ON NUMERICAL METHODS FOR ELLIPTIC TRANSMISSION EIGENVALUE PROBLEMS

ANIRBAN ROY
3490 LAFAYETTE DR,
BETHLEHEM, PA, 18020

Project Supervisors: Anna L. Mazzucato and Victor Nistor
Department of Mathematics, Penn State University,
University Park, PA, 16802.

Abstract. The use of numerical tools to solve challenging problems in mathematics has exploded in the past several decades. The purpose of this paper is to compare the results of two different types of numerical methods in finding solutions to the eigenvalue problem for a second order elliptic partial differential equations (PDE) with boundary and transmission conditions. Transmission properties result from jumps in the coefficients of the equation and require more complex numerical methods to solve the eigenvalue problem than when the coefficients are continuous. We present the setup of both the bisection method to solve the exact equation satisfied by the eigenvalues and an application of the power method on a Finite Element Method discretization to find the largest eigenvalues and eigenfunction. We also provide some numerical evidence as to which method is more efficient given the complexities of our problem.

1. Introduction

In this paper we consider the transmission/boundary value problem for a second order, linear, elliptic partial differential equation (PDE), using the Dirichlet boundary conditions. We study the associated eigenvalue problem, both theoretically and numerically. We discuss two numerical methods to obtain the eigenvalues and eigenfunctions. The first is based on the power method to find the dominant eigenvalue and eigenvector of a Finite Element discretization of the problem. The second is a method of computing the eigenvalues that is based on applying the bisection algorithm to solve the exact equation the eigenvalues must satisfy. In the process, we compare rates of convergence, as well as estimate the number of eigenvalues in a given interval.

The paper is organized as follows. We being our study by introducing the type of equation that we are dealing with and the transmission conditions that will be in effect on our problem. We continue by discussing the basis of piecewise linear functions that we employ for the Finite Element Method (FEM), in particular

Date: March 4, 2011.

a basis of hat functions. We then construct the stiffness and mass matrices and discuss their use in solving our problem. With the setup of the two matrices in place, we show how to employ the power method, a type of iterative method common in numerical analysis, to compute the largest eigenvalue. We also run through an alternative approach to computing the eigenvalues problem called bisection method. We then finish by comparing the advantages and drawbacks of both methods used.

1.1. **Elliptic Transmission Problems.**

The equations we consider involve 2nd-order, linear, differential operators of the form:

$$-(\alpha(x)u')',$$

where u is a function on the interval $[0,1]$, α is a given coefficient, and $u'(x) := \frac{du}{dx}(x)$. This operator is called *elliptic*, if $\alpha(x) \geq \gamma > 0$, for some positive constant γ [1]. The specific form of the operator, as opposed to considering $\alpha(x)u''$, which is also a second-order elliptic operator, is explained later on.

We are interested in studying, in particular, *eigenvalue problems* for these operators. Finding eigenvalues and eigenfunctions requires that u satisfies appropriate *boundary conditions* at the endpoint of the interval. For example, the sound harmonics in music instruments are eigenfunctions of an elliptic operator. In string instruments, the string is tied down at the ends, which corresponds to setting $u = 0$ at the endpoint, as u represent the sound amplitudes. This type of boundary condition is called a *Dirichlet* condition. In brass instruments, on the other hand, the air influx is prescribed at the mouth of the instrument by the player, that is, u' is given. This type of boundary condition is called a *Neumann* condition.

We begin by studying the following model eigenvalue problem:

$$\begin{cases} -(\alpha(x)u')' = \lambda u(x), & 0 < x < 1, \\ u(0) = 0 = u(1), \end{cases} \tag{1}$$

where λ represents an *eigenvalue*, a number satisfying $Au = \lambda u$ for some function u not identically zero, with A being the differential operator $\alpha((x)u')'$. Then, u is called an *eigenfunction* of A. What makes eigenvalue problems difficult is that we need to determine both λ and u.

In general, $\alpha(x)$ is a function, allowed to have jumps at some point in $[0,1]$. Because of the complexity of this problem we take $\alpha(x)$ only of the following form with c a non-zero constant:

$$\alpha(x) = \begin{cases} 1, & 0 \leq x \leq 1/2, \\ c^2 & 1/2 \leq x \leq 1. \end{cases} \tag{2}$$

Since $\alpha(x)$ is piecewise constant but with a jump at $1/2$ there will be *transmission conditions* on u.

Partial differential equations of the elliptic type arise in many fields of science and engineering [2]. For example, they model the dependence of the electrostatic potential on the electric charge in a conductor. They also model static deformation in elastic solids. The coefficients appearing in the equations are material parameters. Real materials are often composite and the material parameters then exhibit jumps across internal surfaces. These surfaces are often called *interfaces*. When interfaces are present, solutions to the PDE must satisfy extra conditions, called *transmission conditions*. The presence of jumps at interfaces can cause scattering of waves, such as in a vibrating string made of two different types of materials [3]. Transmission problems arises often in optics as well as various physics and electrical engineering problems [2].

Studying PDE with jumps in the coefficients is more difficult and the corresponding numerical methods used to solve them more complex, than in that case the coefficients are continuous. The purpose of this paper is to study a model transmission problem in one dimension on an interval, imposing Dirichlet boundary conditions. In this case, the equations reduce to ordinary differential equations (ODE), but the difficulties introduced by the jumps in the coefficients are still present.

To justify the transmission conditions, we start with the so-called *weak formulation* of the problem (1):

$$\int_0^1 \alpha(x)u'(x)\, v'(x)\, dx = \int_0^1 \lambda uv\, dx. \tag{3}$$

Formally, this formulation is obtained by pairing the equation with another suitable function v, called a *test function*, integrating over $[0,1]$, and then integrating by parts. The function v is chosen to satisfy the same Dirichlet boundary conditions as u, namely:

$$v(0) = 0 = v(1),$$

so that there are no extra terms coming from the integration by parts. In the weak formulation, however, the coefficient α is not differentiated, and hence the integral is well defined as long as u and v are differentiable.

To justify rigorously the integration by part, we pair the equation in (1) with v and integrate:

$$-\int_0^1 (\alpha(x)u'(x))\, v(x)\, dx = \int_0^1 \lambda uv\, dx.$$

We split the integral on the left hand side, given that $\alpha((x)u')'$ has a jump at $x = 1/2$ (see equation (2)):

$$-\int_0^{1/2} (u')'v\, dx - \int_{1/2}^1 (c^2u')'v\, dx = \int_0^1 \lambda uv\, dx. \tag{4}$$

Splitting up this integral allows us to have continuous functions in each integrand. We can then integrate by parts in each integral, obtaining the following:

$$-[u'v]_{x=0}^{x=1/2} - [c^2u'v]_{x=1/2}^{x=1} + \int_0^1 \alpha(x)u'v'\,dx = \int_0^1 \lambda uv\,dx. \tag{5}$$

We can see given the boundary conditions that $[u'v]_{x=0}$ and similarly $[c^2u'v]^{x=1}$ will be equal to 0. In order to obtain the weak formulation (3), we need to impose some conditions at the jump site $x = 1/2$, more specifically we must impose the following transmission condition:

$$u'(1/2)_- = c^2u'(1/2)_+, \tag{6}$$

where the subscripts denotes respectively the limits from the left and right. In order to enforce continuity of u, we need to have the following additional transmission condition

$$u(1/2)_- = u(1/2)_+. \tag{7}$$

The importance of the weak formulation will become more apparent later on as we discuss the transcendental equation for use in bisection method.

Furthermore, the weak formulation lends itself naturally to methods that rely on expanding u in certain basis of functions. One of these methods is the *Finite Element Method* (FEM for short), where the basis is given by piecewise polynomial functions [4]. We will use piecewise linear, "hat" functions for simplicity, and describe the construction of the basis in details next in Section 2.1. We will then use an iterative method called the *power method* [4] in order to find the highest eigenvalue and corresponding eigenfunctions. The application of this method is more difficult in the presence of jumps in the coefficients.

In this simple, model problem, we can solve explicitly for the eigenfunctions in (1) in each subinterval $0 < x < 1/2$ and $1/2 < x < 1$. To find the eigenvalues we then need to solve a transcendental equation, which we do using the so-called *bisection method* [4].

The purpose of this paper is to discuss the differences in convergence of these different methods for the largest eigenvalue, as well as their efficiency.

2. Finite Elements and the Power Method

2.1. **Basis of Piecewise Linear Functions.** Partial Differential Equations are found in many mathematical models and real world problems. However often analytical solutions of real-life problems are not available, and we are forced to rely on numerical methods. One such method is the Finite Element Method or FEM, which is a method for solving differential equations based on expanding the solution in a suitable basis of known functions, usually piecewise polynomial

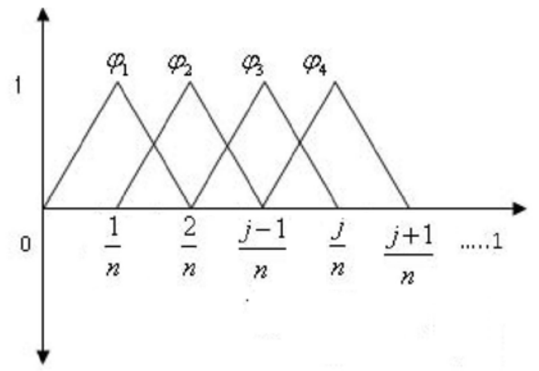

FIGURE 1. Basis of of Piecewise Linear Functions

functions. Here we use piecewise linear functions. The rest of this section is devoted to how we construct a FEM in terms of hat functions on the real line [5].

We begin with discussing the following basis, which we denote by S_n, of hat functions on the interval $[0, 1]$. We will then write the solution u of (3) as an appropriate sum of the functions in the basis. Each function in the basis vanishes at $x = 0$, 1, so that the boundary conditions in (1) are automatically satisfied. Furthermore each function is continuous, so that the second transmission condition (7) is also satisfied automatically. Each basis function is referred to as a *finite element* in the method.

We start by *partitioning* the interval $[0, 1]$ into n equal subintervals, where n is a positive integer. This gives rise to $n+1$ *nodes* of the form $\frac{j}{n}$, $j = 0, \ldots, n$. Later on, we will choose $n = 2k$ for some integer k, so that $1/2$ is a node and hence the jump in the coefficient α occurs at a node. This simplifies our analysis, as we discuss more later on.

For each $1 \leq j \leq n-1$, we construct a "hat" function ϕ_j, as follows.

Each ϕ_j is a hat function with height of 1 (see Figure 2.1), which is non zero exactly on the interval $\dfrac{j-1}{n} < x < \dfrac{j+1}{n}$. Hence, the graph of ϕ_j consist of two lines that we denote respectively as $ax + b$ and $cx + d$, that is:

$$\phi_j(x) = \begin{cases} ax + b, & \dfrac{j-1}{n} \leq x \leq \dfrac{j}{n}, \\ cx + d, & \dfrac{j}{n} \leq x \leq \dfrac{j+1}{n}. \end{cases} \tag{8}$$

We know that for any j, $\phi_j(\dfrac{j}{n}) = 1$, $\phi_j(\dfrac{j-1}{n}) = 0$, and $\phi_j(\dfrac{j+1}{n}) = 0$. We also know that the line segments meet at the point $(j/n, 1)$. Using these facts, we obtain a system of linear equations for the coefficients a, b:

$$a(\frac{j-1}{n}) + b = 0, \qquad (9)$$
$$a(\frac{j}{n}) + b = 1.$$

This system is easily solved and results in $a = n$ and $b = 0$. We similarly find that $c = -n$ and $d = 0$, so that ϕ_j and its derivative ϕ_j' are respectively given by:

$$\phi_j(x) = \begin{cases} nx - j + 1, & \frac{j-1}{n} \le x \le \frac{j}{n}, \\ -nx + j + 1, & \frac{j}{n} \le x \le \frac{j+1}{n}, \end{cases} \qquad (10)$$

and

$$\phi_j'(x) = \begin{cases} n, & \frac{j-1}{n} \le x \le \frac{j}{n}, \\ -n, & \frac{j}{n} \le x \le \frac{j+1}{n}. \end{cases} \qquad (11)$$

We will seek the solution u of (1) in the form:

$$u(x) = \sum_{j=1}^{n-1} c_j \, \phi_j(x). \qquad (12)$$

This is in general only an approximate solution, but it will become closer and closer to the true solution as $n \to \infty$. In practice though n is finite and cannot be taken too large.

2.2. **The Stiffness Matrix.** The idea of the FEM is that the integrals appearing in equation (3) are easily computed exactly if u is replaced by the ϕ_j's, so that we reduce the eigenvalue problem (1) to solving a linear system of algebraic equations for the c_j defined in equation (12) (see for example [5]). There are many efficient methods for finding eigenvalues of matrices, even when n, which gives the size of the matrix, is large [4].

The matrix obtained by explicitly computing the integrals for the ϕ_j's is usually called the *stiffness* matrix if it involves the derivatives and the *mass matrix* if it involves the ϕ_j's only (the names refer to applications of the FEM to problems with elastic materials).

2.3. **Stiffness Matrix when** $\alpha(x) = 1$**.** We are now going to construct the stiffness matrix in the case that the coefficient α is constant (we can then always set it equal to 1). These matrices will be used to apply the power method for the eigenvalues, which we will introduce later. The more difficult case when α has a jump is discussed in the next subsection.

The elements $A_{i,j}$, $i, j = 1, \ldots n-1$ of the stiffness matrix A can be computed by the following integral:

$$A_{i,j} = \int_0^1 \alpha(x)\phi_i'(x)\,\phi_j'(x)\,dx. \tag{13}$$

There are four different cases to consider, namely: $i = j$, $i = j+1$, $i = j-1$, and lastly $|i-j| > 1$. In the first case $i = j$, using (11) we have

$$A_{i,i} = \int_{\frac{j-1}{n}}^{\frac{j+1}{n}} n^2\,dx = 2n.$$

Similarly, we find that

$$A_{i,j} = -n, \qquad \text{for } i = j+1,\ i = j-1.$$

(In fact, the matrix A is symmetric.) On the other hand, (11) also shows easily that, if $|i-j| > 1$, there is no overlap between the two finite elements, so that the integral is 0.

Therefore we have the matrix A in the following form;

$$\begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & 0 & A_{i-1,i-2} & A_{i-1,i-1} & A_{i-1,i} & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & A_{i,i-1} & A_{i,i} & A_{i,i+1} & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & A_{i+1,i} & A_{i+1,i+1} & A_{i+1,i+2} & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix},$$

From the calculations of the entries we did above, we have more specifically

$$\begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & 0 & -n & 2n & -n & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & -n & 2n & -n & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & -n & 2n & -n & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}. \tag{14}$$

2.4. **Stiffness Matrix for the Transmission Problem.**

Let us now take a look at a more complicated example where the coefficient α has a jump at $1/2$, equation (2).

Recall that:

$$A_{i,j} = \int_0^1 (\alpha(x)\phi_i'\phi_j')\,dx.$$

For convenience, we choose $n = 2k$. This choice allows the jump to occur exactly at the node k/n.

We can use the computed values for $A_{i,j}$ from Section 2.3 for the cases when i and j are both less than k or greater than k. In this last case, in fact, the only change that occurs is that the integrals change by multiplication with the constant

c^2, as a result of the jump in the coefficient α. We therefore need to compute one new integral when $i = j = k$:

$$A_{k,k} = \int_{1/2-1/2k}^{1/2} \phi_i' \phi_j' \, dx + \int_{1/2}^{1/2+1/2k} c^2 \phi_i' \phi_j' \, dx = n^2 \, x\big|_{1/2-1/2k}^{1/2} + c^2 n^2 \, x\big|_{1/2}^{1/2+1/2k} = n(c^2+1). \tag{15}$$

Consequently, the stiffness matrix has a form similar to that derived in Section 2.3 with entries:

$$\begin{array}{cc} & \text{column } k \\ & \left[\begin{array}{ccccccccc} \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & -n & 2n & -n & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & -n & n(1+c^2) & -nc^2 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & -nc^2 & 2nc^2 & -nc^2 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{array}\right] \end{array}. \tag{16}$$

with the middle row labelled row k.

2.5. **The Mass Matrix.** We saw before the construction of the stiffness matrix using finite elements. We now apply the same construction to the mass matrix. The mass matrix is obtained in a similar manner as the stiffness matrix, but it does not involve the coefficient α and it uses $\phi_i \; \phi_j$, not their derivatives.

This second matrix is necessary to solve eigenvalue problems, and inhomogeneous problems, as well, using a Finite Element discretization. The computation of each integral is now more complicated as ϕ_j depends on x, while its derivative is piecewise constant.

The mass matrix is defined by

$$M_{i,j} = \int_0^1 (\phi_i \phi_j) \, dx, \tag{17}$$

as it can be seen by (3). Hence, the construction of the mass matrix will not need to account for the jump since $\alpha(x)$ does not appear in the definition. Therefore we only need to consider three cases, $i = j$, $i = j-1$ and lastly $i = j+1$, noting as before that, when $|i-j| > 1$, $M_{i,j} = 0$. We will go into the details for the case $i = j$ case and leave it to the reader to compute the other two integrals. When $i = j$ we have

$$M_{j,j} = \int_{\frac{j-1}{n}}^{\frac{j}{n}} (nx-j+1)(nx-j+1) \, dx + \int_{\frac{j}{n}}^{\frac{j+1}{n}} (-nx+j+1)(-nx+j+1) \, dx, \tag{18}$$

We observe that by making the change of variables $z = j/n - x$ in the second integral on the right-hand side of the above equation, we can rewrite:

$$M_{j,j} = 2\int_{\frac{j}{n}}^{\frac{j+1}{n}} n^2(x-(j+1)/(n))^2\,dx. \tag{19}$$

By a further change of variable $y = x - \dfrac{j+1}{n}$ we can reduce the integral to:

$$M_{j,j} = 2\int_{-1/n}^{0} n^2\,y^2\,dy = 2\int_{0}^{1/n} n^2\,y^2\,dy = \frac{2}{3n}, \tag{20}$$

where we used that the integrand is an even function of y. The remaining integrals require similar change of variables and we leave it to the reader to compute their values.

We can use these values to construct the mass matrix, which is again symmetric and tridiagonal. The matrix has the following form:

$$\frac{1}{6}\begin{bmatrix} \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & -n & 4n & -n & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & -n & 4n & -n & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & -n & 4n & -n & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}. \tag{21}$$

Our goal is to find the dominant eigenvalue for the problem (1). Using the mass and stiffness matrices constructed with finite elements, we are going to compute the largest eigenvalue by the power method. In certain cases one might be more interested in the fundamental eigenvalue or the smallest eigenvalue depending on the problem at hand. The inverse iteration method is a method used to find the smallest eigenvalue (see for example [6]). We will exclusively discuss the power method here.

2.6. **Iterative Implementation of the Power Method.**

In this section we implement the power method to find the largest eigenvalue and corresponding eigenvector for the problem (1) numerically.

The power method is an iterative method used to compute a matrix's dominant eigenvalue and eigenvector [4]. It requires that the matrix has a simple eigenvalue of largest norm and a basis of eigenvectors, that is, n linearly independent eigenvectors. By a simple eigenvalue we mean an eigenvalue with a single eigenvector. We will see below that these requirements are satisfied in our case.

In the weak formulation (3), it is enough to take $v = \phi_j$ for each j, since S_n is a basis. Then, by using the decomposition (12) for u, the weak formulation reduced to the following linear matrix eigenvalue problem:

$$AC = \lambda MC,$$

where the C is the column vector with entries c_j, $j = 1, \ldots, n-1$, defined in (12), A and M are respectively the stiffness and mass matrices. Here, λ and C are the unknowns. From (21) it is easy to see that M is invertible, then, the above equation is equivalent to

$$M^{-1}A\,C = \lambda C, \tag{22}$$

which is an eigenvalue problem for the symmetric matrix $B = M^{-1}A$. Hence, we must invert the mass matrix to setup for power method.

We also note that both A and M have entries that are multiplied each by powers of n. We can go ahead and factor out these factors of n from both matrices:

$$A \;=\; n \begin{bmatrix} \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & 0 & -1 & 2 & - & 0 & 0 & 0 & \ldots \\ \ldots & 0 & 0 & -1 & (1+c^2) & -c^2 & 0 & 0 & \ldots \\ \ldots & 0 & 0 & 0 & -c^2 & 2c^2 & -c^2 & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}, \tag{23}$$

$$M \;=\; \frac{1}{6n} \begin{bmatrix} \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & 0 & -1 & 4 & -1 & 0 & 0 & 0 & \ldots \\ \ldots & 0 & 0 & -1 & 4 & -1 & 0 & 0 & \ldots \\ \ldots & 0 & 0 & 0 & -1 & 4 & -1 & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}, \tag{24}$$

so that we can write

$$A = n\,\tilde{A}, M = \frac{1}{6n}\tilde{M},$$

where $\tilde{A}$ and $\tilde{M}$ are independent of n. With abuse of notation, we write A and M below for $\tilde{A}$ and $\tilde{M}$. This factorization is important in the computation as n can be a large number in practice. With this factorization, the matrix eigenvalue problem becomes:

$$B\,C = \tilde{\lambda}C, \tag{25}$$

where $B = M^{-1}\,A$, and $\tilde{\lambda} = \dfrac{\lambda}{6n^2}$.

Before can implement Power Method we need to find the inverse of M. M is a tridiagonal symmetric matrix we can therefore use the following algorithm for an efficient computation of the inverse. The first step in creating an efficient algorithm for the inverse is to form a new matrix D which contains only the diagonal elements of M.

Then, the inverse of D is given simply by inverting those elements, and can be computed fast. Next, we note that

$$M \;=\; D - A' \;= D\,(I - D^{-1}\,A'),$$

where A' is the matrix of the negative of the off-diagonal elements of M, and I is the identity matrix.

The inverse of the mass matrix has hence the following form.

$$M^{-1} = (1 - D^{-1}A')^{-1}D^{-1}. \tag{26}$$

We define for convenience a new matrix $d = D^{-1}A'$. The point is that d is "small" so that we can approximate $(1-d)^{-1}$ in this way:

$$(1-d)^{-1} \approx 1 + d^2 + d^3 + ... + d^{20} + \dots \tag{27}$$

In practice, we find that in our problem it is enough to take only the first 20 terms. Taking more terms in this expansion has little to no impact on the computed entries to the numerical accuracy we use. By multiplying this approximation by D^{-1} we get an efficient approximation for M^{-1}.

After computing the inverse as above we can now implement the power method. With the inverse of M we can compute B from (25) we have that $BC = \tilde{\lambda}C$, with $\tilde{\lambda}$ and C being the unknowns. To this matrix B we now apply the power method. We note now that the hypotheses necessary for the power method are satisfied. In fact, since both M and A are symmetric, B is symmetric as well, so that it has a basis of eigenvectors. That the dominant eigenvalue is simple follows from the fact that A and M are obtained from the discretization of an elliptic problem (see for example [1]).

We start by selecting an initial guess for C or choose a random vector. If we let U_0 be our initial guess for C, we then set recursively:

$$U_{\ell+1} = \frac{BU_\ell}{\|BU_\ell\|}, \tag{28}$$

where $\| \, \|$ stands for some standard matrix norm (here, we take the standard matrix norm in MATLAB). That is, we multiply U by B and normalize the result at every iteration. The power method is based on the fact that U_ℓ will converge to the eigenfunction corresponding to the largest eigenvalue of B as ℓ approaches infinity [4]. (This eigenvalue is simple, so there is only one such eigenfunction.)

Furthermore, the largest eigenvalue can be found by simply dividing the norm of $\ell+1$ iteration of the eigenvector by that of the ℓ iteration in the limit as ℓ goes to infinity:

$$\tilde{\lambda} \approx \frac{\|U_{\ell+1}\|}{\|U_\ell\|}. \tag{29}$$

So, we can get a good approximation to the largest eigenvalue and corresponding eigenfunction if ℓ is large, but finite.

We must keep in mind from (25) that the approximation of the dominant eigenvalue for the problem (1) is

$$\lambda = 6n^2\,\tilde{\lambda},$$

hence the eigenvalue becomes larger and larger as n grows. In fact, the largest eigenvalue for the problem (1) is infinitely large.

2.7. **Numerical Tests.**

In this section we will discuss some of the results from implementing the power method in MATLAB. To give an example, we choose $n = 7$, and display the first 10 iterations of the power method.

The following tables show each component of the eigenvector for iteration 1 through 10.

		Eigenvector Elements by Position						
Iteration	$\tilde{\lambda}$	1	2	3	4	5	6	7
1	2.1024	-.1622	.2564	-.0672	.2903	-1.2551	1.5258	-.5797
2	2.4921	-.0473	.0871	.0427	.4076	-1.5228	1.7324	-.8440
3	2.5342	-.0094	.0354	.0793	.4462	-1.5394	1.7196	-.9427
4	2.5409	.0002	.0223	.0887	.4598	-1.5282	1.7055	-.9962
5	2.5430	.0026	.0189	.0905	.4630	-1.5144	1.6997	-1.0304
6	2.5440	.0031	.0179	.0902	.4614	-1.5015	1.6986	-1.0541
7	2.5445	.0032	.0175	.0895	.4579	-1.4901	1.6995	-1.0715
8	2.5448	.0032	.0173	.0887	.4438	-1.4803	1.7012	-1.0848
9	2.5450	.0032	.0172	.0879	.4499	-1.4719	1.7031	-1.0952
10	2.5452	.0032	.0170	.0872	.4463	-1.4649	1.7050	-1.1036

TABLE 1. Eigenvectors for the first 10 iterations, $n = 7$.

For a small 7×7 matrix, by the fifth iteration we see that after each iteration the change in the eigenvector becomes smaller and smaller indicating the convergence to a limit vector. This limit is in fact the true eigenvalue of the discrete problem, as it follows by looking at the algorithm for the power method (see again [4] for details).

It is important to note that the convergence rate depends on the choice of the initial random vector. In this case, we have performed several numerical tests with different initial choices and we find the converegence rate to be comparable to the one shown here.

We study numerically the error in the method by also tabulating the difference between the computed eigenvalue at one iteration with the computed eigenvalue at the next one.

Error $n = 7$	
Iterations	$\lvert\tilde{\lambda}_i - \tilde{\lambda}_{i_1}\rvert$
1	-
2	.39
3	.0421
4	.0067
5	.0021

TABLE 2. Change in the computed eigenvalue between each iteration

We next investigate the efficiency of the method depending on the size of the matrix n. Below we tabulate the components of the eigenvector in this case again for iterations 1 through 10 and the corresponding numerical error in the computed eigenvalue. For $n = 20$ we only show some of the components.

We should point out that we cannot strictly compare the results for different values of n, as they pertain to different discretization s of the original problem.

		Eigenvector Elements by Position						
Iteration	λ	14	15	16	17	18	19	20
1	2.4126	-.6373	.9046	-.5527	-.7136	1.5635	-1.0428	.1526
2	2.4908	-.7328	.9378	-.4886	-.7428	1.5594	-1.1160	.2797
3	2.5168	-.7882	.9404	-.4235	-.7521	1.5208	-1.1585	.3830
4	2.5328	-.8267	.9301	-.3676	-.7552	1.4813	-1.1899	.4672
5	2.5440	-.8548	.9123	-.3199	-.7546	1.4456	-1.2137	.5353
6	2.5522	-.8757	.8904	-.2789	-.7519	1.4143	-1.2318	.59
7	2.5583	-.8910	.8666	-.2432	-.7481	1.3872	-1.2454	.6336
8	2.5630	-.9021	.8423	-.2119	-.7440	1.3638	-1.2555	.6683
9	2.5667	-.9099	.8184	-.1841	-.7402	1.3436	-1.2627	.6986
10	2.5696	.9152	.7953	-.1593	-.7368	1.3262	-1.2676	.7171

TABLE 3. Iterations 1-10 of Eigenvector for $n = 20$.

Error $n = 20$	
Iterations	$\|\tilde{\lambda}_i$ - $\tilde{\lambda}_{i_1}\|$
1	-
2	.0782
3	.026
4	.016
5	.0112

TABLE 4. Change in eigenvalue's between iterations, $n = 20$.

In comparing the case $n = 7$ with the case $n = 20$, we see that they both slowly converge to the eigenvalue, as expected. However, it seems *initially* that for $n = 20$ the method gives a better approximation (compare, for instance, the computer eigenvector entries at the second iteration). This may be attributed to having a better initial guess for the starting vector for $n = 20$ then $n = 7$.

On the other hand, the larger the size of the matrix, the slower the convergence of the series (27), which may lead to inaccuracies. This may account for the fact the error in the computed eigenvalue is larger for higher iterations when $n = 20$.

In our case the power method is a very useful way of attaining the dominant eigenvalue and eigenvector. This iterative method was chosen because of the sparsity of our matrix B and will give us an accurate approximation without needing as much computer power as more direct methods, for example, those using Gaussian elimination to invert the matrix.

One of the major drawbacks to power method is finding the eigenvalue for a high value of n. The need to take the matrix inverse creates a great need for computational power, and without the assistance of significant computational power, this method may not work with larger matrices. It is important to note that we computed the inverse of the matrix M by (26) and (27) prior to running power method. There are ways, discussed for instance in [6], to apply the power method without explicitly computing the matrix inverse. This text has algorithms on how to avoid directly computing the inverse and instead solving a set of linear equations and then run iterative methods similar to the power method. There is a general method to find all eigenvalues of a matrix called the QR factorization, but it is difficult to study and implement, and we will not discuss it here. In general, we expect that computing eigenvalues of matrices for large values of n will be slow and computationally expensive.

3. The Bisection Method

We are now going to change our approach to solving the problem (1). Rather than using the FEM, we are now going to use a more elementary approach and solve the ODE in (1) directly. Because of the jump in the coefficient α when $x = 1/2$, we cannot directly integrate the ODE on the interval $[0, 1]$, and we need to split the interval. We then have to solve:

$$\begin{cases} u''(x) = \lambda u(x), & 0 < x < 1/2, \\ c^2 u''(x) = \lambda u(x), & 1/2 < x < 1. \end{cases} \tag{30}$$

Imposing both the boundary and transmission conditions, we find only the trivial solution for $\lambda = 0$, $\lambda > 0$. For $\lambda < 0$, we easily integrate both of the equations twice in order to come up with the form of the solution. We substitute $\lambda = -\mu^2$ for convenience in the calculations. Then, the solution has the form

$$u(x) = u_1(x) := B_1 \cos(\mu x) + D_1 \sin(\mu x), \quad 0 < x < 1/2, \tag{31}$$

and

$$u(x) = u_2(x) := B_2 \cos(\frac{\mu}{c})x + D_2 \sin(\frac{\mu}{c})x, \quad 1/2 < x < 1. \tag{32}$$

The transmission conditions must be satisfied $u_1(1/2)' = c^2\, u_2(1/2)'$, $u_1(1/2) = u_2(1/2)$, as well as the boundary conditions $u_1(0) = 0, u_2(1) = 0$. Applying these conditions to (31), (32) results in the following linear system for the unknowns B_1, D_1, B_2, D_2:

$$\begin{gathered} -D_1 \sin(\frac{\mu}{2}) + B_2 \cos(\frac{\mu}{2c}) + D_2 \sin(\frac{\mu}{2c}) = 0 \\ -D_1(\cos\frac{\mu}{2}) - B_2/c \sin(\frac{\mu}{2c}) + D_2/c \cos(\frac{\mu}{2c}) = 0 \\ 0 + B_2 \cos(\frac{\mu}{c}) + D_2 \sin(\frac{\mu}{c}) = 0 \end{gathered} \quad . \tag{33}$$

This system is homogeneous and therefore it has non-zero solutions if and only if the determinant of the coefficient matrix is zero. Taking the determinant and setting it equal to zero results in the following equation for μ:

$$f(\mu) = \sin(\mu/2)\cos(\mu/2c) + c\cos(\mu/2)\sin(\mu/2c) = 0. \tag{34}$$

Remember that the eigenvalues are the numbers λ that satisfy $Au = \lambda u$, where in this case A is the differential operator $\frac{d}{dx}(\alpha(x)\frac{d}{dx})$. The eigenvalues are given here as $\lambda = -\mu^2$ where μ is precisely mapped to zero by f. Knowing the eigenvalues allow us to determine the eigenfunction from (31) and (32).

Now the task of finding a valid solution has reduced to get *non-zero* values of μ such that $f(\mu) = 0$. Looking at equation (34), we see that it is not a trivial task to find all the zeros of f, since f is a transcendental function. We therefore need to find the zeros numerically.

In order to do this we introduce a method common in Numerical Analysis referred to as the *bisection method* [4].

3.1. **Intervals for the Bisection Method.**

The bisection method is based on the intermediate value property for continuous functions. The intermediate value property states that, given any continuous function f on $[a, b]$, if β is in between $f(a)$ and $f(b)$, then there exists a $e \in (a, b)$ such that $\beta = f(e)$. In this case $\beta = 0$. We know the function f in equation (34) is continuous, so if we can find an interval $[a, b]$ such that $f(a) < 0$ and $f(b) > 0$ then we can find a point $e \in (a, b)$ such that $f(e) = 0$.

The bisection method uses the intermediate value theorem to find a zero of continuous functions that have the property $f(a) \cdot f(b) < 0$, so that we are guaranteed a zero in $[a, b]$. With $e = \frac{1}{2}(a + b)$, we then check whether $f(a) \cdot f(e) < 0$. If this is true, then f has a zero in $[a, e]$.

If not, then $f(a) \cdot f(e) > 0$ and hence f must have a zero in the other interval $[e, b]$.

So, we can assume that $f(a) \cdot f(e) < 0$ without loss of generality. We then replace e for b, and we now have a new interval which is half as large as the original interval $[a, b]$, where f must have a zero.

If $f(a) \cdot f(e) > 0$ and $f(e) \cdot f(b) < 0$ our left endpoint changes from a to $e = \frac{1}{2}(a + b)$. a has now been changed to e and a new interval being half the previous size still contains a zero value for f.

We keep repeating this process until we find the value that f maps to zero by the condition $f(a) \cdot f(e) = 0$. The bisection method is called the method of interval halving as with each repetition the intervals size gets halved each time but preserve a zero for f. In general, it will take infinitely many iterations to find the zero, but in practice once can stop when the value of f become comparable to machine precision.

We are going to implement bisection method to find the solutions of (34) in order to find the eigenvalues for our problem. The trick is to find an interval to start from. We observe that f is zero when $\sin(\frac{\mu}{2})$ and $\sin(\frac{\mu}{2c})$ are both zero or $\cos(\frac{\mu}{2})$ and $\cos(\frac{\mu}{2c})$ are both zero . Since sin and cos are never zero at the same point, we can divide equation (34) after rearranging it by $\cos(\frac{\mu}{2c})\cos(\frac{\mu}{2})$, to get a new function:

$$\tilde{f}(\mu) = \tan(\frac{\mu}{2}) + c\tan(\frac{\mu}{2c}) = 0. \tag{35}$$

When the denominator is zero, the graph of $\tilde{f}$ has a vertical asymptotes. At every other point, the function $\tilde{f}$ is well defined. The vertical asymptotes occur precisely at the points where $\frac{\mu}{2} = \frac{\pi}{2} + l\pi$ for some integer l, or $\frac{\mu}{2c} = \frac{\pi}{2} + k\pi$ for some other integer k.

Since the tangent goes from negative infinity to infinity in these intervals, we know there will be exactly one zero between two consecutive of these multiples, given that the tangent is a strictly increasing function and the sun of two increasing functions is increasing. For special values of c, precisely when:

$$c = \frac{2k+1}{2l+1}, \qquad k, l \in \mathbb{Z}, \tag{36}$$

the asymptotes correspond to zeros of f as well. For simplicity, we assume that c is different than once of these values.

Therefore, if we set $\mu_l = \pi + 2l\pi$ $\mu_k^c = \pi c + 2ck\pi$ for constant integers l and k, these will help determine values for our interval to apply the bisection method. The points μ_l, μ_k^c, will cut the real line into intervals, we know there will be exactly one zero in each interval.

3.2. **Implementing the Bisection Method.**

In order to apply bisection method as discussed in the previous section, we need to find the starting intervals where $f(\mu) = 0$ will exist. We find the intervals which contain a zero of the function defined in (35) will lie between the values for $\mu_l = \pi + 2l\pi$ $\mu_k^c = c\pi + 2ck\pi$, l and k integers, and therefore this depends on the value of c. We can construct an array such that it creates the endpoints to which we can apply the bisection algorithm upon. We can set values for l and k and given the jump c we can list the resulting μ_l and μ_k^c sequentially in an array.

The following code is in MATLAB, but it can be arranged in any other programming language:

```
c=4;
L=0;
K=0;
T= 50+50/c;
for j=1:T;
```

```
    if ((2*K +1) > (2*L+1)*c)
        a(j)=(2*L+1)*c*\pi;
        L=L+1$;
    else
        a(j)=(2*K+1)*\pi;
        K=K+1;
    end;
end.
```

After running this code we now have an array a, containing the possible intervals that can contain a zero of $\tilde{f}$, corresponding to an eigenvalue for our problem. These are the same intervals that we may perform bisection method upon. Table 5 shows the first six numbers that the bisection algorithm can use as starting values listed in order on the real line.

Array used for the bisection	
endpoints	μ_l and μ_k
1	3.1426
2	9.4248
3	12.5664
4	15.7080
5	21.9911
6	28.2743

TABLE 5. First 6 Values for μ_k and μ_l arranged in order.

We use the code to check for the relative ordering of μ^l and μ_k^c, since it depends whether $c > 1$ or $c < 1$. Therefore the array already contains the proper ordering of the vertical asymptotes. The need for this check becomes apparent after looking at Table 6. With $c = 4$ we see that the first two values of the array are 3.1416 and 9.4248. It is important to note here that both of these values come from μ_k as the first value for μ_l=12.5664.

The bisection method gives an approximate value for $\mu \approx 4.48$. To check convergence, we include in Table 6 a list of computed values for $f(\mu)$ and the difference in the value of μ between iterations. Both becomes smaller and smaller as we iterate. Recall that the true μ solves $f(\mu) = 0$.

As stated before, we know that in each interval between μ_l and μ_k, there is exactly one solution therefore from our list we perform bisection using the endpoints in the array that we had constructed. Using the fact that there exists one solution for any two points in the array we can show how many eigenvalues will lie in any interval.

For instance let the interval be [0,T]. We have that by the definition of the eigenvalue that:

$$\lambda = -\mu_l^2 \in [-T, 0] \longrightarrow \mu \in [-\sqrt{T}, \sqrt{T}] \longrightarrow -\sqrt{T} < (2l+1)\pi < \sqrt{T}. \tag{37}$$

Convergence to $\mu = 4.48$			
iteration i	μ_i	$f(\mu)$	$\|\mu_{i-1}$ - $\mu_i\|$
1	6.2832	-1.4142	-
2	4.7124	-.1978	1.5708
3	3.9270	.4540	.7854
4	4.3197	.1419	.3927
5	4.5160	-.0257	.1963
6	4.4179	.0588	.0981
7	4.4670	.0167	.0497
8	4.4915	-.0045	.0245

Table 6. Iterations to compute the eigenvalue.

We see that each l is then found in the following interval:

$$\frac{-\sqrt{T}}{2\pi} - 1/2 < l < \frac{\sqrt{T}}{2\pi} - 1/2. \tag{38}$$

Taking the difference we can find the total number of nodes come out to be:

$$\frac{-\sqrt{T}}{2\pi} - 1/2 - (\frac{\sqrt{T}}{2\pi} - 1/2) = \frac{\sqrt{T}}{\pi}. \tag{39}$$

A similar result can be done for so that the number of nodes for μ_k in this range comes out to be $\frac{\sqrt{T}}{c\pi}$. We can sum them up the nodes for μ_l and μ_k to find the total number of nodes dividing $[0, T]$ for bisection on (35). Thus we see that, based on the size of the interval, and how many total nodes divide up the interval, we can predict how many eigenvalues will be in that interval. Since we have one solution in every set of points in the interval for n nodes we will have $n - 1$ eigenvalues.

3.3. **Convergence of the Bisection Method.**

Applying the bisection method was very useful for this one-dimensional eigenvalue problem. We were able to show how many eigenvalues there will be in the given interval, and achieving convergence was relatively cheap in regards to computational power. This was as a result of being able to reduce the original problem to a simpler one through theoretical analysis.

For instance, once we find the system of equations based upon (31) (32) and our constraints, we only need to find the determinant of the system. From there we apply the computer code for setting up the array of endpoints to be used in bisection method, from our values of c , l , and k. Convergence of the bisection method was achieved quickly as the algorithm described in Section (3.2) was done on an interval by interval basis. Although this method is fast and useful for the eigenvalue problem, its applications to higher-dimensions is much more limited.

4. Conclusions

There were advantages and disadvantages of using both power method and bisection method. The power method allows us only to find the dominant eigenvalue for our matrix. Where as bisection method could find as many eigenvalues as we wanted depending on the size of our interval.

The main problem with the power method is the convergence rate. Our stiffness matrices had small sizes, namely 7×7, or a slightly larger 20×20. When attempting even a modestly larger matrix, convergence as expected took slightly more time and computational power. In our particular case, the power method was still effective as our matrix was very sparse. For larger problems, the power method may be difficult to implement, unless some preliminary work is done on the matrix itself.

The bisection method makes up for speed by allowing the user to divide the problem into many simpler sub-problems, none of which were computational draining as finding an inverse of a matrix. Therefore the speed of convergence for the one-dimensional eigenvalue problem was clearly in favor of the bisection method. However, the bisection method's usefulness is limited beyond the one-dimensional case, where a finite element style iterative method would be more appropriate. Although both methods proved to work accurately, the bisection method gave the eigenvalues in a fast and efficient way.

References

[1] Lawerence c Evans *Introduction to Partial Differential Equations*, American Mathematical Society, 1998

[2] Mark E. Davis*Numerical Methods and Modeling for Chemical Engineers*, John Wiley and Sons, Canada, 1984

[3] Walter Strauss, *Partial Differential Equations, An Introduction*, Second, John Wiley and Sons, River Street Hoboken, NJ, 2008.

[4] David Kincaid and Ward Cheney, *Numerical Analysis: The Mathematics of Scientific Computing*, Third, Brooks/Cole, Pacfic Groove, CA, 2002.

[5] Barna Szabo and Ivo Babuska, *Finite Element Analysis*, John Wiley and Sons, Canada, 1991.

[6] Klaus-Juergen Bathe and Edward L. Wilson, *Numerical Methods in Finite Element Analysis*, Prentice-Hall, Inc, 1976.