Eigenvalues of Differential Operators

Solange Mukeshimana

College of Science and Technology

School of Science

Department of Mathematics

Master of Science

Huye 2016

Eigenvalues of Differential Operators

by

Solange Mukeshimana

PG 214003399

A dissertation submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In the College of Science and Technology

Supervisor: Fredrik Berntsson

April 2016

Declaration

I declare that this Dissertation contains my own work except where specifically acknowledged.

Solange Mukeshimana, PG 214003399

Signed

Date

Dedication

To my mother,sisters and my husband this work is dedicated to.

Acknowledgments

I highly appreciate my Supervisor Fredrik Berntsson, for his guidance,motivation and his professional supervision as well as for giving necessary explanation, vivacious and bright ideas in this work.

I am grateful to Dr Isidore MAHARA, Dr. Lydie MPINGANZIMA for their brilliant ideas and their moral encouragement. I would also take this opportunity to thank my beloved parent Cecile KARAHANYUZE, my Sister Celine IRAMPAYE, and my husband Benjamin BWANAKWELI and relatives for their support.

I am also beholden to the prized support of ISP through EAUMP-UR node. I will eternally be thankful to Dr. Froduald MINANI, the coordinator of master program in Applied Mathematics. And to all teaching staff in masters program of Applied Mathematics. I would also like to thank my colleagues of the class, students and Professors that I have been meeting in the summer schools well thought-out by EAUMP for their unforgotten intellectual debate and support.

Finally, I expand my thanks to Angelique DUKUNDE, Emmanuel TWAGIZIMANA, Aphrodice NDUHUYABAGABO, Viateur IRAKARAMA, and John HARERIMANA, your humanity and counsel have been to me a special help, I am so appreciative. I would like to say God bless you all.

Solange Mukeshimana

Abstract

In this thesis, the theory of an eigenvalue of a matrix and differential operator is described and classified. The goal of this work is to investigate how the accurate the eigenvalues and eigenvectors, calculated using finite difference method approximation, are. Numerical examples demonstrate how to achieve this ambition.

Keywords

Eigenvalue Problem, Linear Differential Operator.

Contents

Chapter 1

Introduction

Eigenvalue problems for differential operators have attracted a lot of attention as they have many applications. In this work we are going to deal with the eigenvalue problems for one dimensional equations with given boundary conditions.
In linear algebra an eigenvalue λ and eigenvector x of a matrix A satisfy a relation,

$$Ax = \lambda x.$$

The eigenvalue problem has been studied extensively and good algorithms for computing eigenvalues and eigenvectors exists; see [6, 23, 4]. Also given a computed pair $(\hat{x}, \hat{\lambda})$ we can estimate the error by computing the residual $\|A\hat{x} - \hat{\lambda}\hat{x}\|_2$.

In physics we instead study eigenvalues and eigenfunctions of linear differential operators

$$Lu = \lambda u.$$

Often eigenvalues correspond to the energy in a system; and are thus of physical importance. For instance a gas emits light because the electrons wave functions jump between eigenstates. A single photon with energy corresponding to $\lambda_1 - \lambda_2$ is emitted [15].

In simple cases the eigenvalues of a differential operator L can be calculated analytically; but in most case numerical methods are needed. Often a finite difference approximation of $Lu = \lambda u$ leads to a matrix eigenvalue problem $Ax = \lambda x$.

The goal of this work is to investigate how accurate the eigenvalues and eigenvectors, calculated using finite difference approximation, are. In order

to do so we will create an approximate eigenfunction $\hat{u}$ by, e.g., spline interpolation starting from the calculated eigenvector $\hat{x}$. The residual $\|L\hat{u} - \hat{\lambda}\hat{u}\|_2$ should give an indication about the accuracy of the obtained approximate eigenvalue $\hat{\lambda}$.

The aims of this work are:

- Give a good summary of the theory of linear differential operators; including known results about their spectra. Specifically self-adjoint operators.
- Find a simple example where the calculations can be done analytically.
- Implement a finite difference approximation of the example above and compare the computed eigenvalues with the analytical ones.
- Implement a procedure for obtaining an eigenfunction of L given an eigenvector of the corresponding matrix A. Compute the residual $\|L\hat{u} - \hat{\lambda}\hat{u}\|_2$ and compare with the actual errors.
- Try to generalize the error estimate for matrix-eigenvalues to differential operators. Possibly to self-adjoint operators with only point spectra.

Chapter 2

Literature Review

In this chapter we are going to describe briefly, linear operator , eigenvalue of linear operator, and spectrum of linear operator. We will also give some properties of this spectrum.

2.1 Linear operators

Let us introduce the definition of a linear operator L as a mapping of a vector space U to itself that is to say the *range* $R(L) \subset U$ then an eigenpair (λ, u) satisfies

$$Lu = \lambda u. \tag{2.1.1}$$

The domain $D(L)$ of L is a vector space and the range $R(L)$ lies in a vector space over the same field. Also, linearity means that for all $x, y \in D(L)$ and scalars α, we have [9]:

$$L(x + y) = Lx + Ly, \quad \text{and} \quad L(\alpha x) = \alpha Lx.$$

If $Lu = \lambda u$ or $(L - \lambda I)u = 0$ for some $u \neq 0$, then λ is in the point spectra, that is λ is an eigenvalue of L. The vector u is then called an eigenvector of L. We call (λ, u) an eigenpair of L. This means that the resolvent $r(\lambda) = (L - \lambda I)$ where I is identity either has a non trivial null-space (consisting of the eigenfunction). This is point spectra. Or the resolvent is unbounded meaning null-space is empty (no eigenvector exist).

As an example let us consider differentiation $L = \frac{d}{dx}$ acting on the space $L^2(-\infty, \infty)$. We note that $\frac{d}{dx}\left(e^{i\lambda x}\right) = i\lambda e^{i\lambda x}$. This means that any number

$i\lambda$ is in spectrum. But the supposed eigenfunction $u(x) = e^{i\lambda x}$ is not in the space $L^2(-\infty, \infty)$ since $|u| = 1$ and thus

$$\int_{-\infty}^{\infty} |u(x)|^2 dx = \infty.$$

In finite dimension, one of the most important tools for studying linear operators is certainly the notion of eigenvalues. If $r(\lambda)$ is not one-to-one and does not have an inverse. Then there is $u \neq 0$ satisfying $Lu = \lambda u$. In this case λ is an eigenvalue of L and belong to the point spectra of L, thus we will mainly focus on the accuracy of eigenvalues and eigenfunction as mentioned in our objectives in introduction of this thesis.

Let $X \neq 0$ be complex normed space and $L : D(L) \to X$ a linear operator with domain $D(L) \subset X$. With L we associated the operator

$$L_\lambda = L - \lambda I$$

where λ is a complex number and I is the identity operator on $D(L)$. If L_λ has inverse, we denote it by $R_L(\lambda)$, that is

$$R_L(\lambda) = L_\lambda^{-1} = (L - \lambda I)^{-1}$$

and call it the resolvent operator of L.

A regular point λ of L is a complex number such that

- $R_L(\lambda)$ exist,
- $R_L(\lambda)$ is bounded,
- $R_L(\lambda)$ is defined on a set which is dense in X.

The resolvent set $\rho(L)$ of L is the set of all regular values λ of L. Its complement $\sigma(L) = \mathbb{C} - \rho(L)$ is the complex plane $\mathbb{C}$ is called the spectrum of L, and a $\lambda \in \sigma(L)$ is called spectral value of L [9].

The spectrum $\sigma(L)$ is partitioned into three disjoint sets as follows.

- The point spectrum or discrete spectrum $\sigma(L)$ is the set such that $R_L(\lambda)$ does not exist. A $\lambda \in \sigma(L)$ is called an eigenvalue of L.
- The continuous spectrum $\sigma(L)$ is the set such that $R_L(\lambda)$ exists and is defined on a set which is dense in X but $R_L(\lambda)$ is unbounded.
- The residual spectrum $\sigma(L)$ is the set such that $R_\lambda(L)$ exists and may be bounded or not but does not defined on a set which is dense in X i.e the domain of $R_L(\lambda)$ is not dense in X [9].

2.2 Eigenvalues of matrices

We know that many physical systems occuring in engineering applications can be symbolized as discrete models involving matrices. Some key parameters describing physical systems are closely recounted to eigenvalue of the matrix representing the system. That is why the eigenvalue analysis is everywhere in all branches of modern engineering. Eigenvalue analysis is also used in the design of car stereo systems, where it helps to decrease the vibration of the car due to the music [24].

The eigenvector of a square matrix A is the non zero vector x that, after being multiplied by the matrix, remain parallel to the original vector. For each eigenvector, the corresponding eigenvalues is the factor by which the eigenvector is scaled when multiplied by the matrix. The mathematical formula: Given a square matrix A we can find number λ and vector x ($x \neq 0$) such that

$$Ax = \lambda x. \tag{2.2.2}$$

This problem is called the eigenvalue problem, the number λ is called an eigenvalue of the matrix A and the non-zero vector x is called an eigenvector corresponding to the eigenvalue λ. Note that we are not interested in the trivial solution $x = 0$. Eigenvectors are only unique up to a multiplicative factor, means that if x satisfies (2.2.2) for some λ then so does cx, where c is any constant [23].

If we want to find the eigenvalue, we note that $\lambda x = \lambda I x$ with I an identity matrix. The equation (2.2.2) can be written as

$$Ax - \lambda I x = 0,$$

or

$$(A - \lambda I)x = 0. \tag{2.2.3}$$

Equation (2.2.3) has non trivial solution $x \neq 0$, so the matrix $A - \lambda I$ of this system is singular, which means that

$$\det(A - \lambda I) = 0. \tag{2.2.4}$$

This demonstrates that the following theorem holds:

Theorem 2.2.1 *A number λ is an eigenvalue of a square matrix A if and only if, it satisfies the equation*

$$det(A - \lambda I) = 0. \tag{2.2.5}$$

If (λ, x) is an eigenpair then $(A - \lambda I)x = 0$, $x \neq 0$, and therefore the eigenvalues are the roots of $p_A(\lambda) = det(A - \lambda I) = 0$, since $p_A(\lambda)$ has n roots we may write

$$p_A(\lambda) = (\lambda_1 - \lambda)(\lambda_2 - \lambda) \dots (\lambda_n - \lambda).$$

this question is called the characteristic equation of the matrix A representing the linear operator. For a finite dimensional vector space, eigenvalues of a linear operator are solutions of a the characteristic equation.

Eigenvalue have to be computed numerically in practice. And only approximate eigen pairs can be obtained. The residual $r = A\hat{x} - \hat{\lambda}\hat{x}$, offers a way to judge the accuracy of an approximate eigenvalue $\hat{\lambda}$.

Since eigenvalues are very sensitive to small changes in the coefficients of the characteristic polynomial and the insufficiency of this representation become understandable with the approach of the modern digital computer that why is extremely important to calculate the residual since it should give us an indication about the accuracy of the obtained approximate eigenvalue $\hat{\lambda}$.

Proposition 2.2.2 *Let $A \in \mathbb{C}^{n \times n}$ be non-defective with eigenvector matrix X. Let $(\hat{x}, \hat{\lambda})$ be an approximate eigenpair of A, with $\|\hat{x}\|_2 = 1$ and put the residual $r = A\hat{x} - \hat{x}\hat{\lambda}$. There is an eigenvalue λ of A such that*

$$| \lambda - \hat{\lambda} | \leq \kappa_2(X)\|r\|_2.$$

proof: Let $(\hat{x}, \hat{\lambda})$ be an approximate eigenpair and $r = A\hat{x} - \hat{\lambda}\hat{x}$ then let

$$B = A - r\hat{x}^H,$$

we see that

$$\begin{aligned} B\hat{x} &= (A - r\hat{x}^H)\hat{x} \\ &= A\hat{x} - r\hat{x}^H\hat{x} \\ &= \hat{\lambda}\hat{x}, \end{aligned}$$

so $(\hat{x}, \hat{\lambda})$ is an exact eigenpair of B. Let X be the matrix of eigenvectors of the matrix A so, $X^{-1}AX = D$, and transform

$$\begin{aligned} X^{-1}BX &= X^{-1}AX - X^{-1}r\hat{x}^H X \\ &= D + F \end{aligned}$$

where $D = diag(\lambda_i)$, since $\hat{\lambda}$ is an eigenvalue of $D + F$, as $D + F$ and B are similar[5], there is an eigenvector $u \neq 0$ such that

$$(D + F)u = \hat{\lambda}u$$

This means that

$$Du + Fu = \hat{\lambda}u,$$

so that

$$(D - \hat{\lambda}I)u = -Fu \Longrightarrow u = -(D - \hat{\lambda}I)^{-1}Fu$$

On the other hand

$$\begin{aligned} \|u\|_2 &\leq \frac{1}{\min|\lambda - \hat{\lambda}|}\|F\|_2\|u\|_2 \\ \Longrightarrow \min|\lambda - \hat{\lambda} &\leq \|F\|_2 \\ &= \|X^{-1}r\hat{x}^H X\|_2 \\ &\leq \|X\|_2\|X^{-1}\|_2\|r\|_2\|X\|_2 \\ &= \kappa_2(X)\|r\|_2 \end{aligned}$$

since $\|X\|_2 = 1$ and $\|X\|_2\|X^{-1}\|_2 = \kappa_2(X)$, thus

$$| \lambda - \hat{\lambda} | \leq \kappa_2(X) \parallel r \parallel_2 .$$

A similar result but not exactly the same can be found in [5]. Hence for motivating the proof can you look at [4].

If A is symmetric, then any two eigenvectors from different eigenvalues are orthogonal. This can be seen as follows:

Let u_1 and u_2 be two eigenvectors from two eigenvalues λ_1 and λ_2. Let us show that that $(u_1, u_2) = 0$

$$\begin{aligned} \lambda_1(u_1, u_2) &= (\lambda_1 u_1)^T u_2 && \text{(by definition)} \\ &= (Au_1)^T u_2 && \text{(definition of eigenvector)} \\ &= u_1^T A^T u_2 && \text{(transpose of product)} \\ &= u_1^T (Au_2) && \text{(A is symmetric)} \\ &= u_1^T (\lambda_2 u_2) && \text{(definition of eigenvector)} \\ &= \lambda_2(u_1 u_2) \end{aligned}$$

Thus $(\lambda_1 - \lambda_2)(u_1, u_2) = 0$ but $\lambda_1 - \lambda_2 \neq 0$ since two eigenvalues are different. Consequently $(u_1, u_2) = 0$.

Corollary 2.2.3 *If A is symmetric or hermitian then* $| \lambda - \hat{\lambda} | \leq \|r\|_2$.

Proof: For symmetric or hermitian matrix, the matrix X of eigenvectors is orthogonal or unitary so that $X^{-1} = X^H$ and

$$\|X\| = \|X^H\| = 1.$$

Therefore $\kappa(X) = 1$, and we have

$$| \lambda - \hat{\lambda} | \leq \|r\|_2.$$

Note that if A is $n \times n$ matrix , then (2.2.3) is a polynomial equation (in λ) of degree n and it has n solutions; also Solution to equation (2.2.3) may be repeated in this case we say that the eigenvalue λ has multiplicity $m_\lambda \geq 1$; if A is $n \times n$ matrix; then $m_\lambda \leq n$.

Example 2.2.4 Consider the matrix

$$A = \begin{bmatrix} 3 & 6 & -8 \\ 0 & 0 & 6 \\ 0 & 0 & 2 \end{bmatrix}$$

to find the eigenvalues of A, we must compute the $det(A - \lambda I) = 0$ and set this expression equal to 0, and solve for λ.

$$A - \lambda I = \begin{bmatrix} 3 & 6 & -8 \\ 0 & 0 & 6 \\ 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} = \begin{bmatrix} 3-\lambda & 6 & -8 \\ 0 & -\lambda & 6 \\ 0 & 0 & 2-\lambda \end{bmatrix}$$

we can use the given lemma above to find its determinant

$$det(A - \lambda I) = -\lambda(3 - \lambda)(2 - \lambda)$$

Setting this equal to 0 and solving for λ, we get that $\lambda = 0, 2$, or 3. These are the three eigenvalues of A.

Another result that is sometimes useful is

Theorem 2.2.5 *If $A \in \mathbb{C}^{n \times n}$ is partitional as follows,*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$$

Then $\lambda(A) = \lambda(A_{11}) \cup \lambda(A_{22})$

Proof:

$$Ax = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
$$= \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

where $x_1 \in \mathbb{C}^p$ and $x_2 \in \mathbb{C}^q$, with p number of rows and q number of columns. If $x_2 \neq 0$, then $A_{22}x_2 = \lambda x_2$ and so $\lambda \in \lambda(A_{22})$. If $x_2 = 0$, then $A_{11}x_1 = \lambda x_1$ and so $\lambda \in \lambda(A_{11})$. It follows that $\lambda(A) \subset (\lambda(A_{11}) \cup \lambda(A_{22}))$. Since $\lambda(A)$ and $\lambda(A_{11}) \cup \lambda(A_{22})$ have both the same cardinality, we therefore have that $\lambda(A) = \lambda(A_{11}) \cup \lambda(A_{22})$; see [5].

Corollary 2.2.6 *If A is a triangular matrix then the eigenvalues of A are the diagonal entries .*

Example 2.2.7 Find the eigenvalues of the matrix

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}$$

Use corollary (2.2.6) the eigenvalues of A are $\lambda_1 = 1$ and $\lambda_2 = 2$.

Theorem 2.2.8 *The eigenvalues of a symmetric matrix A (with real entries) are real and the eigenvalues of a non symmetric matrix A are complex conjugate pairs*

Example 2.2.9 Find the eigenvalues of the matrix

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

$$A - \lambda I = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 1-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix}.$$

Let us solve the characteristic equation $det(A - \lambda I) = 0$. This means that $(1-\lambda)(2-\lambda) - 1 = 0$. Thus the eigenvalues of A are $\lambda_1 = \frac{3}{2} + \frac{1}{2}\sqrt{5}$, $\lambda_2 = \frac{3}{2} - \frac{1}{2}\sqrt{5}$.

Example 2.2.10 Find the eigenvalues of the matrix

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

$$A - \lambda I = \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 1-\lambda & 1 \\ -1 & 2-\lambda \end{bmatrix}$$

Let us solve the characteristic equation $det(A - \lambda I) = 0$. This means that $(1-\lambda)(2-\lambda)+1 = 0$. Thus the eigenvalues of A are $\lambda_1 = \frac{3}{2}+\frac{\sqrt{3}}{2}i$, $\lambda_2 = \frac{3}{2}-\frac{\sqrt{3}}{2}i$.

2.3 Linear Differential Operators

The eigenvalue for a boundary value problem for a partial differential equation can be expressed as follows: Consider the two problems

Problem 1 Find λ and $u(x) \neq 0$ such that

$$\begin{cases} -u''(x) = \lambda u, & 0 \leq x \leq l, \\ u(0) = u(l) = 0. \end{cases}$$

Problem 2 Find λ and $u(x) \neq 0$ such that

$$\begin{cases} u''(x) + \lambda u = 0, & 0 \leq x \leq l, \\ u(0) = 0, \\ u(l) + u'(l) = 0. \end{cases}$$

A non-zero solution u only exists for certain values of λ, i.e. the eigenvalues. In order to make the relation between this and the previous eigenvalue problem for matrices or linear operators clear, we need to introduce notation. Note that both the problems involve the same differential operator, i.e. $-d^2/dx^2$, but since the boundary conditions differ we get different eigenvalues and eigenfunctions.

It is very important to introduce differential operator in our work because linear differential equation can be expressed in terms of the differential operator notation. Differential operator is frequently denoted by L that is $\frac{dy}{dx} = Ly$. The symbol L is called a differential operator because it transforms a differentiable function into another function. Moreover, we require this to

be a linear operator: we must have $L(\alpha f(x) + \beta g(x)) = \alpha L f(x) + \beta L g(x)$ for any constants α and β and functions f and g [14].

A general second order linear differential operator in one variable x can be written as

$$Lu = a_2(x)\frac{d^2u}{dx^2} + a_1(x)\frac{du}{dx} + a_0(x)u,$$

where a_0,a_1, and a_2 are the coefficients.

A linear operator has a domain $D(L)$ and a range $R(L)$. For the eigenvalue problem to work out both the domain and the range needs to be subset of the same underlying space. In order to have a norm and a scalar product we use the Hilbert space L^2 of square integrable functions as the basis space. It is also common to use the Sobolev space H^1 [17].

The space $L^2(\Omega)$ where Ω is an interval is a collection of complex valued square integrable functions f on Ω i.e., $\int_\Omega |f(x)|^2 dx < \infty$, with inner product

$$\langle f, g\rangle = \int_\Omega f(x)\overline{g(x)}\, dx$$

and associated norm

$$\|f\| = \sqrt{\int_\Omega |f|^2\, dx}.$$

The domain is the restriction of functions from $L^2(\Omega)$ that have a sufficient number of derivatives:

$$D(L) = \left\{u \in C^2(\Omega) \text{ and } \int_\Omega |u(x)|^2 dx < \infty\right\} = C^2(\Omega) \cap L^2(\Omega)$$

If we take $\Omega = [0, 1]$ and Dirichlet boundary condition are $u(0) = 0$ and $u(1) = 0$, then

$$D(L) = \left\{u \in C^2([0, 1]) \subset L^2(\Omega), \text{ such that } u(0) = u(1) = 0\right\}$$

The range is the large space $R(L) = C^0(\Omega) \subset L^2(\Omega)$.

Returning to Problem 1 we note that

$$C^2([0, 1]) \subset L^2([0, l])$$

so we can define the domain as follows:

$$D(L) = \left\{f \in C^2([0, l]),\ f(0) = f(l) = 0\right\}.$$

Then the operator is

$$L : D(L) \subset L^2([0,l]) \to L^2([0,l]).$$

with

$$Lu = -u''.$$

The differential operator equation can be written as

$$Lu = \lambda u.$$

For Problem 1, observe that since $\sin(\Omega x)'' = -\Omega^2 \sin(\Omega x)$ and $\sin(0) = \sin(n\pi) = 0$, we can construct eigenvalues and eigenfunctions using sines. We have the following result:

Lemma 2.3.1 *Let*

$$\begin{cases} -u''(x) = \lambda u, & 0 \le x \le l, \\ u(0) = u(l) = 0 \end{cases}$$

be an eigenvalue problem with boundary value conditions, and let λ and $u(x) \neq 0$ be an eigenvalue and eigenfunction respectively, for λ_n satisfying the equation $((n\pi)/l)^2 - \lambda = 0$ we have $u(x) = \sin((n\pi)x)/l)$.

Returning to Problem 2 to find the eigenvalues, let us examine the three possibilities $\lambda < 0,\ \lambda = 0,\ \lambda > 0$.

If $\lambda < 0$ the general solution is of the form

$$u(x) = a\cosh(\sqrt{-\lambda}x) + b\sinh(\sqrt{-\lambda}x).$$

Thus,

$$u'(x) = \sqrt{-\lambda}(a\sinh(\sqrt{-\lambda}x) + b\cosh(\sqrt{-\lambda}x)).$$

Using the boundary condition $u(0) = 0$, we get $a = 0$ and on the other hand, the boundary condition $u(l) + u'(l) = 0$ implies that

$$b[\sinh(\sqrt{-\lambda}l) + \sqrt{-\lambda}\cosh(\sqrt{-\lambda}l)] = 0.$$

But since

$$\sinh(\sqrt{-\lambda}l) + \sqrt{-\lambda}\cosh(\sqrt{-\lambda}l) > 0$$

we have that $b = 0$. Hence the problem does not have negative eigenvalues.

If $\lambda = 0$ then the general solution is of the form

$$u(x) = ax + b.$$

The boundary conditions give $u(0) = 0$ when $b = 0$ and $u(l) + u'(l) = 0$ means that $a = 0$, thus $\lambda = 0$ is not an eigenvalue of the problem.

Finally, if $\lambda > 0$ the general solution is

$$u(x) = a\cos(\sqrt{\lambda}x) + b\sin(\sqrt{\lambda}x)$$

and its derivative given by

$$u'(x) = -a\sqrt{\lambda}\sin(\sqrt{\lambda}x) + b\sqrt{\lambda}\cos(\sqrt{\lambda}x).$$

The boundary condition $u(0) = 0$ is possible when $a = 0$. With the boundary condition $u(l) + u'(l) = 0$ we have

$$b\sin(\sqrt{\lambda}l) + b\sqrt{\lambda}\cos(\sqrt{\lambda}l) = 0 \tag{2.3.6}$$

From equation (2.3.6) we can find the value of

$$-\sqrt{\lambda} = \tan(\sqrt{\lambda}l) \tag{2.3.7}$$

This equation implies that the values of the eigenvalues λ are intersections of the graphs of two functions $y = \sqrt{\lambda}$ and $y = -\tan(\sqrt{\lambda}l)$. Note that there are infinitely many intersections. So the eigenfunctions are

$$u(x) = \sin(\sqrt{\lambda_n}x) \tag{2.3.8}$$

for $n = 1, 2, \ldots$.

We can now find the eigenvalues λ_n, $n = 1, 2, \ldots$ from equation (2.3.7). Observe that here we do not include zero eigenvalue since an eigenfunction is just a sine function.

Let $\sqrt{\lambda}l = \mu$, then $\sqrt{\lambda} = \frac{\mu}{l}$. We therefore have from (2.3.7) the following equation:

$$-\frac{\mu}{l} = \tan(\mu).$$

To find the solution of (2.3.7) means to find the intersection of the graphs of function $\tan\mu$ with the graph of $-\frac{\mu}{l}$ we can see that there are infinite number of intersection points. Let $\alpha_n = \sqrt{\lambda_n}$ then α_n satisfy $n\pi - \frac{\pi}{2} < \alpha_n < n\pi$ for all $n \geq 1$ means that our solution belongs in the second and the fourth quadrants of circle.

Lemma 2.3.2 *Let*

$$\begin{cases} u''(x) + \lambda u = 0, & 0 \le x \le l, \\ u(0) = 0, \\ u(l) + u'(l) = 0; \end{cases}$$

be an eigenvalue problem with boundary value conditions, and let λ *and* $u(x) \neq 0$ *be an eigenvalue and eigenfunction respectively. For* λ_n *satisfying the equation* $\tan(\sqrt{\lambda_n} l) + \sqrt{\lambda_n} = 0$ *we have* $u(x) = \sin(\sqrt{\lambda_n} x)$.

Since in our work we deal with operator L which is symmetric, or self adjoint [9], then the eigenvalues are real and eigenfunctions corresponding to different eigenvalues are orthogonal with respect to each other. Let us show orthogonality of our eigenfunctions with $n \neq m$,

$$\begin{aligned} \int_0^1 \sin \alpha_n x \sin \alpha_m x dx &= \frac{1}{2} \int_0^1 \cos(\alpha_n - \alpha_m) x dx - \frac{1}{2} \int_0^1 \cos(\alpha_n + \alpha_m) x dx \\ &= \frac{1}{2(\alpha_n - \alpha_m)} \sin(\alpha_n - \alpha_m) - \frac{1}{2(\alpha_n + \alpha_m)} \sin(\alpha_n + \alpha_m) \end{aligned}$$

Using the fact that $\sin \alpha_n = -\alpha_n \cos \alpha_n$

$$\begin{aligned} \frac{\sin(\alpha_n - \alpha_m)}{2(\alpha_n - \alpha_m)} &= \frac{1}{2(\alpha_n - \alpha_m)} [\sin \alpha_n \cos \alpha_m - \sin \alpha_m \cos \alpha_n)] \\ &= \frac{1}{2(\alpha_n - \alpha_m)} [-\alpha_n \cos \alpha_n \cos \alpha_m + \alpha_m \cos \alpha_n \cos \alpha_m] \\ &= \frac{1}{2(\alpha_n - \alpha_m)} (-\cos \alpha_n \cos \alpha_m)[\alpha_n - \alpha_m] \\ &= -\frac{1}{2} \cos \alpha_n \cos \alpha_m. \end{aligned}$$

Similarly,

$$\frac{1}{2(\alpha_n + \alpha_m)} \sin(\alpha_n + \alpha_m) = -\frac{1}{2} \cos \alpha_n \cos \alpha_m$$

Therefore,

$$\begin{aligned} \int_0^1 \sin \alpha_n x \sin \alpha_m x dx &= -\frac{1}{2} \cos \alpha_n \cos \alpha_m + \frac{1}{2} \cos \alpha_n \cos \alpha_m \\ &= 0. \end{aligned}$$

To show that zero boundary conditions in the PDE means the space $D(L)$ is linear. Let us verify if the linear combination of two solutions is belong in the space $D(L)$.

If u_1 and u_2 satisfy our problem, then $\forall \alpha, \beta \in \mathbb{R}$, $\alpha u_1 + \beta u_2$ also satisfy our problem, in fact

$$\begin{aligned}(\alpha u_1 + \beta u_2)''(x) + \lambda(\alpha u_1 + \beta u_2)(x) &= \alpha(u_1'' + \lambda u_1) + \beta(u_2'' + \lambda u_2) \\ &= 0, \quad 0 \leq x \leq l\end{aligned}$$

$$(\alpha u_1 + \beta u_2)(0) = \alpha u_1(0) + \beta u_2(0) = 0$$

$$(\alpha u_1 + \beta u_2)'(l) + (\alpha u_1 + \beta u_2)(l) = \alpha(u_1(l) + u_1'(l)) + \beta(u_2(l) + u_2'(l)) = 0$$

Therefore zero boundary conditions in the partial differential equation means the space D(L) is linear.

2.4 Partial Differential Equations

Let $x = (x_1, \ldots, x_n)^T$ be independent variables and $u = u(x)$ be a differentiable function. A partial differential equation (PDE) involves a differential operator that includes derivatives in more than one of the independent variables $x_1, \ldots, x_n$.

As an example, let $\Omega = \{(x, y) \in \mathbb{R}^2 : 0 < x < 1,\ 0 < y < 1\}$ a domain in $\mathbb{R}^2$ and consider the Helmholtz equation with Dirichlet boundary conditions:

$$\begin{cases} u_{xx} + u_{yy} + \lambda^2 u = 0, & (x, y) \in \Omega, \\ u(0, y) = 0, & 0 < y < 1, \\ u(1, y) = 0, & 0 < y < 1, \\ u(x, 0) = 0, & 0 < x < 1, \\ u(x, 1) = 0, & 0 < x < 1. \end{cases}$$

This is an eigenvalue problem where a solution $u \neq 0$ only exists for certain values λ. The differential operator

$$L = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

, involves derivatives in two independent variables. For simplicity we restrict ourselves to functions of one variable. In that case, we deal with ordinary differential equations. Let us mention here that ordinary differential equations are more used for describing the trajectory of a classical particle moving in one dimension, the state of a quantum particle moving in one dimension, ...

Chapter 3

Discretization and Approximate Eigenvalues of Linear Operators

Discretization is the process by which an infinite dimensional problem is replaced by a finite dimensional one. This can be described by a continuous function being replaced by its value at a certain set of discrete points. One can also refer to a differential or an integral equation being approximated by analogous expressions which prescribe values at only a finite number of discrete points. During the discretization process, linear operators are also approximated by matrices. The standard definition of derivative is

$$u'(x) = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h}.$$

Computers cannot deal with the $\lim_{h \to 0}$ and hence a discrete analogue of the continuous case needs to be used in order to approximate the derivatives.

In the following, we introduce the notation that will be used in this section. Let $x_1, x_2, \ldots, x_n$ be equidistant grid points on the interval $[a, b]$ of the real line and $u_i = u(x_i)$ denote the value of the function u at the node x_i of the computational grid. We collect all the function values in a vector

$$U = (u_1, u_2, \ldots, u_n)^T.$$

Thus the vector U is a discrete approximation of the function $u(x)$. The stepsize is

$$h_i = \max_{1 \leq i \leq n-1} |x_{i+1} - x_i|.$$

If the grid points are equally spaced, i.e., all intervals are of equal size; then the stepsize h is constant.

In the next subsection we will describe corresponding discretizations of linear differential operators. Always keeping in mind that by u we mean a function and U we mean the vector representing its discretizations.

3.1 The finite difference method

The finite difference methods are used to approximate derivatives to solve differential equations numerically. They thus allow us to replace the differential equations by finite difference equations. For more clarification, see for example [14, 18, 8].

Let us now describe that technique. We first discretize the domain of the given problem. For this, let $[0, l]$ be an interval and divide it into N equally spaced intervals of size h defined by $h = l/N$. In that case, there are a total number of $N + 1$ grid points labeled x_i for uniform grid $x_i = ih$, $i = 0, 1, \ldots, N$. As mentioned above, the value of the function u at the grid point x_i is denoted by $u_i = u(x_i)$. We now describe the finite differences for derivatives. In our work we focus on the central difference. We thus define the first and second order derivatives of the function u with respect to x as

$$\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1} - u_{i-1}}{2h}$$

and

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2},$$

respectively. We observe that the derivatives are approximated by differences between function values at adjacent points in the grid. This means that the continuous differential operator, acting on the function u, is replaced by a matrix A acting on the vector U. The discrete approximation is close to the continuous one, and use the value of the function evaluated at a point and adjacent point(s) to approximate the derivative of the function at the point. The hypothesis made here is that the functions to be differentiated are well behaved.

In summary, a finite difference solution basically involves three steps:

- Dividing the solution into grids of nodes. Divide the interval $[0, l]$ into N equally spaced intervals of size h, also let $u_i = u(x_i)$. Then $h = l/N$,

there are a total of $N+1$ grid points labeled x_i for uniform grid $x_i = ih$, $i = 0, 1, ..., N$. Since we focus on central difference, the approximation of first-order derivative is

$$(\frac{\partial u}{\partial x})_i \approx \frac{u_{i+1} - u_{i-1}}{2h}$$

and approximation of second-order derivative is

$$(\frac{\partial^2 u}{\partial x^2})_i \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

- Approximating the given differential equation by finite difference equivalence that approximates derivatives by differences between function values at different grid points.
- Solving the difference equations subject to the prescribed boundary and/or initial conditions.

3.2 The finite difference approximation of a differential operator

Let us approximate the second order derivative of the given function by using central difference which has the following formula:

$$u''(x_i) \approx \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2}.$$

In this section, we describe the finite difference discretization of the differential operator problem

$$Lu = \lambda u, \tag{3.2.1}$$

so that we can end up with a matrix problem.

For simplicity, the differential operator L is defined by $\frac{d^2}{dx^2}$. At each interior grid point x_i we have

$$Lu(x_i) = \lambda u(x_i)$$

and for such grid point x_i we use a central difference approximation, i.e.,

$$L(u(x_i)) = u''(x_i) = \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2}. \tag{3.2.2}$$

Using the definition of our differential operator this system can be written as

$$\begin{cases} u''(x_0) &= \lambda u(x_0) \\ u''(x_1) &= \lambda u(x_1) \\ u''(x_2) &= \lambda u(x_2) \\ & \vdots \\ u''(x_n) &= \lambda u(x_n). \end{cases} \tag{3.2.3}$$

The above system is valid for all interior points. Thus we have $n-2$ linear equations. At the boundary points different equations are needed. This will be explained in details when concrete problems are solved. Now we have a system of n equations.

Substituting $u''(x_i)$ into the system (3.2.3) we get a system on n linear combination

$$\begin{aligned} \frac{u(x_1) - 2u(x_0) + u(x_{-1})}{h^2} &= \lambda u(x_0), \\ \frac{u(x_2) - 2u(x_1) + u(x_0)}{h^2} &= \lambda u(x_1), \\ \frac{u(x_3) - 2u(x_2) + u(x_1)}{h^2} &= \lambda u(x_2), \\ & \vdots \\ \frac{u(x_{n+1}) - 2u(x_n) + u(x_{n-1})}{h^2} &= \lambda u(x_n), \end{aligned}$$

from which we can define the vector

$$\mathbf{U} = \begin{pmatrix} u(x_0) \\ u(x_1) \\ \vdots \\ u(x_{n-1}) \\ u(x_n) \end{pmatrix}$$

In this case, the system (3.2.3) may be written as

$$AU = \lambda U \tag{3.2.4}$$

where A is a matrix having the coefficients of each $u(x_i)$ in the system of linear equations and λ is an eigenvalue of A.

We return to our Problem 1 that was discussed in section 2.2 to do the following example we can follow the same procedure as we did for problem 2. We now apply the idea above to the Problems 1 and Problem 2. We start with Problem 1. We recall that this problem consists of finding eigenvalues and eigenfunctions of

$$\begin{cases} -u''(x) = \lambda u, & 0 \leq x \leq l \\ u(0) = u(l) = 0, \end{cases} \tag{3.2.5}$$

As previously seen the eigenvalues and eigenfunctions for (3.2.5) are given by $(n\pi/l)^2$ and $u(x) = \sin((n\pi x)/l)$, respectively. Following now the previous discussions on the finite difference for differential operator, let us introduce a vector

$$\mathbf{U} = \begin{pmatrix} u(x_0) \\ u(x_1) \\ \vdots \\ u(x_{n-1}) \\ u(x_n) \end{pmatrix}.$$

The vector $\mathbf{U}$ contains $n+1$ unknowns so that the matrix A that approximates the differential operator is of size $(n+1) \times (n+1)$. Incorporating the boundary conditions $u_0 = u_n = 0$, we can eliminate unknowns and reduce the dimension of the problem. We are left with the vector that contains unknowns $\mathbf{U} = (u_1, \ldots, u_{n-1})^T$ and an $(n-1) \times (n-1)$ matrix A.

For the numerical computation, after solving the eigenvalue problem for the matrix A, we have to add $u_0 = u_n = 0$ to get the approximated eigenfunctions.

We consider now Problem 2 that consists of determining eigenvalues and eigenfunctions of

$$\begin{cases} -u''(x) = \lambda u, & 0 \leq x \leq l \\ u(0) = 0, \\ u(l) + u'(l) = 0. \end{cases} \tag{3.2.6}$$

As described in Lemma 2.3.2, the eigenvalues are obtained in the intersection of the graphs of the functions $y = \tan(\sqrt{\lambda_n} l)$ and $y = -\sqrt{\lambda_n}$. The eigenfunctions are given by $u(x) = \sin(\sqrt{\lambda_n} x)$. For the finite difference of the

differential operator, we get the vector $\mathbf{U}$ of unknowns defined by

$$\mathbf{U} = \begin{pmatrix} u(x_0) \\ u(x_1) \\ \vdots \\ u(x_{n-1}) \\ u(x_n) \end{pmatrix}.$$

We now discuss how the boundary conditions for this problem are included . The boundary condition $u_0 = 0$ means that the unknown can be eliminated from the equation: The matrix will thus have dimension n–1. For the boundary condition $u(l) + u'(l) = 0$, the last row of the matrix A; matrix whose entries are the coefficients of each $u(x_i)$ in the system of linear equations, should contain the finite difference approximation of $u''(l)$. We give some details following G.D. Smith; see [18]. Let us introduce a ghost point x_{n+1} and write

$$-u''(x_n) \approx \frac{-u(x_{n+1}) + 2u(x_n) - u(x_{n-1})}{h^2}. \tag{3.2.7}$$

The boundary condition $u'(x_n) + u(x_n) = 0$ is discretized as

$$\frac{u(x_{n+1}) - u(x_{n-1})}{2h} + u(x_n) = 0.$$

From this, we have that

$$u(x_{n+1}) = -2hu(x_n) + u(x_{n-1}).$$

Replacing the above expression of $u(x_{n+1})$ in (3.2.7) we get

$$-u''(x_n) \approx \frac{-2u(x_{n-1}) + 2(1+h)u(x_n)}{h^2}.$$

Thus, the coefficients of $u(x_{n-1})$ and $u(x_n)$ are inserted in the last row of the matrix A.

3.3 The approximation of eigenfunctions using cubic splines

A spline function is a function that consists of polynomial parts connected together with certain smoothness condition [16, 7, 11]. More precisely, a cubic spline function, having knots $x_1, x_2, \ldots, x_n$ is a function ϕ such that knots change from x_1 to x_n and

- on each interval $[x_{i-1}, x_i]$, ϕ is a cubic polynomial.
- ϕ has continuous derivative on $[x_0, x_n]$.

We remark from this first condition that on each interval $[x_{i-1}, x_i]$ the spline function $\phi(x)$ is cubic polynomial. We can thus write

$$\phi_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i,\ x_{i-1} \leq x \leq x_i, \quad i = 1, 2, \ldots, n, \tag{3.3.8}$$

where a_i, b_i, c_i, and d_i are the coefficients of the polynomial. This means that it is easy to compute derivatives or integrals of spline functions. From the second condition, we remark that the second derivative is expressed by

$$\phi_i''(x_{i-1}) = \phi_{i-1}'', \quad \phi_i''(x_i) = \phi_i'', \quad i = 1, 2, \ldots, n. \tag{3.3.9}$$

In order to uniquely determine a cubic splines we need to specify function values at the knots, $x_1, \ldots, x_n$. We also need to specify two end point conditions. In our experiments we want to use cubic splines to take eigenvectors of the finite difference equations and extend them into functions that approximate eigenfunctions of the underlying linear differential operator. Thus we have function values u_i at the grid points, and the boundary conditions for the differential operator L gives us natural end point conditions that makes the natural cubic spline unique.

Chapter 4

Numerical results

In this section we present a few numerical experiments that illustrate the results in the previous sections. We also study the accuracy of the finite different approximation for computing eigenvalues and eigenvectors.

4.1 Numerical results for Problem 1

Here we present the numerical results for Problem 1 that is solved in the interval $[0, 1]$. In other words, we consider the following problem

$$\begin{cases} -u''(x) = \lambda u, & 0 \leq x \leq 1, \\ u(0) = u(1) = 0. \end{cases}$$

We first choose a computational grid of size $N = 10$ and compute eigenvalues and eigenvectors of the matrix A as described above in Section 3.2. We then select, for tests, the second eigenvalue λ_2 and the corresponding eigenvector $u_2(x)$. We interpolate the vector $u_2(x)$ in order to get zero boundary data at the points $x = 0$ and $x = 1$; see Figure 4.1 on the left. We also differentiate $u_2(x)$ twice using cubic splines and show that the given differential equation is satisfied, i.e., $-u_2'' \approx \lambda_2 u_2(x)$. The approximated results are displayed in Figure 4.1 on the right. Though not respresented here with supporting figures, we have observed that as we increase the grid size the graphs of $-u_2''$ and $\lambda_2 u_2(x)$ are more closely.

In Figure 4.2 (left), we illustrate the results corresponding to the first 10 analytical (λ_k) and numerical ($\bar{\lambda}_k$) eigenvalues, respectively. In this case, we choose a finite difference discretization of size $N = 50$. We also display

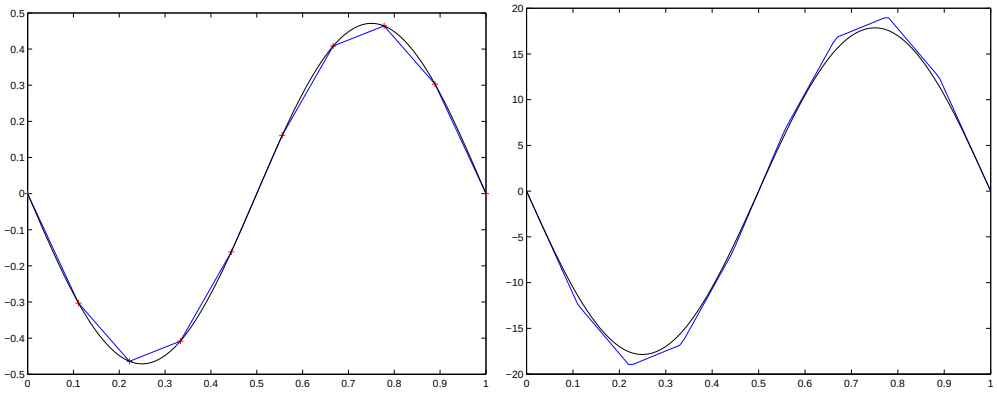

Figure 4.1: On the left, we display the interpolation of the eigenvector u_2. On the right the black line correspond to $\lambda_2 u_2$ while the blue line correspond to $-u_2''$. All the figures correspond to Problem 1.

in Figure 4.2 (right) the semilogy plot of the error between the computed and analytical eigenvalues, i.e., $|\lambda_k - \bar{\lambda}_k|$, $k = 1, 2, \ldots, 10$. The numerical eigenvalues are close to the analytical ones and the error is small.

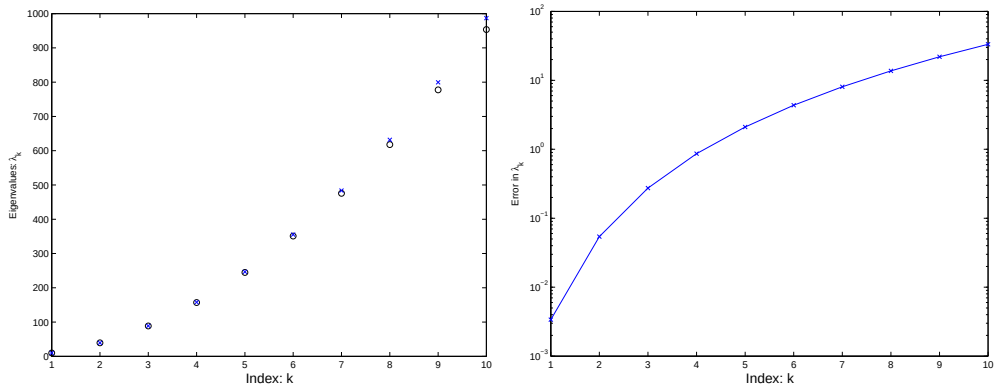

Figure 4.2: The first $n = 10$ eigenvalues λ_k for the Problem 1. We display the exact eigenvalues (left graph, blue x) and the numerical approximation obtained using a finite difference discretization of size $N = 50$ (black o). We also show the semilogy plot of the error $|\lambda_k - \bar{\lambda}_k|$ (right graph).

For results in Figure 4.3, we display the numerical approximations of

eigenfunctions $u_2(x)$ and $u_7(x)$. The computational grid has size $N = 50$. We observe that the frequency of oscillation becomes high or as the eigenvalue increases. This is the expected result since the analytical eigenvectors have the form $u_n(x) = \sin(n\pi x)$, for $n = 1, 2, \ldots$.

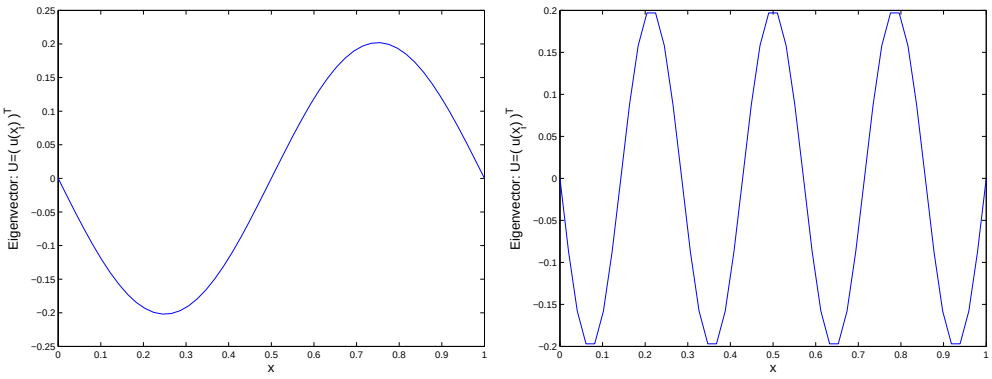

Figure 4.3: We display two eigenvectors $u_2(x)$ on the left and $u_7(x)$ on the right for the Problem 1.

In Figure 4.4, we present the absolute errors $|\lambda_k - \bar{\lambda}_k|$ for different sizes of the computational grid, namely $N = 20$, 60, and 100. We observe that the error is reduced considerably as the grid size increases.

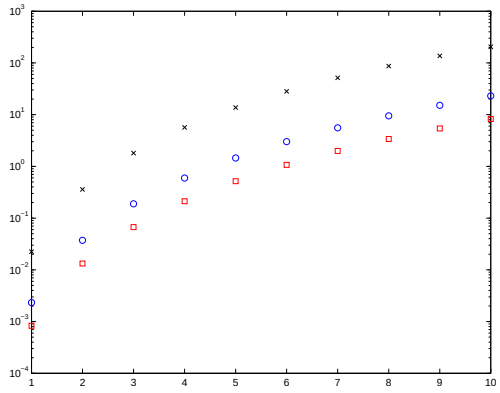

Figure 4.4: We display errors $|\lambda_k - \bar{\lambda}_k|$ for different grid sizes, $N = 20$ (black x), $N = 60$ (blue o) and $N = 100$ (red $\square$).

4.2 Numerical results for Problem 2

Since the interpretations on the numerical results for Problem 2 are the same as for Problem 1, we only present corresponding graphs. Recall that we aim at finding eigenvalues and eigenfunctions in the interval $[0, 1]$. We first choose a computational grid of size $N = 10$ and compute eigenvalues and eigenvectors of the matrix A as described above in Section 3.2. We then pick, for tests, the second eigenvalue λ_2 and the related eigenvector $u_2(x)$. For the cubic spline interpolation we need boundary conditions at $x = 0$ and $x = 1$ and $u(1)+u'(1) = 0$. Here we make sure the condition $u(1)+u'(1) = 0$ is vanished at $x = 1$, where is used as the end point condition for the spline; see Figure 4.5 on the left. We also differentiate $u_2(x)$ twice using cubic splines and show that the given differential equation is satisfied, i.e., $-u_2'' \approx \lambda_2 u_2$. The results are displayed in Figure 4.5 on the right.

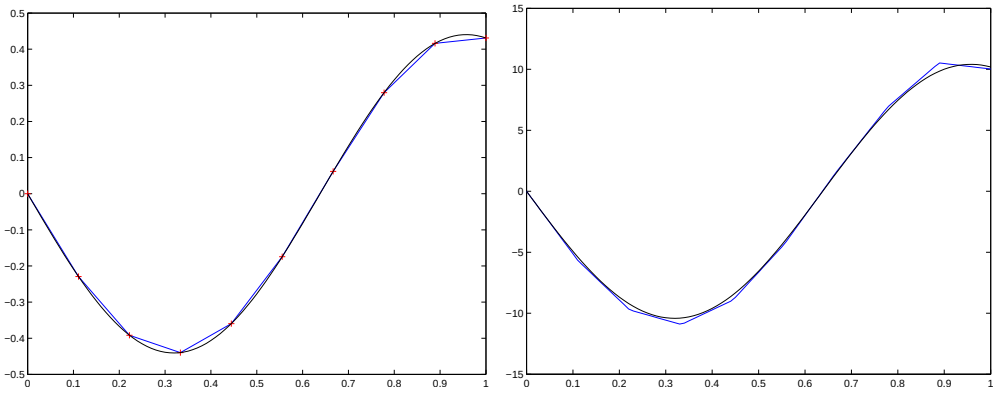

Figure 4.5: On the left, we display the interpolation of the eigenvector u_2. On the right the black line correspond to $\lambda_2 u_2$ while the blue line correspond to $-u_2''$. All the figures correspond to Problem 2. Note that for this problem $u(1) + u'(1) = 0$ where is used as the end point condition for the spline.

For results in Figure 4.7, we display the numerical approximations of eigenfunctions $u_2(x)$ and $u_7(x)$. The computational grid has size $N = 50$. We observe that the frequency of oscillation becomes high or as the eigenvalue increases. This is the expected result since the analytical eigenvectors have the form $u(x) = \sin(\sqrt{\lambda_n}x)$, for $n = 1, 2, \ldots$.

In Figure 4.8, we illustrate the absolute errors $|\lambda_k - \bar{\lambda}_k|$ for different sizes

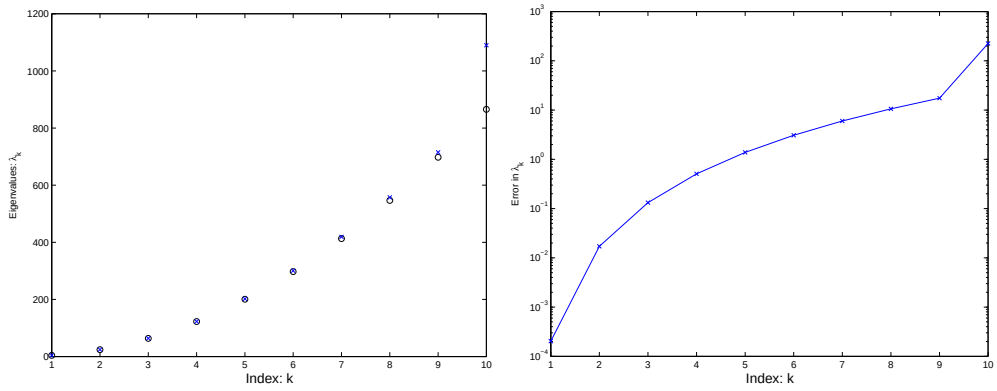

Figure 4.6: The first $n = 10$ eigenvalues λ_k for the Problem 2. We display the exact eigenvalues (left graph, blue x) and the numerical approximation obtained using a finite difference discretization of size $N = 50$ (black o). We also show the semilogy plot of the error $|\lambda_k - \bar{\lambda}_k|$ (right graph).

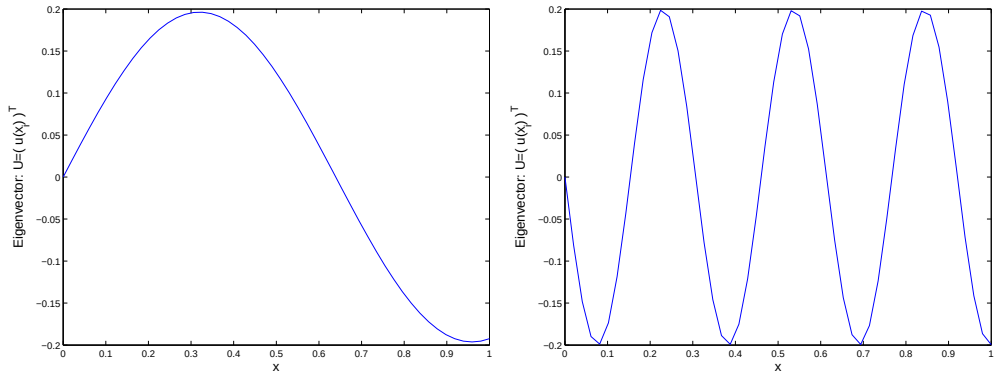

Figure 4.7: We display two eigenvectors $u_2(x)$ on the left and $u_7(x)$ on the right for the Problem 2.

of the computational grid, namely $N = 20$, 60, and 100. We observe that the error is reduced considerably as the grid size increases.

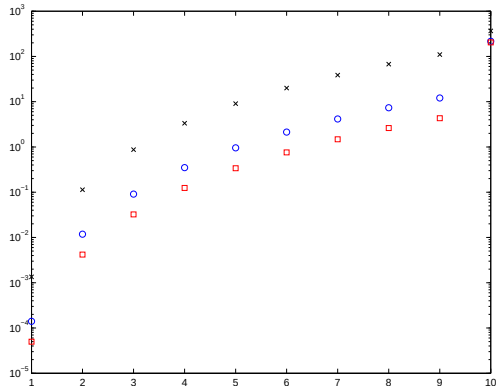

Figure 4.8: We display errors $|\lambda_k - \bar{\lambda}_k|$ for different grid sizes, $N = 20$ (black x), $N = 60$ (blue o) and $N = 100$ (red □).

4.3 Estimation of Error based on the Residual

It is important to compute eigenvalues and eigenfunctions accurately. The eigenvalues of a matrix can be computed with good precision. Direct method, for example the QR algorithm, are only available for relatively small matrix sizes. Thus having approximate eigenvalues and eigenfunctions of a matrix or generally a linear operator we need to be able to judge the accuracy.

Previously we investigated the accuracy for approximate eigenpairs for matrices. The results were given in Proposition 2.2.2 and, for symmetric matrices in Corollary 2.2.3. In this section we will investigate whether the same results can be used also for linear differential operators. Since we investigate only self-adjoint differential operators we will use the following estimate:

$$| \lambda - \hat{\lambda} | \leq \| r \|_2,$$

where the approximate eigenvalue $\bar{\lambda}_k$ is obtained from the finite difference approximation, and the approximate eigenfunction $\hat{u}_k$ is obtained by spline interpolation on the eigenvectors of the discretized problem.

Since the eigenfunctions are obtained using cubic spline interpolation we have in the interval $x_{i-1} < x < x_i$ the eigenfunction given by a cubic poly-

nomial $p_i(x)$. Thus we can compute the residual by integrating

$$I_i = \int_{x_{i-1}}^{x_i} \left(-\frac{d^2}{dx^2} p_i(x) - \bar{\lambda} p_i(x) \right)^2 dx.$$

This can easily be calculated analytically using Matlab functions for working with polynomial expressions. This makes it easier to compute L^2–norms of the approximate eigenfunctions and of the residuals. Even if an eigenvector U_k is normalized (in Euclidean norm) this does not imply that the eigenfunction $u_k(x)$ is normalized in the L^2–norm so we need to normalize it before we calculate the error estimate.

Suppose we want an approximate eigenpairs of a discrete problem of size N_1; obtained by discretizing a differential equation using finite differences. But its too large to compute using direct methods. Instead we pick a smaller dimension $N_2 \ll N_1$ and solve the eigenvalue problem for the smaller discretization. We then get eigenpairs $(\bar{\lambda_k}, \bar{x_k})$ where the eigenvectors are in $\mathbb{R}^{N_2}$. Using the spline interpolation technique we can find the interpolating cubic spline corresponding to $\bar{x_k}$. The spline is then evaluated on the computational grid of size N_1 that correspond to the large eigenvalue problem. This creates the eigenpair $(\bar{\lambda_k}, \widetilde{x}_k)$. Even though we can't solve the eigenvalue for the large size we can still setup the finite difference matrix A(of size $N_1 \times N_1$) and compute the norm of the residual. Thus we can see if we get the expected approximate eigenpair $(\bar{\lambda_k}, \widetilde{x}_k)$.

For the experiment we pick $N_2 = 50$ and $N_1 = 500$ and we consider $N_1 = 500$ which is considered too large for direct eigenvalue methods. We instead chose $N_2 = 50$ and compute the first 10 eigenvalues of the smaller matrix. The eigenvectors are then interpolated onto the larger grid of size $N_1 = 500$ giving us approximate eigenvalues $\lambda_k(N_2)$ and also approximate eigenvectors $u_k^{(N_2,N_1)}$. In Figure 4.9 we show the error $|\lambda_k^{(N_1)} - \lambda_k^{(N_2)}|$ and the corresponding residual based error estimates from Corollary 2.2.3. We observe that Actual errors and the error estimated are very close.

We also display four graphs in order to compare error for first 10 computed eigenvalues and first 10 analytical eigenvalues for problem 1 and problem 2 see Figure 4.10.

For the experiment we choose size of eigenvectors $N_1 = 500$ and computational grid of size $N_2 = 50$ and we compute eigenvalues and eigenvectors of the matrix A. Here we show that the finite difference discretizations lead to two different matrices $A^{(N_1)}$ and $A^{(N_2)}$. The approximate eigenpairs for $A^{(N_1)}$

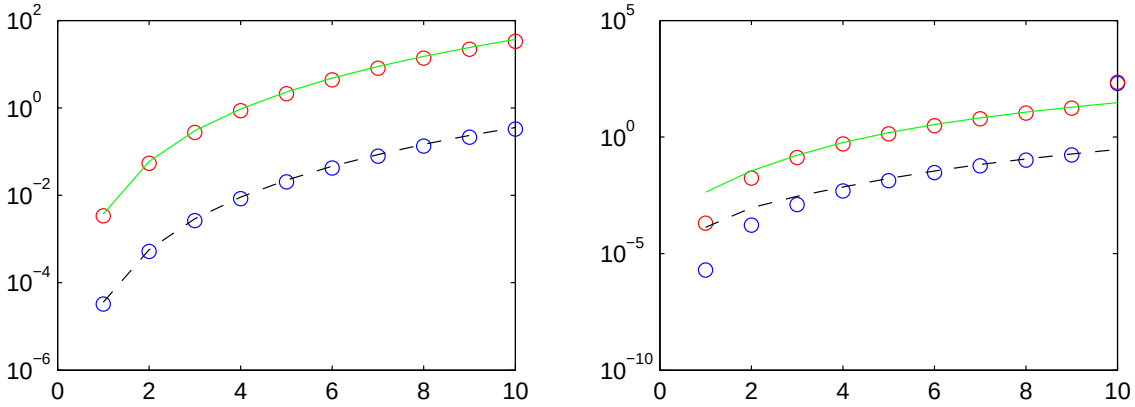

Figure 4.9: We display errors $|\lambda_k(N_1) - \lambda_k(N_2)|$ for different grid sizes, $N_1 = 500$ and $N_2 = 50$. On the left (blue o) and (red o) correspond to actual errors and (black –) and (green line) represent estimated error for Problem 1 and on the right(blue o) and (red o) represent actual errors and (black –) and (green line) correspond to estimated error for Problem 2.

are constructed using eigenpairs of $A^{(N_2)}$. On the left plot we show the approximation for the second eigenvector $u_2^{(N_2,N_1)}$ $\lambda_2^{(N_2)} u_2^{(N_2,N_1)}$ and $A^{(N_1)} u_2^{(N_2,N_1)}$ see Figure 4.11. After we compute residual estimate for N_2 taken as size of eigenvectors and we interpolate to a grid of size N_1.

By computing the norm of the residual and the difference between λ_i for N_1 and N_2 we get 1.7467 and 1.5776 respectively. We conclude that the investigation of the accuracy for approximate eigenpairs for matrices satisfy the conditions given in Proposition 2.2.2 and, for symmetric matrices in Corollary 2.2.3.

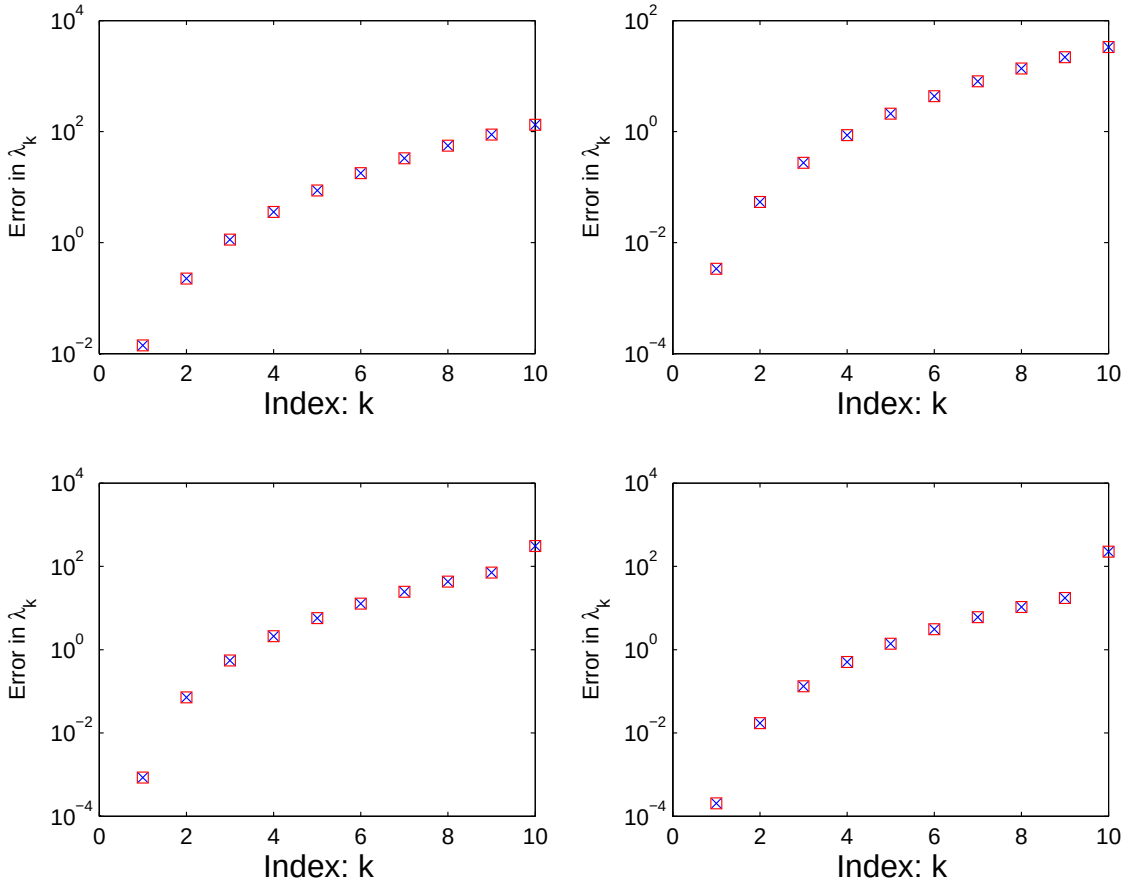

Figure 4.10: We display errors $|\lambda_k - \Lambda_k|$ for different grid sizes, $N = 25$ and $N = 50$. On the left, above we have a graph for problem 1 for $N = 25$ and below we have a graph for problem 2 for $N = 25$. On the right, above we have a graph for problem 1 for $N = 50$ and below we have a graph for problem 2 for $N = 50$.For all graphs (blue x) represent errors in the first 10 numerical eigenvalues when compared to the analytical ones and (red □) correspond to estimated error.

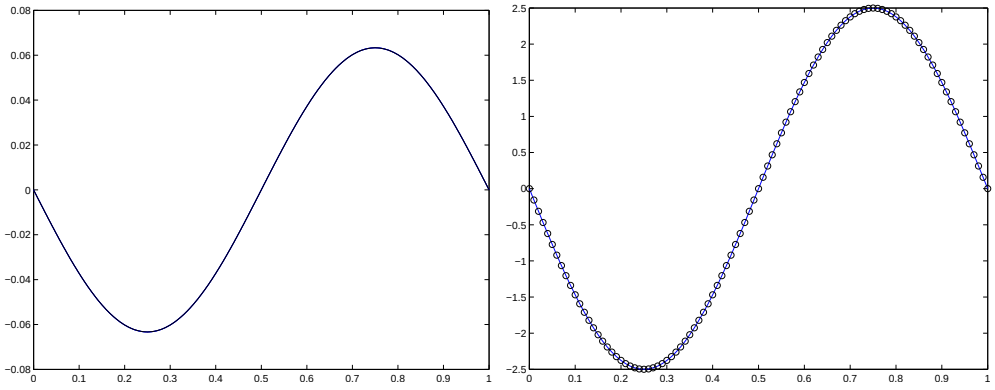

Figure 4.11: On the left, we display the interpolation of the eigenvector $u_2^{(N_2,N_1)}$. On the right (black o) correspond to $\lambda_2^{(N_2)} u_2^{(N_2,N_1)}$ while the whole line correspond to $A^{(N_1)} u_2^{(N_2,N_1)}$ and both lines correspond exactly . This figure correspond to Problem 1.

In Figure 4.12 (left), we illustrate the semilogy plot of actual error corresponding to the first 10 eigenvalues. In this case, we choose a finite difference discretization of size $N = 25, 50, 100, 200$. We also display in Figure 4.12(right) the semilogy plot of the error estimation between the computed and analytical eigenvalues, i.e., $|\lambda_k - \bar{\lambda}_k|$, $k = 1, 2, \ldots, 10$ for problem 1. The results show that the error estimate holds for this case and is relatively sharp.

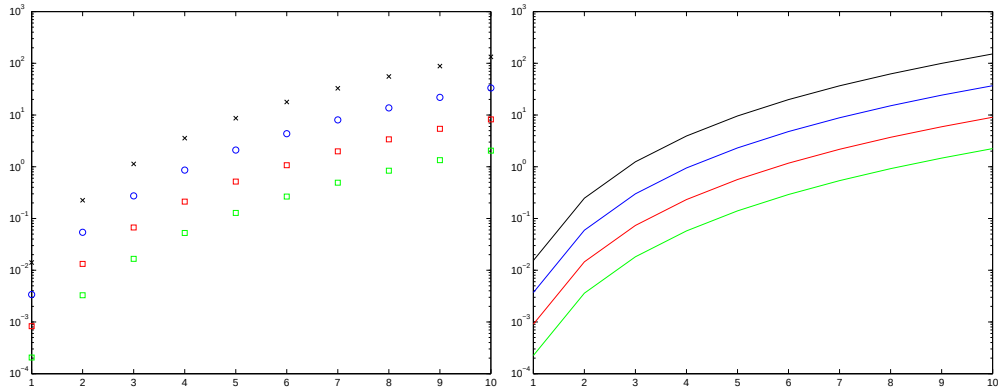

Figure 4.12: The first $n = 10$ eigenvalues λ_k for the Problem 1. On left graph using we display the exact error where (black x) correspond to $N = 25$, (blue o) correspond to $N = 50$ (red □) correspond to $N = 100$ and (green □) correspond to $N = 200$. We also show the semilogy plot of the error estimate $|\lambda_k - \bar{\lambda}_k|$ (right graph) for different grid sizes, $N = 25$ (black line), $N = 50$ (blue line), $N = 100$ (red line) and $N = 200$ (green line).

In Figure 4.13 (left), we present the semilogy plot of actual error corresponding to the first 10 eigenvalues. In this case, we choose a finite difference discretization of size $N = 25, 50, 100, 200$. In Figure 4.13(right) the semilogy plot of the error estimation between the computed and analytical eigenvalues, i.e., $|\lambda_k - \bar{\lambda}_k|$, $k = 1, 2, \ldots, 10$ for problem 2.

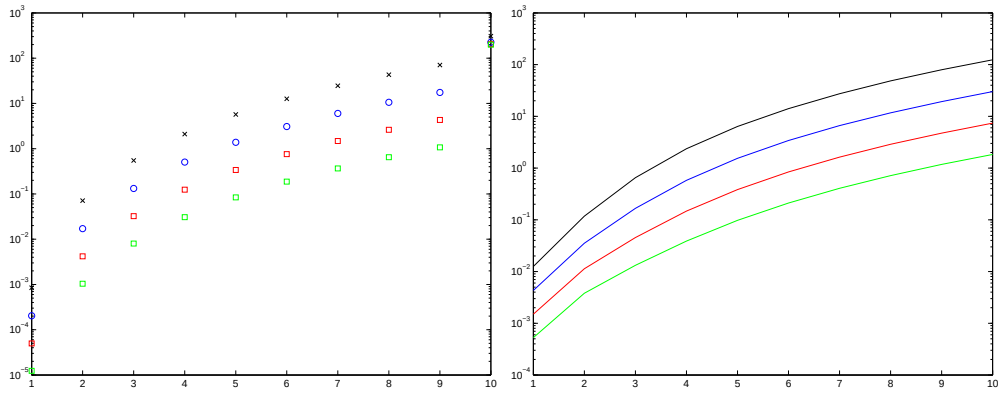

Figure 4.13: The first $n = 10$ eigenvalues λ_k for the Problem 2. On left graph using we display the exact error where (black x) correspond to $N = 25$, (blue o) correspond to $N = 50$ (red $\square$) correspond to $N = 100$ and (green $\square$) correspond to $N = 200$. We alsoshow the semilog plot of the error estimate $|\lambda_k - \bar{\lambda}_k|$ (right graph) for different grid sizes, $N = 25$ (black line), $N = 50$ (blue line), $N = 100$ (red line) and $N = 200$ (green line).

Conclusion and Recommendations

The residual based on error estimate gives a good indication of the actual error for the matrices. So the estimate in the proposition (2.2.2) is usually fairly sharp. By using the numerical eigenvectors and spline interpolation we create approximation eigenfunctions of the original differential operator acting on the approximate eigenfunction. So the residual based error estimate gives us a way to estimate the error in the discretization of the differential operator. For future work there are other interpolation schemes, e.g. trigonometric interpolation, that uses basis functions that can easily be differentiated or integrated, and that can be made to satisfy boundary conditions. One suggestion that we recommend is to investigate alternatives to splines.

Bibliography

[1] Anton Howard, Chris Rorres (2005). *Elementary Linear Algebra*, 9th edition, John Wiley & Sons, New York.

[2] Arfken George B, Hans J. Weber (2011). *Mathematical Methods for Phyisicst*, Miami University, Oxford, OH, University of Virginia, Charlottesville, VA.

[3] Arieh, Iserles (1996).*A First Course in the Numerical Analysis of Differential Equations*, Cambridge Univ. Press.

[4] Berntsson F (2014). *Computational Linear Algebra, 2009*. Linköping University.

[5] Golub, Gene H. "CF van Loan (2013). *Matrix Computation*, 4th edition, The John Hopkins University Press, United States of America.

[6] Hammerling, R., O. Koch, Ch Simon, and E. Weinmüller (2010). *Numerical Solution of Singular Eigenvalue Problems for ODEs with a Focus on Problems Posed on Semi-Infinite Intervals*. Elsevier B.V.Austria.

[7] James V.Lambers (April 11,2013). *Numerical Analysis*. University of Southern Mississippi, USA

[8] King Jonathan, Pawan Dhakal (June 2, 2014). *Finite Difference Schemes and the Schrödinger Equation*. Dartmouth college, united states

[9] Kreyszig, Erwin (1978). *Introductory functional analysis with applications*. Jonh Wiley & Sons.Inc. New York.

[10] Kuldeep Singh (2014).*linear algebra*. First edition, University of Hertfordshire.

[11] Kurt Arbenz et Otto Bachmann (1992). *Eléments d'analyse numérique et appliquée* (Book in French ,First edition,ISBN2-88074-242-0,1992). Presses Polytechniques et Universitaires Romandes, Lausanne.

[12] McKinley, Sky, and Megan Levine (1998). *Cubic spline interpolation.* no. 1 : 1049-1060. College of the Redwoods 45, California.

[13] Michael, T. Heath (1997). *Scientific computing: An introductory survey.* The McGraw-Hill Companies, University of Illinois at Urbana-Champaign.

[14] Olver, Peter J (2008). *Lecture notes on numerical analysis.* Dept. of Mathematics, Uni. of Minnesota.

[15] Panju, Maysum (5 May 2011). *Iterative methods for computing eigenvalues and eigenvectors.* arXiv:1105.1185v1 [math.NA]. University of Waterloo, Canada.

[16] Pollock, D. S. G (1993). *Smoothing with cubic splines.* Queen Mary and Westfield College,The University of London.

[17] Robinson, James C (2004). *An introduction to ordinary differential equations.* Cambridge University Press, New York.

[18] Smith, Gordon D (1993).*Numerical solution of partial differential equations: finite difference methods.* 3rd Edition, Oxford university press.

[19] Strauss, Walter A (2008). *Partial differential equations.* 2nd edition. John Wiley & Sons, Inc, United states.

[20] Thomas, James William (2013).*Numerical partial differential equations: conservation laws and elliptic equations.* Vol. 33. Springer Science & Business Media, Verlag.

[21] Thomas, J. W (1995).*Numerical Partial Differential Equations: Finite Difference Methods.* Vol. 22. Springer Science & Business Media, Verlag.

[22] Zill, Dennis, and Warren Wright (2012). *Differential equations with boundary-value problems.* 8th edition, Cengage Learning, United states.

[23] *www.math.ucsb.edu/ helena/teaching/math124b/eigenvaluenotes.pdf*

[24] *http://www.maths.manchester.ac.uk/ bespalov/teaching/2E1-LA-notes-3.pdf*

Chapter 5

Matlab codes

Appendix A: Analytical eigenvalues matlab code for problem1

```
function[Lambda] = AnalyticalEigenvaluesExample1(N,L)

% This is about the analytical computation of eigenvalues Lambda =
% (n*pi/L)^2
%
Lambda = zeros(N,1);
for n= 1:N
    Lambda(n)= ((n*pi)/L)^2;
end
```

Appendix B:Matlab code to create eigenpairs for problem1

```
function [x,U,lambda,A]=CreateEigenPairsExample1(N,L)


x=(0:N-1)/(N-1);   % x(1)=0 and x(N)=L are the boundary values. So
                   %the unknowns are
                    % u(x(2)),....,u(x(N-1)).
```

```
h=L/(N-1);

  % code for example from your example but with N-2 dimension.

A=sparse(N-2,N-2); %the matrix has only zeros initially
A(1,1)=2/h^2;
A(1,2)=-1/h^2;
A(end,end)=2/h^2;
A(end,end-1)=-1/h^2;
for i=2:N-3
    A(i,i+1)=-1/h^2;
    A(i,i-1)=-1/h^2;
    A(i,i)=2/h^2;
end
 % Solve eigenvalue problem

 [X,D]=eig(full(A));

 % Now the eigenfunctions are X(:,1), X(:,2), etc.

 lambda=diag(D);  % eigenvalues

 % Due to boundary conditions each eigenfunction should have u(0)=0 and
 % u(1)=0. So put en extra zero row first and last in the matrix X to
 % create the full eigenfunction.


 U=[zeros(1,N-2) ; X ; zeros(1,N-2)];

 %
 % Order eigenvalues so smallest absolute value first
 %
   [s,ind]=sort(abs(lambda),'ascend');
   lambda=lambda(ind);
   U=U(:,ind);

end
```

Appendix C:Matlab codes to interpolate eigenfunction for problem1

```
function [pp]=InterpolateEigenfunctionExample1( x , U )

 % Example 1 has zero boundary conditions. These are included in the
 % vector U so just use natural end point conditions.

 pp=csape(x,U,'variational');

end
```

Appendix D:Matlab codes to normalize eigenfunction

```
function[pp]=NormalizeEigenFunction(pp)
I=0;
for i=1:length(pp.breaks)-1,

    a=pp.breaks(i);b=pp.breaks(i+1);
    c=pp.coefs(i,:);
    r=conv(c,c);r=polyint(r);          % conv is multiply so r:= r^2
    I=I+polyval(r,b-a)-polyval(r,a-a); % integrate r^2 on [a,b]

end
I=sqrt(I);
%
% Normalized eigenfunction

pp.coefs=pp.coefs/I;
```

Appendix E: Analytical eigenvalues matlab code for problem2

```
function Lambda = AnalyticalEigenvaluesExample2(n,L)

% Create a function that implements equation (2.7)

f=@(lambda) tan(sqrt(lambda))+sqrt(lambda);

% Create a vector with the first eigenvalues. The first root is zero.
% Ignore

Lambda = 0;
x0=0.1; % starting guess
x=0;
for i=2:n+1  % Should find n eigenvalues

    while (x-Lambda(end))<10^5*eps
      x0=x0+0.1;
      [x,fval,flag]=fzero( f , x0 , optimset('TolFun',100*eps ));
      if flag ~= 1 , x=Lambda(end); , end
    end
    Lambda(i)=x;
end;
Lambda=Lambda(2:end)';
```

Appendix F:Matlab code to create eigenpairs for problem2

```
%Function(A): Matrix(N-2,N-2)
% This function creates a sparse matrix A from the finite difference
%discretization of
% the Dirichlet eigenvalue problem -u''=Lambda u, 0<x<L, with boundary
%conditions u(0)=u(L)+u'(L)=0 and display
```

```
% eigenpair [V,D].

% Input: N- size of the matrix, L=bound
% Output: A-matrix, V eigenvector , D eigenvalues (diagonal)

function [x,U,lambda,A]=CreateEigenPairsExample2(N,L)


x=(0:N-1)/(N-1);   % x(1)=0 and x(N)=L are the boundary values.
                   %So the unknowns are
                    % u(x(2)),....,u(x(N-1)).
h=L/(N-1);

  % code for example from your example but with N-2 dimension.

A=sparse(N-1,N-1); %the matrix has only zeros initially
A(1,1)=2/h^2;
A(1,2)=-1/h^2;

for i=2:N-2
    A(i,i+1)=-1/h^2;
    A(i,i-1)=-1/h^2;
    A(i,i)=2/h^2;
end
A(end,end)=2*(1+h)/h^2;
A(end,end-1)=-2/h^2;

 % Solve eigenvalue problem

 [X,D]=eig(full(A));

 % Now the eigenfunctions are X(:,1), X(:,2), etc.

 lambda=diag(D);  % eigenvalues

 % Due to boundary conditions each eigenfunction should have u(0)=0.
 % So put en extra zero row first in the matrix X to
 % create the full eigenfunction.
```

```
 U=[zeros(1,N-1) ; X ];

 %
 % Order eigenvalues so smallest absolute value first
 %
   [s,ind]=sort(abs(lambda),'ascend');
   lambda=lambda(ind);
   U=U(:,ind);

end
```

Appendix G:Matlab codes to interpolate eigenfunction for problem2

```
function [pp]=InterpolateEigenfunctionExample2( x , U )

 % Example 2 has zero boundary conditions. These are included in the
 % vector U so just use natural end point conditions.

 pp=csape(x,U,[2 1],[0 -U(end)]);

end
```