}_+}{\partial \tau^2} - \frac{\partial}{\partial \eta}\left[\frac{\partial \hat{u}_+}{\partial \eta} + 2a\left(\frac{\eta+1}{2}\right)\frac{\partial^2 \hat{u}_+}{\partial \eta \partial \tau}\right] = 0, \tag{2.20}$$

where $u_+(x,t) \equiv \hat{u}_+(\eta,\tau)$. Dropping "ˆ" for convenience, (2.19) and (2.20) become

$$\begin{cases} \dfrac{\partial^2 u_-}{\partial \tau^2} - \dfrac{\partial^2 u_-}{\partial \xi^2} = 0, \\[2ex] \dfrac{\partial^2 u_+}{\partial \tau^2} - \dfrac{\partial}{\partial \eta}\left[\dfrac{\partial u_+}{\partial \eta} + 2a\left(\dfrac{\eta+1}{2}\right)\dfrac{\partial^2 u_+}{\partial \eta \partial \tau}\right] = 0. \end{cases} \tag{2.21}$$

From continuity of u and $\dfrac{\partial u}{\partial x} + \beta(x)\dfrac{\partial^2 u}{\partial x \partial t}$ at $x = 0$ we get that

$$u_-(1,\tau) = u_+(-1,\tau), \tag{2.22}$$

$$\frac{\partial u_-}{\partial \xi}(1,\tau) = \frac{\partial u_+}{\partial \eta}(-1,\tau), \tag{2.23}$$

which takes care of the boundaries conditions for u_- and u_+ at $x = 0$. The other two boundary conditions are

$$u_-(-1,\tau) = 0, \tag{2.24}$$

$$u_+(1,\tau) = 0. \tag{2.25}$$

Transforming (2.21) to a system of first order equations yields

$$\begin{cases} \dfrac{\partial u_-}{\partial \tau} = v_-, \\[2ex] \dfrac{\partial u_+}{\partial \tau} = v_+, \\[2ex] \dfrac{\partial v_-}{\partial \tau} = \dfrac{\partial^2 u_-}{\partial \xi^2}, \\[2ex] \dfrac{\partial v_+}{\partial \tau} = \dfrac{\partial}{\partial \eta}\left[\dfrac{\partial u_+}{\partial \eta} + \alpha(\eta)\dfrac{\partial v_+}{\partial \eta}\right], \end{cases} \tag{2.26}$$

where $\alpha(\eta) \equiv 2a(\frac{\eta+1}{2})$. Expanding u_-, v_-, $\alpha(\eta)$, u_+, and v_+ using a truncated series of Chebyshev polynomials results in

$$u_-^N = \sum_{k=0}^{N} a_k(\tau) T_k(\xi), \tag{2.27}$$

$$v_-^N = \sum_{k=0}^{N} e_k(\tau) T_k(\xi), \tag{2.28}$$

$$\alpha^N = \sum_{k=0}^{N} d_k T_k(\eta), \tag{2.29}$$

$$u_+^N = \sum_{k=0}^{N} f_k(\tau) T_k(\eta), \tag{2.30}$$

$$v_+^N = \sum_{k=0}^{N} g_k(\tau) T_k(\eta). \tag{2.31}$$

Substituting (2.27), (2.28), (2.29), (2.30), and (2.31) into (2.26) we get

$$\begin{cases} \dfrac{\partial u_-^N}{\partial \tau} &= v_-^N, \\[2ex] \dfrac{\partial u_+^N}{\partial \tau} &= v_+^N, \\[2ex] \dfrac{\partial v_-^N}{\partial \tau} &= \dfrac{\partial^2 u_-^N}{\partial \xi^2}, \\[2ex] \dfrac{\partial v_+^N}{\partial \tau} &= \dfrac{\partial}{\partial \eta}\left[\dfrac{\partial u_+^N}{\partial \eta} + \alpha^N \dfrac{\partial v_+^N}{\partial \eta}\right]. \end{cases} \tag{2.32}$$

Applying the inner product (A.2) with the appropriate independent variable and using (A.9), (2.32) reduces to

$$\begin{cases} \dfrac{\mathrm{d}a_k}{\mathrm{d}\tau} &= e_k, \\[2ex] \dfrac{\mathrm{d}f_k}{\mathrm{d}\tau} &= g_k, \\[2ex] \dfrac{\mathrm{d}e_k}{\mathrm{d}\tau} &= a_k^{(2)}, \\[2ex] \dfrac{\mathrm{d}g_k}{\mathrm{d}\tau} &= f_k^{(2)} + \dfrac{2}{\pi c_k}\displaystyle\int_{-1}^{1} \dfrac{\partial}{\partial\eta}\left(\alpha^N \dfrac{\partial v_+^N}{\partial\eta}\right) \dfrac{T_\ell(\eta)}{\sqrt{1-\eta^2}}\mathrm{d}\eta, \end{cases} \tag{2.33}$$

where in a similar process of obtaining (2.7)

$$\frac{2}{\pi c_k}\int_{-1}^{1} \frac{\partial}{\partial\eta}\left(\alpha^N \frac{\partial v_+^N}{\partial\eta}\right) \frac{T_\ell(\eta)}{\sqrt{1-\eta^2}}\mathrm{d}\eta = \left[\frac{1}{2}\sum_{k=0}^{N} D_{\ell k} g_k^{(1)}(\tau)\right]^{(1)}. \tag{2.34}$$

$D_{\ell k}$ represents elements of a matrix of coefficients of d_k. From (2.24), (2.22), (2.27), and (2.30) we obtain

$$\mathbf{P}_1\mathbf{a}_1 + \mathbf{P}_2\mathbf{a}_2 = \mathbf{Q}_1\mathbf{f}_1 + \mathbf{Q}_2\mathbf{f}_2, \tag{2.35}$$

where $\mathbf{a}_1$, $\mathbf{f}_1$, are column vectors of the first $N-1$ elements of $\mathbf{a}$ (a_k coefficients in u_-^N) and $\mathbf{f}$ (f_k coefficients in u_+^N) respectively and $\mathbf{a}_2$ and $\mathbf{f}_2$ are the last two elements of $\mathbf{a}$ and $\mathbf{f}$ respectively. The matrices in (2.35) are given by

$$\mathbf{P}_1 = \begin{bmatrix} T_0(-1) & T_1(-1) & T_2(-1) & \cdots & T_{N-2}(-1) \\ T_0(1) & T_1(1) & T_2(1) & \cdots & T_{N-2}(1) \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} T_{N-1}(-1) & T_N(-1) \\ T_{N-1}(1) & T_N(1) \end{bmatrix},$$

$$\mathbf{Q}_1 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ T_0(-1) & T_1(-1) & T_2(-1) & \cdots & T_{N-2}(-1) \end{bmatrix}, \text{ and } \mathbf{Q}_2 = \begin{bmatrix} 0 & 0 \\ T_{N-1}(-1) & T_N(-1) \end{bmatrix}.$$

In a similar way (2.25), (2.23), (2.27), and (2.30) are used to get

$$\hat{\mathbf{Q}}_1\mathbf{f}_1 + \hat{\mathbf{Q}}_2\mathbf{f}_2 = \hat{\mathbf{P}}_1\mathbf{a}_1 + \hat{\mathbf{P}}_2\mathbf{a}_2, \tag{2.36}$$

where

$$\hat{\mathbf{Q}}_1 = \begin{bmatrix} T_0(1) & T_1(1) & T_2(1) & \cdots & T_{N-2}(1) \\ T_0'(-1) & T_1'(-1) & T_2'(-1) & \cdots & T_{N-2}'(-1) \end{bmatrix}, \quad \hat{\mathbf{Q}}_2 = \begin{bmatrix} T_{N-1}(1) & T_N(1) \\ T_{N-1}'(-1) & T_N'(-1) \end{bmatrix},$$

$$\hat{\mathbf{P}}_1 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ T_0'(1) & T_1'(1) & T_2'(1) & \cdots & T_{N-2}'(1) \end{bmatrix}, \text{ and } \hat{\mathbf{P}}_2 = \begin{bmatrix} 0 & 0 \\ T_{N-1}'(1) & T_N'(1) \end{bmatrix}.$$

$T_k'(\pm 1)$ is given in (A.6). Assuming $\hat{\mathbf{Q}}_2$ and $\mathbf{P}_2$ are invertible, we may rearrange equations (2.35) and (2.36) to obtain the following matrix-vector equations.

$$\begin{bmatrix} \mathbf{I} & -\mathbf{P}_2^{-1}\mathbf{Q}_2 \\ -\hat{\mathbf{Q}}_2^{-1}\hat{\mathbf{P}}_2 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{a}_2 \\ \mathbf{f}_2 \end{bmatrix} = \begin{bmatrix} -\mathbf{P}_2^{-1}\mathbf{P}_1 & \mathbf{P}_2^{-1}\mathbf{Q}_1 \\ \hat{\mathbf{Q}}_2^{-1}\hat{\mathbf{P}}_1 & -\hat{\mathbf{Q}}_2^{-1}\hat{\mathbf{Q}}_1 \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{f}_1 \end{bmatrix}, \tag{2.37}$$

where $\mathbf{I}$ is a square identity matrix that is conformable to the matrices in its row and column. By letting

$$\mathbf{K} \equiv \begin{bmatrix} \mathbf{I} & -\mathbf{P}_2^{-1}\mathbf{Q}_2 \\ -\hat{\mathbf{Q}}_2^{-1}\hat{\mathbf{P}}_2 & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{P}_2^{-1}\mathbf{P}_1 & \mathbf{P}_2^{-1}\mathbf{Q}_1 \\ \hat{\mathbf{Q}}_2^{-1}\hat{\mathbf{P}}_1 & -\hat{\mathbf{Q}}_2^{-1}\hat{\mathbf{Q}}_1 \end{bmatrix}$$

(2.37) becomes

$$\begin{bmatrix} \mathbf{a}_2 \\ \mathbf{f}_2 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{f}_1 \end{bmatrix}.$$

Replacing the last two elements of $\mathbf{a}$ and $\mathbf{f}$ we get that

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & & \mathbf{0} \\ & \mathbf{K}_1 & \\ \mathbf{0} & & \mathbf{I} \\ & \mathbf{K}_2 & \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{f}_1 \end{bmatrix},$$

where $\mathbf{K}_1$ and $\mathbf{K}_2$ are the first two and last two rows of $\mathbf{K}$ respectively and $\mathbf{I}$ is an $(N-1) \times (N-1)$ identity matrix. Thus, we get that

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{f} \end{bmatrix} = \mathbf{S} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{f}_1 \end{bmatrix}, \tag{2.38}$$

where

$$\mathbf{S} = \begin{bmatrix} \mathbf{I} & & \mathbf{0} \\ & \mathbf{K}_1 & \\ \mathbf{0} & & \mathbf{I} \\ & \mathbf{K}_2 & \end{bmatrix}.$$

Assuming that v_+ and v_- are smooth enough, we can show that the boundary conditions for v_+ and v_- are similar to that of u_+ and u_-, hence

$$\begin{bmatrix} \mathbf{e} \\ \mathbf{g} \end{bmatrix} = \mathbf{S} \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{g}_1 \end{bmatrix} \tag{2.39}$$

is obtained in the same way as (2.38). The procedure for obtaining a system of equations for the Chebyshev coefficients is similar to that of method I. Thus,

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{a} &= \mathbf{e}, \\[2ex] \dfrac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{f} &= \mathbf{g}, \\[2ex] \dfrac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{e} &= \mathbf{M}^2\mathbf{a}, \\[2ex] \dfrac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{g} &= \mathbf{M}^2\mathbf{f} + \mathbf{M}(\mathbf{D}_1 + \mathbf{D}_2 + \mathbf{D}_3)\mathbf{M}\mathbf{g}. \end{cases} \tag{2.40}$$

By substituting (2.38) and (2.39) into (2.40), we formulate the eigenvalue problem for the system of Chebyshev coefficients as

$$\lambda \begin{bmatrix} \tilde{\mathbf{a}}_1 \\ \tilde{\mathbf{f}}_1 \\ \tilde{\mathbf{e}}_1 \\ \tilde{\mathbf{g}}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{M}_f^2\mathbf{S} & \mathbf{X}\mathbf{S} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{a}}_1 \\ \tilde{\mathbf{f}}_1 \\ \tilde{\mathbf{e}}_1 \\ \tilde{\mathbf{g}}_1 \end{bmatrix}, \tag{2.41}$$

where

$$\mathbf{M}_f^2 = \begin{bmatrix} \mathbf{M}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^2 \end{bmatrix}$$

and

$$\mathbf{X} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}(\mathbf{D}_1 + \mathbf{D}_2 + \mathbf{D}_3)\mathbf{M} \end{bmatrix}.$$

$\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{D}_3$ have the same structure as $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{B}_3$ with the elements of $\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{D}_3$ utilizing the Chebyshev coefficients for α^N. Note that $\mathbf{M}^2$ and $\mathbf{M}(\mathbf{D}_1 + \mathbf{D}_2 + \mathbf{D}_3)\mathbf{M}$ are $(N-1) \times (N+1)$ matrices.

Chapter 3

Results

In this chapter, numerical results for both problems formulated in the previous chapter are presented. All results were obtained from code written in MATLAB®* version R2013b on a 64-bit linux workstation. To solve the eigenvalue problems we first computed the Chebyshev coefficients b_ℓ and d_k for given functions of $\beta(x)$ and $\alpha(\eta)$. Analytically this was done by multiplying $\beta(x) = \sum_{\ell=0}^{N} b_\ell T_\ell(x)$ by $T_k(x)/\sqrt{1-x^2}$ and using the orthogonality property (A.2) to get

$$b_\ell = \frac{2}{\pi c_\ell} \int_{-1}^{1} \beta(x) \frac{T_\ell(x)}{\sqrt{1-x^2}} \mathrm{d}x. \tag{3.1}$$

Similarly,

$$d_k = \frac{2}{\pi c_k} \int_{-1}^{1} \alpha(\eta) \frac{T_k(\eta)}{\sqrt{1-\eta^2}} \mathrm{d}\eta. \tag{3.2}$$

Closed form solutions to (3.1) and (3.2) are not always guaranteed for certain functions so to get the coefficients in another way, numerical integration was used. Using Gaussian quadrature with Gauss-Lobatto nodes and weights,

$$b_\ell \approx \frac{2}{\pi c_\ell} \sum_{\ell=0}^{N} \beta(x_j) T_\ell(x_j) w_j, \qquad j = 0, 1, 2, \ldots, N, \tag{3.3}$$

where $x_j = \cos \frac{\pi j}{N}$ and $w_j = \begin{cases} \pi/2N, & j = 0, N; \\ \pi/N, & j = 1, 2, 3, \ldots, N-1. \end{cases}$

Thus, b_ℓ may be approximated using the inverse Fourier transform. In MATLAB, this

*MATLAB is a registered trademark of The MathWorks, Inc., Web: www.mathworks.com

was done using the built-in `ifft` function. d_k is obtained the same way as b_ℓ.

$$\beta(x) = \begin{cases} 0, & x \in [-1, 0], \\ x^p, & x \in (0, 1] \end{cases} \tag{3.4}$$

and

$$\alpha(\eta) = 2\left(\frac{\eta + 1}{2}\right)^p \tag{3.5}$$

were chosen because it can be shown that they satisfy certain conditions given in theorems 1.2.1 and 1.2.2. $p > 1$. We numerically approximated all the coefficients computed. For $p = 2$, we computed the exact coefficients but there was no significant difference with the approximated coefficients. All eigenvalues were computed with MATLAB's `eigs` function with the option set to compute the smallest eigenvalues in magnitude. The eigenvalues were also computed with machine precision accuracy according to MATLAB's documentation for `eigs`.

3.1 Results for method I

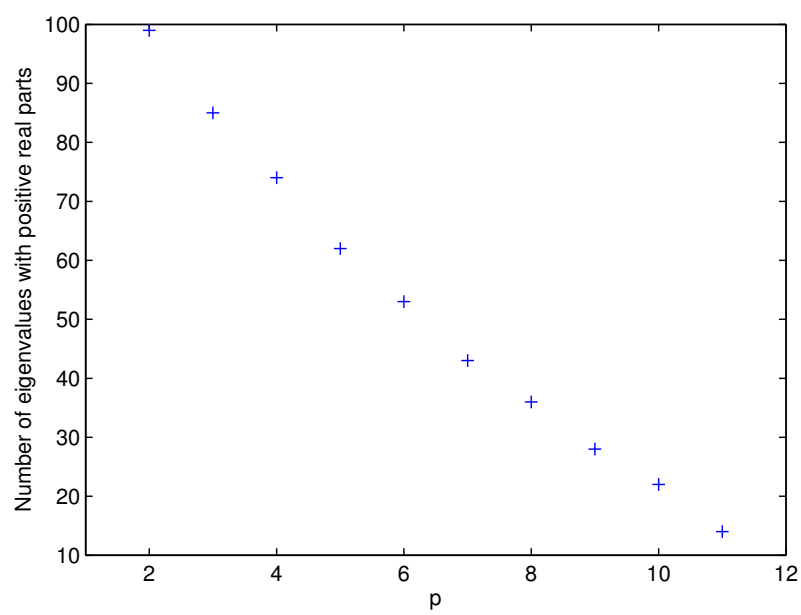

Figure 3.1: The number of eigenvalues with positive real parts versus p.

In all our computations we obtained some eigenvalues with real parts greater than zero. As seen from Fig. 3.1, as p increased the number of eigenvalues with positive real parts decreased. This was because an increase in p constitutes a smoother function $\beta(x)$, hence derivatives are approximated better. However, this does not explain the fact that eigenvalues with positive real parts exist. We will later on address this issue. Motivated by theorem 1.2.2 and work done by Embree [4], we investigated the behavior of eigenvalues for $p = 2$, $p = 3$, and $p = 4$. The eigenvalues shown in Fig. 3.2 for $p = 2$

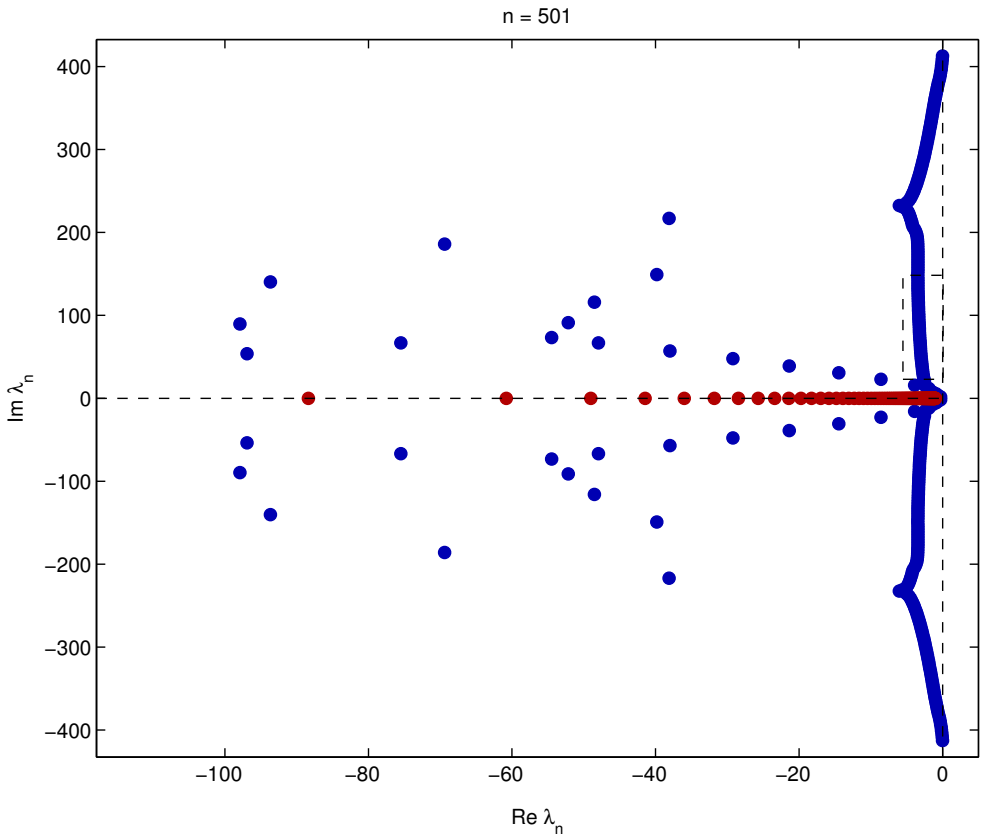

Figure 3.2: $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 2$.

were computed using 402 nodes for which 600 eigenvalues were computed. The number (n) of eigenvalues that are shown in Fig. 3.2 is 501. At low frequencies the real and imaginary parts of the complex eigenvalues varied linearly. The real parts decayed less rapidly as the frequency increases. We believe that eigenvalues at high frequencies close

to the imaginary axis are poor approximations because their real parts are heading to 0 instead of $-\infty$. Motivated by work done by Embree [4], we considered the eigenvalues in the rectangular region to determine any relationship between the real and imaginary parts. From Fig. 3.3 it is seen that eigenvalues ($\lambda = \omega + i\mu$) in the rectangular region

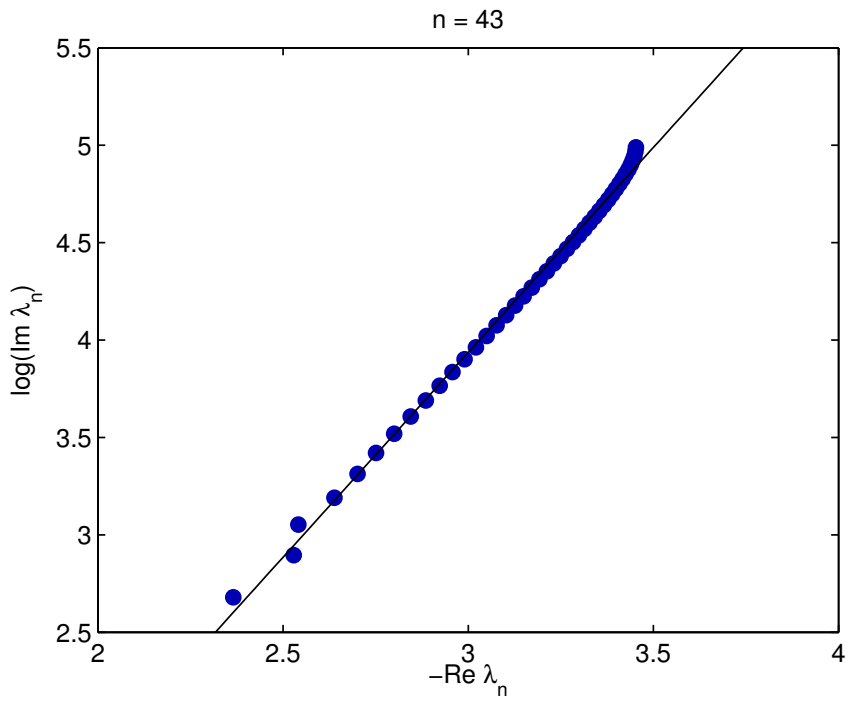

Figure 3.3: $\log(\mathrm{Im}\lambda_n)$ versus $-\mathrm{Re}\lambda_n$ for $p = 2$.

of Fig. 3.2 obey

$$\mu \approx \exp(\theta\omega + \gamma). \tag{3.6}$$

The parameters θ and γ were obtained using a least-squares fit as:

$$\theta \approx -2.1042 \qquad \gamma \approx -2.3757.$$

The curve $\mu(\omega) = \exp(\theta\omega + \gamma)$ is plotted in Fig. 3.4 and it approximates eigenvalues at low frequencies very well. We will address the issue of poor approximations at higher frequencies later on.

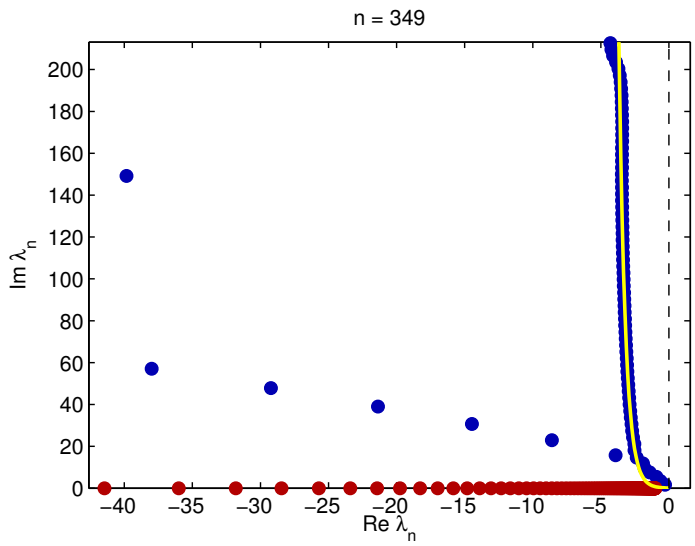

Figure 3.4: Restricted $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 2$.

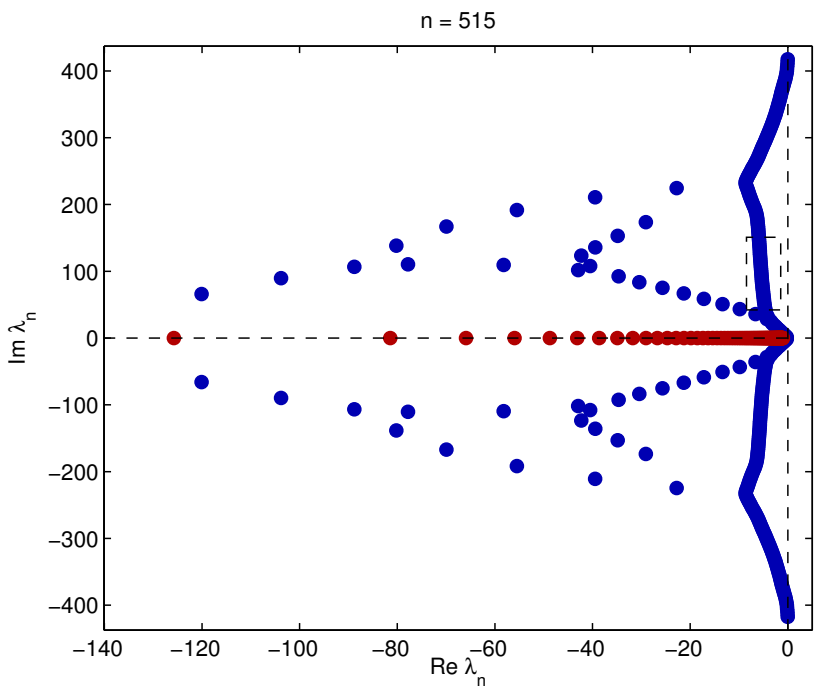

Figure 3.5: $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 3$.

From Fig. 3.5 when $p = 3$, we see that the eigenvalues follow a more structured pattern mainly because of the smoothness of $\beta(x)$. Restricting our attention to eigenvalues in the rectangular region we see that (3.6) holds where

$$\theta \approx -0.9467 \qquad \gamma \approx -0.5786.$$

The curve $\mu(\omega) = \exp(\theta\omega + \gamma)$ is plotted in Fig. 3.6. From Fig. 3.7 when $p = 4$, we see

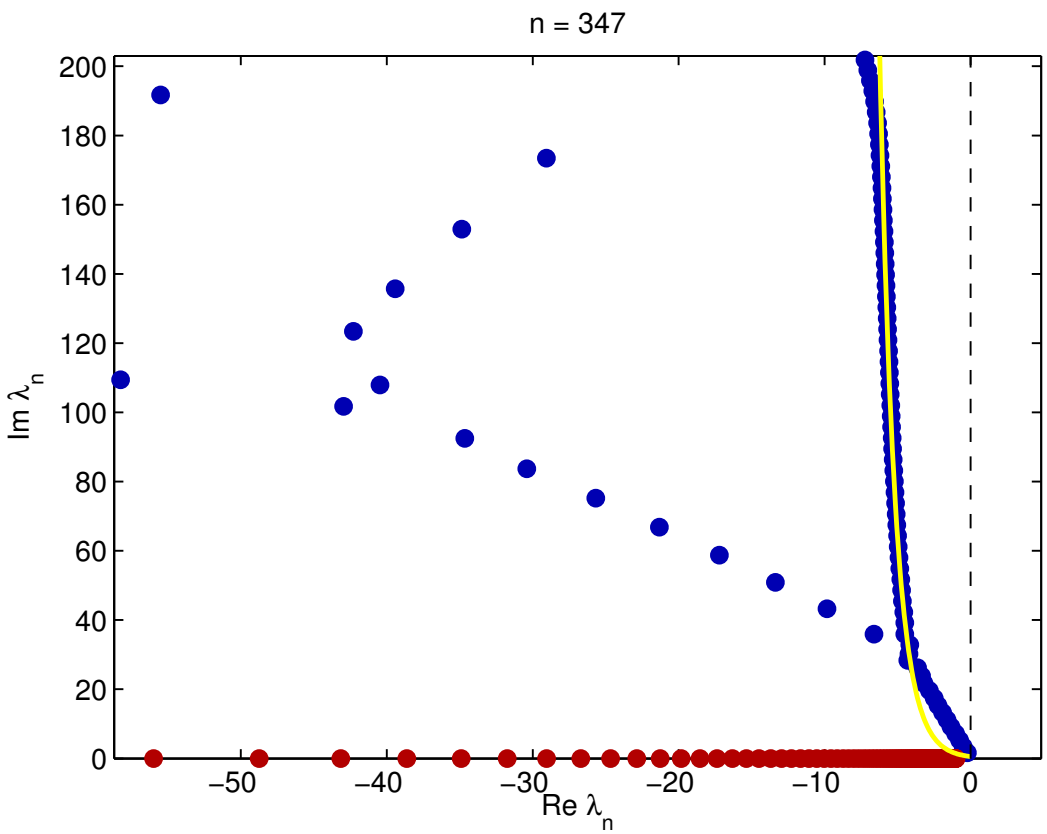

Figure 3.6: Restricted $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 3$.

that the eigenvalues follow a more structured pattern than those computed for $p = 3$. Again restricting our attention to eigenvalues in the rectangular region we see that (3.6) also holds where

$$\theta \approx -0.6769 \qquad \gamma \approx -0.4001.$$

The curve $\mu(\omega) = \exp(\theta\omega + \gamma)$ is shown in Fig. 3.8.

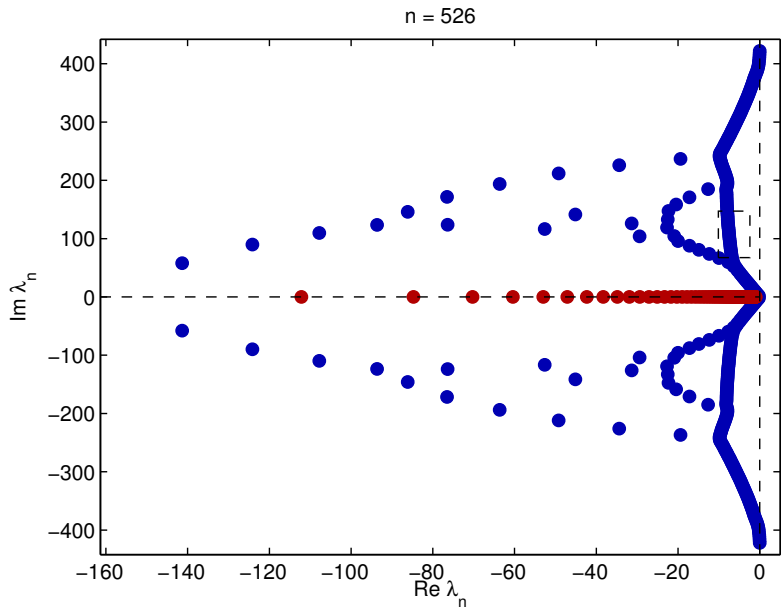

Figure 3.7: $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 4$.

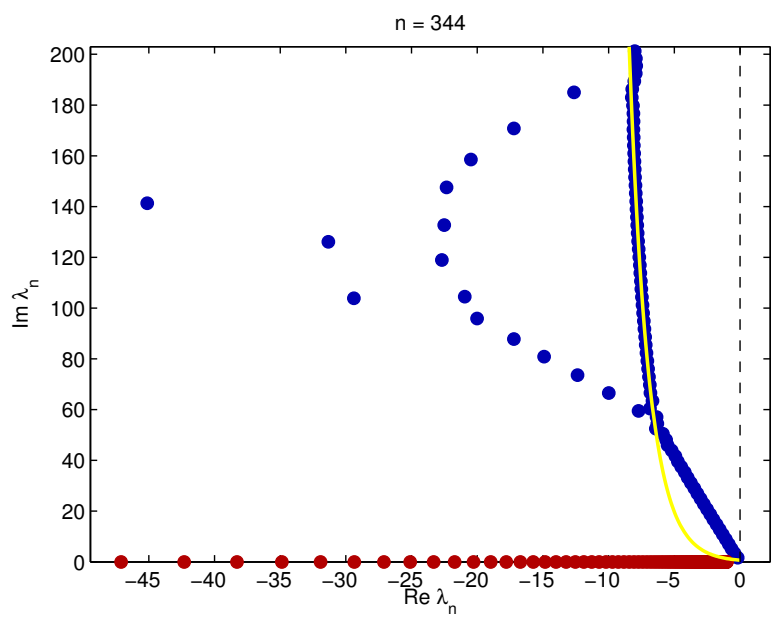

Figure 3.8: Restricted $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 4$.

3.2 Results for method II

For these results, we also computed 600 eigenvalues. Fig. 3.9 shows that the number

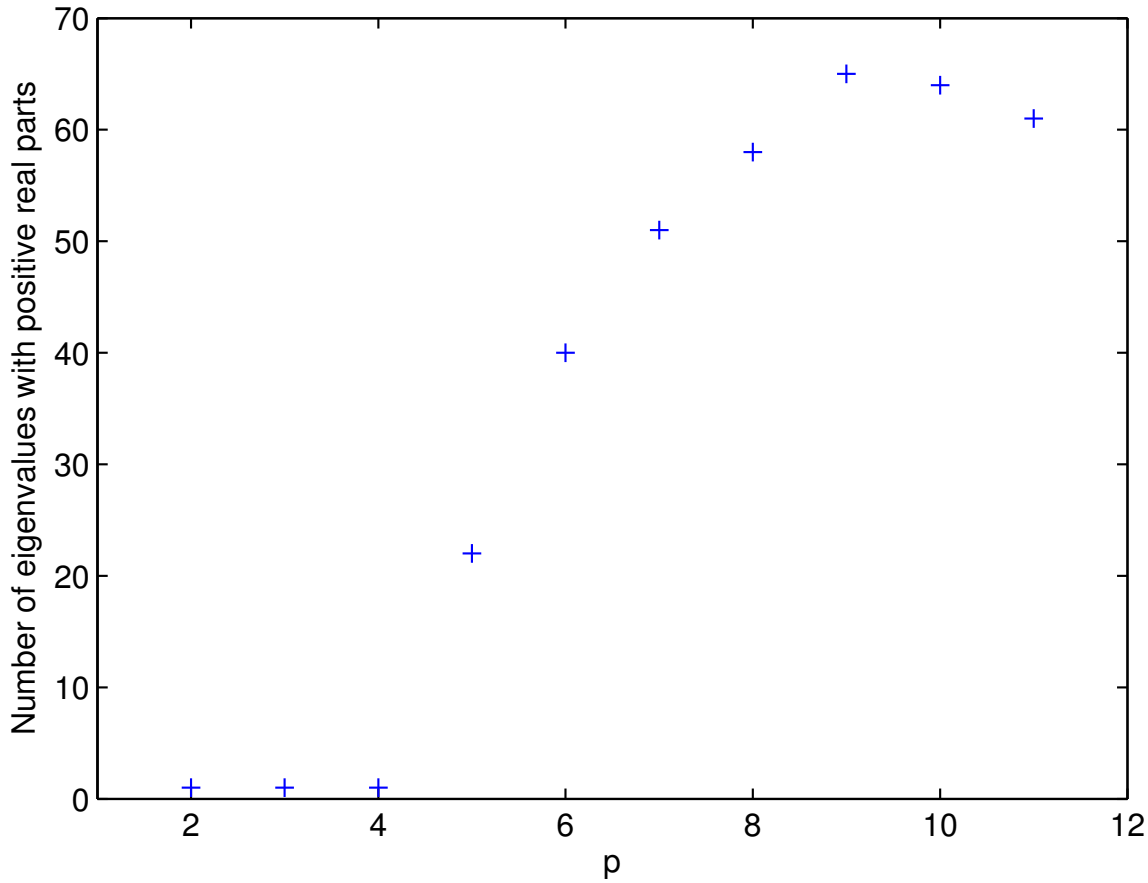

Figure 3.9: The number of eigenvalues with positive real parts versus p for method II.

of eigenvalues with positive real part increase with increasing p ($2 \leq p \leq 9$) and then decrease for $p > 9$. We suspect that the reason for this was poor approximations in the quadrature used in computing the coefficients d_k.

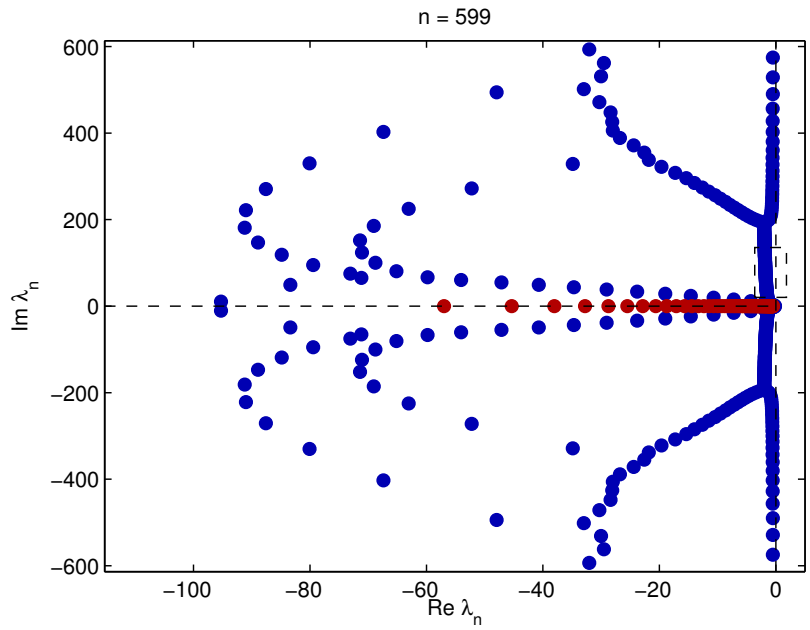

Figure 3.10: $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 2$ in method II.

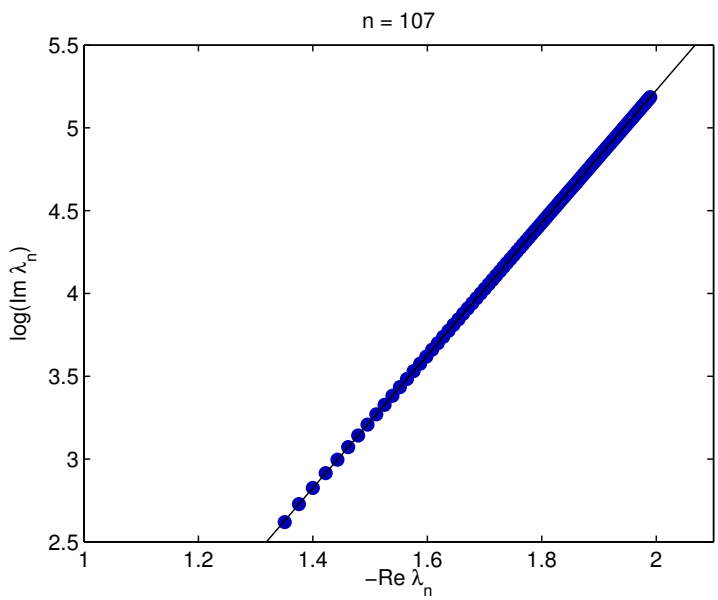

Figure 3.11: $\log(\mathrm{Im}\lambda_n)$ versus $-\mathrm{Re}\lambda_n$ for $p = 2$ in method II.

From Fig. 3.10, the eigenvalues followed a more structured pattern than those computed for $p = 2$ in method I. There were also fewer eigenvalues with real parts approaching zero which suggests that eigenvalues computed with method II are overall more accurate than those computed from method I. Considering eigenvalues in the rectangle region of Fig. 3.10, we obtained that the eigenvalues obeyed (3.6) as shown in Fig. 3.11. The parameters θ and γ were obtained from a least-squares fit as:

$$\theta \approx -4.0052 \qquad \gamma \approx -2.7824.$$

The curve $\mu(\omega) = \exp(\theta\omega + \gamma)$ is plotted in Fig. 3.12.

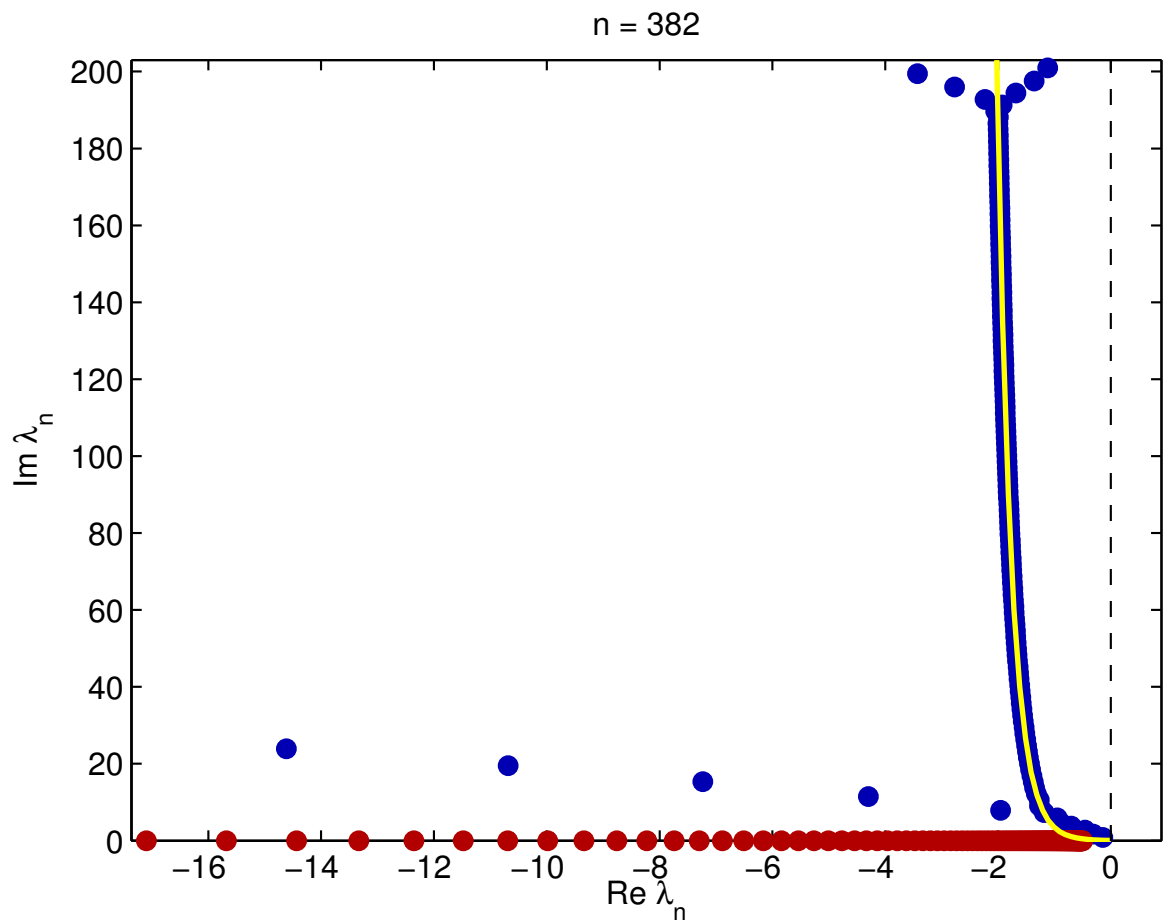

Figure 3.12: Restricted $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 2$ in method II.

For $p = 3$, $\theta \approx -1.9986$ and $\gamma \approx -1.5275$.

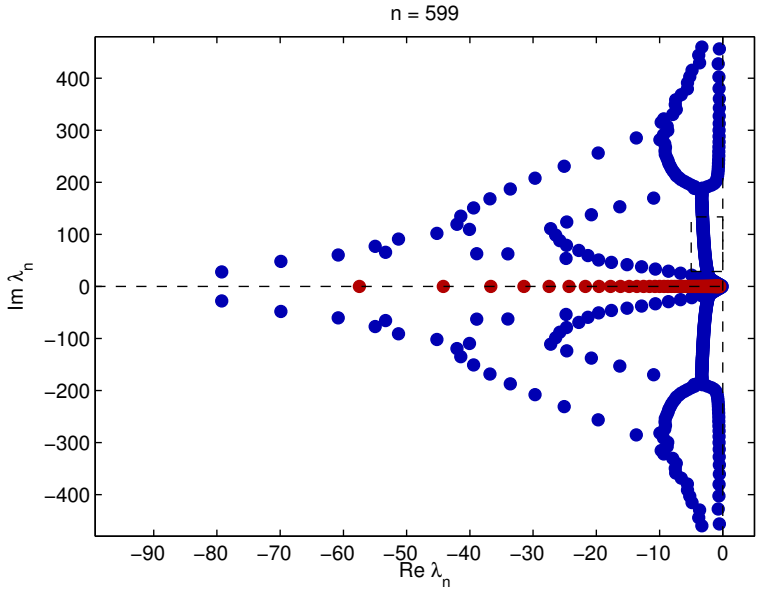

Figure 3.13: $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 3$ in method II.

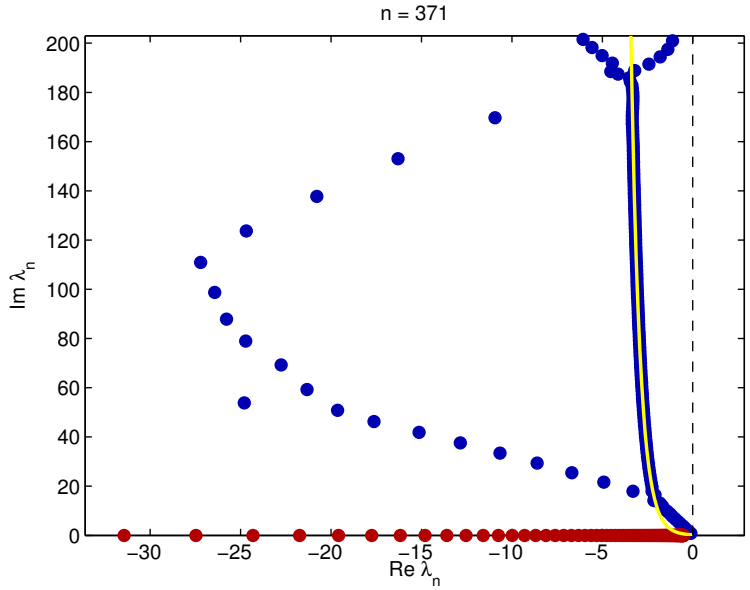

Figure 3.14: Restricted $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 3$ in method II.

For $p = 4$, $\theta \approx -1.3497$ and $\gamma \approx -1.0821$.

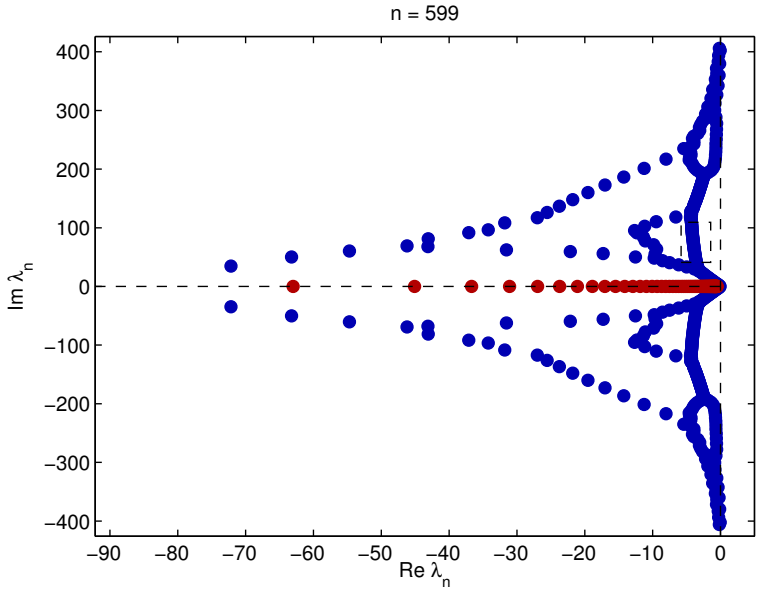

Figure 3.15: $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 4$ in method II.

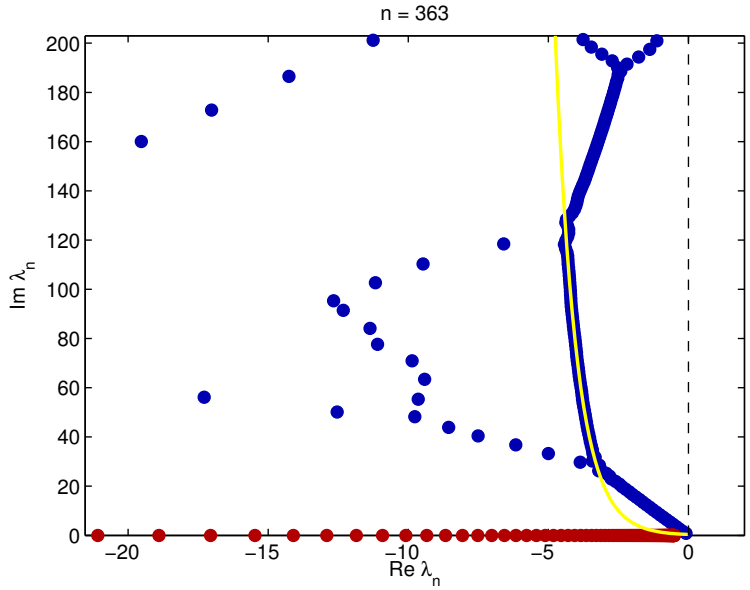

Figure 3.16: Restricted $\mathrm{Im}\lambda_n$ versus $\mathrm{Re}\lambda_n$ for $p = 4$ in method II.

The main reason we obtained eigenvalues with positive real parts and poor eigenvalue approximations was that the matrices were generally ill-conditioned. By ill-conditioned we mean that, small perturbations in some of the matrix elements resulted in large differences in some eigenvalues computed. We investigated sensitivity of the eigenvalues by computing the condition number of each eigenvalue using MATLAB's `condeig` function. In all cases ($p = 2$, $p = 3$, and $p = 4$) for the results of both methods presented previously, the minimum condition number has order of magnitude 1. For $p = 2$, $p = 3$, and $p = 4$ for method I, the order of magnitude of the maximum condition number was 10^7, 10^{10}, and 10^{10} respectively. Likewise for method II, the order of magnitude for the maximum condition number was 10^{10}, 10^{11}, and 10^{11} respectively.

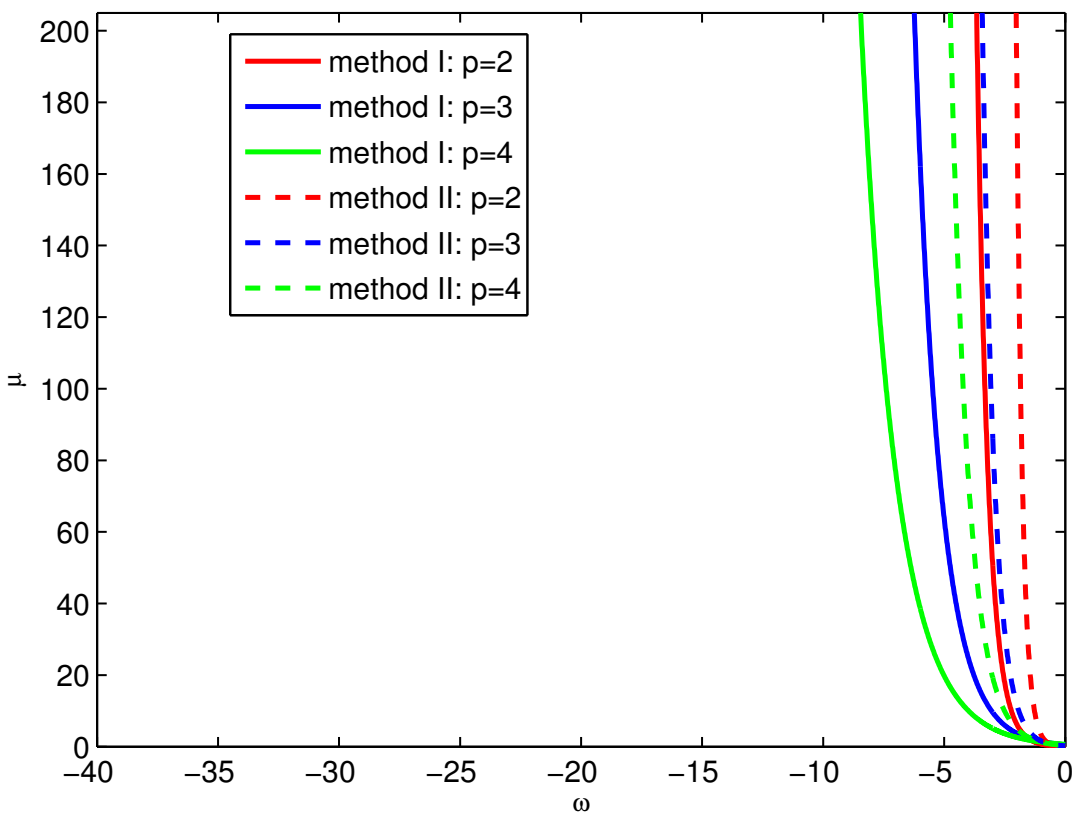

Figure 3.17: Comparison of μ versus ω for $p = 2$, $p = 3$, and $p = 4$ of method I and method II.

From Fig. 3.17, it is seen that there is some scaling relationship between the graphs of μ versus ω for $p = 2$ of both methods. A similar relationship also exists between the graphs of μ versus ω for $p = 3$ and $p = 4$ of both methods. It turns out that

$$\theta \approx \frac{\theta'}{2}$$

and

$$\gamma \approx \log 2 + \gamma'$$

where θ and θ' are parameters in $\mu(\omega)$ for method I and method II respectively. γ and γ' are also parameters in $\mu(\omega)$ for method I and method II respectively. These results confirm that on the curves in Fig. 3.17, eigenvalues computed from method I are twice as large in magnitude as the eigenvalues computed from method II. Motivation for these results is drawn from the fact that we analytically verified that the eigenvalues of (2.21) – (2.25) with $a = 0$ were half those of (1.1) with $\beta = 0$.

Chapter 4

Summary

The discrete eigenvalue problem associated with (1.1) was formulated on a single and double spatial domain. Both discrete eigenvalue problems were solved using the Chebyshev-Tau method. Eigenvalues were computed using

$$\beta(x) = \begin{cases} 0, & x \in [-1, 0], \\ x^p, & x \in (0, 1], \end{cases} \qquad p = 2, 3, 4$$

and

$$\alpha(\eta) = 2\left(\frac{\eta + 1}{2}\right)^p \qquad p = 2, 3, 4.$$

More that half the eigenvalues computed from both methods had negative real parts. At some low frequencies, the real part (ω) and imaginary part (μ) were related by $\mu = \exp(\theta\omega + \gamma)$. θ and γ were computed using a least-squares fit. A summary of θ and γ is given in Table 4.1 and Table 4.2 . Poor eigenvalue approximations and eigenvalues

p	θ	γ
2	-2.1042	-2.3757
3	-0.9467	-0.5786
4	-0.6769	-0.4001

Table 4.1: θ and γ for method I.

with positive real parts were caused by ill-conditioned matrices.

p	θ	γ
2	-4.0052	-2.7824
3	-1.9986	-1.5275
4	-1.3497	-1.0821

Table 4.2: θ and γ for method II.

References

[1] Shuping Chen, Kangsheng Liu, and Zhuangyi Liu. Spectrum and stability for elastic systems with global or local Kelvin-Voigt damping. *SIAM J. APPL. MATH.*, 59(2):651–668, 1998.

[2] Qiong Zhang. Exponential stability of an elastic string with Kelvin-Voigt damping. *Z. Angew. Math. Phys.*, 61:1009–1015, March 2010.

[3] Michael Renardy. On localized Kelvin-Voigt damping. *ZAMM Z. Angew. Math. Mech.*, 84(4):280–283, February 2004.

[4] Mark Embree. Eigenvalues and resolvent norm estimates for Kelvin-Voigt damping. (Unpublished), November 2008.

[5] C.Canuto, M.Y.Hussaini, A.Quarteroni, and T.A.Zang. *Spectral Methods Fundamentals in Single Domains*. Springer, 2010.

[6] David Gottlieb and Steven A.Orszag. *Numerical Analysis of Spectral Methods: Theory and Applications*. SIAM, 1977.

Appendix A

Properties of Chebyshev polynomials and expansions

Chebyshev polynomials of the first kind are polynomial functions obtained from a singular Stürm-Liouville problem. They are given by

$$T_k(x) = \cos[k \arccos(x)]. \tag{A.1}$$

If T_k is normalized so that $T_k(1) = 1$ then,

$$T_k(x) = \cos k\theta \quad \text{and} \quad x = \cos\theta.$$

We can show from a trigonometric identity substitution and by induction that the following recurrence relation is obtained

$$T_{k+1} = 2xT_k - T_{k-1},$$

where $T_0(x) \equiv 1$ and $T_1(x) \equiv x$.
Chebyshev polynomials are orthogonal in the inner product with respect to the weight $(1-x^2)^{-1/2}$ i.e.

$$\langle T_k, T_\ell \rangle = \int_{-1}^{1} T_k(x) T_\ell(x)(1-x^2)^{-1/2} \mathrm{d}x = \begin{cases} 0, & k \neq \ell; \\ \pi c_k/2, & \text{otherwise.} \end{cases} \tag{A.2}$$

$$c_k = \begin{cases} 2, & \text{if} \quad k = 0; \\ 1, & \text{if} \quad k = 1, 2, 3 \ldots \end{cases} \tag{A.3}$$

Using trigonometric identities, (A.1) and some manipulations we can show that

$$T_k(\pm 1) = (\pm 1)^k, \tag{A.4}$$

$$T_k(x)T_\ell(x) = \frac{T_{k+\ell}(x) + T_{|k-\ell|}(x)}{2}, \tag{A.5}$$

and

$$T_k'(\pm 1) = (\pm 1)^{k+1}k^2. \tag{A.6}$$

Furthermore, if a known function $f(x)$ is expanded with a series of Chebyshev polynomials as $\sum_{k=0}^{\infty} \hat{f}_k T_k(x)$ then,

$$f' = \sum_{k=0}^{\infty} \hat{f}_k^{(1)} T_k(x), \tag{A.7}$$

where

$$\hat{f}_k^{(1)} = \frac{2}{c_k} \sum_{\substack{p=k+1 \\ p+k \text{ odd}}}^{\infty} p\hat{f}_p \qquad k \geq 0. \tag{A.8}$$

In a similar way

$$f'' = \sum_{k=0}^{\infty} \hat{f}_k^{(2)} T_k(x), \tag{A.9}$$

where

$$\hat{f}_k^{(2)} = \frac{1}{c_k} \sum_{\substack{p=k+2 \\ p+k \text{ even}}}^{\infty} p(p^2 - k^2)\hat{f}_p \qquad k \geq 0. \tag{A.10}$$

The truncated form of (A.8) in matrix-vector notation is given as:

$$\hat{\mathbf{f}}^{(1)} = \mathbf{M}\hat{\mathbf{f}}, \tag{A.11}$$

where

$$\mathbf{M} = \frac{2}{c_k}\begin{bmatrix} 0 & 1 & 0 & 3 & 0 & 5 & 0 & \cdots & 0 & N \\ 0 & 0 & 2 & 0 & 4 & 0 & 6 & \cdots & & 0 \\ 0 & 0 & 0 & 3 & 0 & 5 & 0 & 7 & \cdots & N \\ \vdots & 0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ & \vdots & 0 & \ddots & & & & & & N \\ & & \vdots & \ddots & & & & & & \vdots \\ & & & & & & & & & \\ & & & & & & & & & \\ 0 & 0 & 0 & \cdots & & & & & & 0 \end{bmatrix} \tag{A.12}$$

Similarly, the truncated form of (A.10) in matrix-vector notation is given as:

$$\hat{\mathbf{f}}^{(2)} = \mathbf{M}^2\hat{\mathbf{f}}, \tag{A.13}$$

In (A.8) and (A.10), c_k is given by (A.3). The results from (A.8) and (A.10) are obtained from (A.1), a trigonometric identity, and some manipulations. They can also be generalized to the kth derivative.

Appendix B

Sample MATLAB code

B.1 Method I – $p = 2$

```
clear all
close all

nn = 402; % This has to be even

x = cos(pi.*(0:nn-1)./(nn-1)); % Chebyshev-Gauss-Lobatto nodes
beta = x.^2; beta(nn/2+1:nn) = 0;

b = cbt(beta);

% exact coefficients for x^2
% b = [(pi/4)*(2/(pi*2)),(2/3)*(2/pi),(pi/8)*(2/pi), ...
%     2*(-2*sin((3:nn-1)*pi/2)./(-4*(3:nn-1)+(3:nn-1).^3))./pi];

% preallocated matrices
M = zeros(nn,nn);
Q1 = [ones(nn-2,1) ((-1).^(0:nn-3))'];
Q2inv = [ones(2,1)./2 ((-1).^(0:1))'./2];
S = [eye(nn-2,nn-2); -(Q1*Q2inv)'];

% constructing B1
```

```
B1 = tril(toeplitz(b(1:end)));
B1=.5*B1;
B1(2:end,1)=2*B1(2:end,1);
B1(1:nn+1:nn*nn)=2*B1(1:nn+1:nn*nn);

% constructing B2
B2 = triu(toeplitz(b(1:end)),1);
B2=.5*B2;

% constructing B3
B3 = zeros(nn,nn);
B3(2:end-1,2:end-1) = fliplr(triu(toeplitz(flipud(b(3:end)))));
B3 = .5*B3;

% constructing M
k = 1:nn+1:nn*nn;
for i=nn-1:-2:1
    M(k(1:i)+nn-i)=nn-i:nn-1;
end
M = M';
M(2:end,:) = 2*M(2:end,:);

M2=M*M;

MBM=M*(B1*M+B2*M+B3*M);

N = [zeros(nn-2,nn-2), eye(nn-2,nn-2);...
    M2(1:nn-2,:)*S, MBM(1:nn-2,:)*S];

d = eigs(N,3*length(N)/4,'sm');
e = d(real(d)<0 & imag(d)>0);
f = d(real(d)<0 & imag(d)<0);
g = d(real(d)<0 & imag(d)==0);

figure(1)
plot(real(e),imag(e),'o','markersize',7,'markeredgecolor',[0,0,.7],...
    'markerfacecolor',[0,0,.7])
hold on
plot(real(f),imag(f),'o','markersize',7,'markeredgecolor',[0,0,.7],...
```

```
    'markerfacecolor',[0,0,.7])
plot(real(g),imag(g),'o','markersize',7,'markeredgecolor',[.7,0,0],...
    'markerfacecolor',[.7,0,0])
plot([min(real(e))-20 0],[0 0],'k--')
plot([0 0],[-1000 1000],'k--')
axis([min(real(e))-20,5,-max(imag(e)+20),max(imag(e)+20)])
xlabel('Re \lambda_n')
ylabel('Im \lambda_n')
title(['n = ',num2str(length(e)+length(f)+length(g))])

% linear polynomial fit of te
te=e((imag(e)<150)&(imag(e)>14)&(real(e)>-3.5));
n=length(te);
p = polyfit(-real(te),log(imag(te)),1);
lmu = p(1)*linspace(2,4,100)+p(2);

figure(2)
plot(-real(te),log(imag(te)),'o','markersize',7,'markeredgecolor',...
    [0,0,.7],'markerfacecolor',[0,0,.7])
hold on
plot(linspace(2,4,100),lmu,'k-')
xlabel('-Re \lambda_n')
ylabel('log(Im \lambda_n)')
title(['n = ',num2str(n)])
axis([2,4,2.5,5.5])

figure(3)
plot(real(e),imag(e),'o','markersize',7,'markeredgecolor',[0,0,.7],...
    'markerfacecolor',[0,0,.7])
hold on
plot(real(g),imag(g),'o','markersize',7,'markeredgecolor',[.7,0,0],...
    'markerfacecolor',[.7,0,0])
plot(linspace(-10,0),exp(-p(1)*linspace(-10,0)+p(2)),'k-')
plot([0 0],[-1000 1000],'k--')
axis([min(real(e))-20,5,0,max(imag(e)+20)])
xlabel('Re \lambda_n')
ylabel('Im \lambda_n')
title(['n = ',num2str(length(e)+length(g))])
```

B.2 Method II – $p = 2$

```
clear all
close all

nn = 202;

eta = cos(pi.*(0:nn-1)./(nn-1)); % Chebyshev-Gauss-Lobatto nodes
alpha = 2*(((eta+1)/2).^2);

d = cbt(alpha);

% exact coefficients for 2*((eta+1)/2)^2
% d=[3*pi/4,pi/2,pi/8,3*sin(pi*(4:nn))./(4*(4:nn)-5*(4:nn).^3+(4:nn).^5)];
% d(1)=(1/pi)*d(1);
% d(2:end)=(2/pi)*d(2:end);

% preallocated matrices
M = zeros(nn,nn);
P = [(-1).^(0:nn-1);ones(1,nn)];
Q = [zeros(1,nn);(-1).^(0:nn-1)];
Qh = [ones(1,nn);((0:nn-1).^2).*(-1).^(1:nn)];
Ph = [zeros(1,nn);((0:nn-1).^2)];

P1 = P(1:2,1:end-2); Q1 = Q(1:2,1:end-2);
Ph1 = Ph(1:2,1:end-2); Qh1 = Qh(1:2,1:end-2);

P2 = P(1:2,end-1:end); Q2 = Q(1:2,end-1:end);
Ph2 = Ph(1:2,end-1:end); Qh2 = Qh(1:2,end-1:end);

Y = [eye(2,2), -P2\Q2; -Qh2\Ph2, eye(2,2)];
Z = [-P2\P1, P2\Q1; Qh2\Ph1, -Qh2\Qh1];

K = Y\Z;

S = [eye(nn-2,nn-2), zeros(nn-2,nn-2); K(1:2,:); ...
     [zeros(nn-2,nn-2), eye(nn-2,nn-2)]; K(3:4,:)];
```

```
% constructing D1
D1 = tril(toeplitz(d(1:end)));
D1=.5*D1;
D1(2:end,1)=2*D1(2:end,1);
D1(1:nn+1:nn*nn)=2*D1(1:nn+1:nn*nn);

% constructing D2
D2 = triu(toeplitz(d(1:end)),1);
D2=.5*D2;

% constructing D3
D3 = zeros(nn,nn);
D3(2:end-1,2:end-1) = fliplr(triu(toeplitz(flipud(d(3:end)))));
D3 = .5*D3;

% constructing M
k = 1:nn+1:nn*nn;
for i=nn-1:-2:1
    M(k(1:i)+nn-i)=nn-i:nn-1;
end
M = M';
M(2:end,:) = 2*M(2:end,:);

Mf=[M,zeros(nn,nn);zeros(nn,nn),M];

M2f=Mf*Mf;

MDM=M*(D1*M+D2*M+D3*M);

X=[zeros(nn-2,nn),zeros(nn-2,nn);
   zeros(nn-2,nn),MDM(1:nn-2,:)];

N = [zeros(2*nn-4,2*nn-4), eye(2*nn-4,2*nn-4); ...
    M2f([1:nn-2,nn+1:2*nn-2],:)*S, X*S];

dd = eigs(N,3*length(N)/4,'sm');
e = dd(real(dd)<0 & imag(dd)>0);
f = dd(real(dd)<0 & imag(dd)<0);
g = dd(real(dd)<0 & imag(dd)==0);
```

```
figure(1)
plot(real(e),imag(e),'o','markersize',7,'markeredgecolor',[0,0,.7],...
    'markerfacecolor',[0,0,.7])
hold on
plot(real(f),imag(f),'o','markersize',7,'markeredgecolor',[0,0,.7],...
    'markerfacecolor',[0,0,.7])
plot(real(g),imag(g),'o','markersize',7,'markeredgecolor',[.7,0,0],...
    'markerfacecolor',[.7,0,0])
plot([min(real(e))-20 0],[0 0],'k--')
plot([0 0],[-1000 1000],'k--')
axis([min(real(e))-20,5,-max(imag(e)+20),max(imag(e)+20)])
xlabel('Re \lambda_n')
ylabel('Im \lambda_n')
title(['n = ',num2str(length(e)+length(f)+length(g))])

% linear polynomial fit of te
te=e((imag(e)<180)&(imag(e)>12)&(real(e)>-2));
nn=length(te);
p = polyfit(-real(te),log(imag(te)),1);
lmu = p(1)*linspace(1,2.1,100)+p(2);

figure(2)
plot(-real(te),log(imag(te)),'o','markersize',7,'markeredgecolor',...
    [0,0,.7],'markerfacecolor',[0,0,.7])
hold on
plot(linspace(1,2.1,100),lmu,'k-')
xlabel('-Re \lambda_n')
ylabel('log(Im \lambda_n)')
title(['n = ',num2str(nn)])
axis([1,2.1,2.5,5.5])

figure(3)
plot(real(e),imag(e),'o','markersize',7,'markeredgecolor',[0,0,.7],...
    'markerfacecolor',[0,0,.7])
hold on
plot(real(g),imag(g),'o','markersize',7,'markeredgecolor',[.7,0,0],...
    'markerfacecolor',[.7,0,0])
plot(linspace(-10,0),exp(-p(1)*linspace(-10,0)+p(2)),'k-')
```

```
plot([0 0],[-1000 1000],'k--')
axis([min(real(e))-20,5,0,max(imag(e)+20)])
xlabel('Re \lambda_n')
ylabel('Im \lambda_n')
title(['n = ',num2str(length(e)+length(g))])
```

B.3 Chebyshev transform function

```
function a = cbt( b )
%CBT Implements the Chebyshev transform using IFFT
%   This function computes Chebyshev coefficients of functions
%   sampled at Chebyshev-Gauss-Lobatto nodes.

b = b(:);

N = length(b) - 1;

b = [b; b(N:-1:2)];

a = ifft(b);

a = [a(1); 2*a(2:N); a(N+1)];

end
```