THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF MATHEMATICS

MORE EFFICIENT CHOICES OF LINEAR BASES IN NUMERICAL METHODS

CAROL GAERTNER
Summer 2010

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Architectural Engineering and Mathematics
with honors in Mathematics

Reviewed and approved* by the following:

Victor Nistor
Professor of Mathematics
Thesis Supervisor

Svetlana Katok
Professor of Mathematics
Honors Adviser

* Signatures are on file in the Schreyer Honors College.

Abstract

In this thesis, a comparison is presented between two linear bases as used in numerical methods. Specifically, the mathematical theory of the finite element method, which is a numerical analysis technique that is used to approximately solve differential equations, is developed for the one-dimensional case. The main premise of the finite element method is to subdivide the domain of the given problem into smaller regions or elements, in order that the problem can be reduced to solving a finite linear system of equations. Generally, consideration is given to a basis of piecewise linear functions when using the finite element method. The first presented basis of piecewise linear functions is the basis that is typically used for the finite element method; it consists of shifted and equally scaled triangle functions, where the shifting and scaling depends upon the number of subdivisions. The second basis of piecewise linear functions that is presented is constructed from subsets of the typical basis, as the typical basis is developed for 2^n subdivisions of the domain $[0, 1]$.

The linear system of equations is developed for the one-dimensional case using both the typical and the alternative bases. The system corresponding to the alternative basis is also modified in an effort to create a diagonally dominant matrix. Then, an iterative numerical method is implemented in the programming language C++ to solve both of the linear systems of equations, in order to compare the efficiency with which the given problem can be solved depending on the choice of basis. Specifically, the Jacobi method is used to solve the systems due to its guaranteed convergence for diagonally dominant matrices. The results of the implementation provide support for the use of the alternative basis in numerical methods.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I am grateful to my friends and family, especially my parents, for their unfaltering support. I would also like to thank Van Cyr for his helpful conversations and Sean Quinn Marlow for his software assistance. Finally, I would like to acknowledge my advisor, Professor Victor Nistor, for his patience and guidance.

Chapter 1

Introduction

1.1 The Finite Element Method

The finite element method is a numerical analysis technique that is used to find approximate solutions to differential equations. These differential equations typically model problems that arise in a variety of engineering disciplines. When using the finite element method, the domain in which the problem is defined is subdivided into a finite number of smaller regions or elements. For example, triangles or quadrilaterals may be used for the subdivisions in two dimensions. Therefore, the finite element method tends to be an appropriate choice for solving problems with complex geometries. This discretization process reduces a continuum problem with an infinite number of unknowns into a problem with a finite number of unknowns that can be solved using linear algebra techniques.

The name *finite element method* first appeared in 1960 in a paper by R. W. Clough on plane elasticity problems. However, significant development of the finite element method began to appear approximately two decades earlier in both the fields of applied mathematics and structural engineering. The applied mathematicians were interested in solving boundary value problems of continuum mechanics, while the engineers were being faced with increasingly more complex shell structures. Richard Courant was the first applied mathematician to publish work on the use of piecewise continuous functions defined over triangular elements in 1943. Meanwhile, structural engineers intuitively realized that they could approximate a plate structure as a truss, which is simply an assembly of individual rods with known characteristics. The structural engineer A. Hrenikoff formalized this intuition in 1941 as the *frame-work method*.

With the advent of computing power in the 1950s, the finite element method gained acceptance as a useful and practical numerical technique for obtaining approximate solutions to a variety of continuum problems. The first commercial finite element method software was produced by

Control Data Corporation in 1964, and only linear problems could be solved in a turnaround time of often several days. Since then, the engineers have worked on applying the finite element method to nonstructural problems, such as heat transfer and fluid flow, and the mathematicians have made advances on establishing errors, bounds, and convergence criteria.

In general, applications of the finite element method fall into one of three different categories. The first category encompasses the *equilibrium problems* or time-independent problems, and the majority of applications of the finite element method are considered to be of this type. For example, an equilibrium problem might involve solving for displacements or stresses due to a thermal loading. The second category involves *eigenvalue problems* or steady-state problems that occur in solid and fluid mechanics. For example, a civil engineer might need to solve a problem involving the interactions between water and a dam. Finally, the third category is an extension of the first and second categories. The third category consists of *propagation problems* or time-dependent problems that arise when the dimension of time is added to an equilibrium or eigenvalue problem.

1.2 Objective

The basic mathematical theory of the finite element method will be developed for the one-dimensional case in Chapter 2. First, a weak formulation will be produced for a given one-dimensional boundary value problem, and it will be shown that the weak formulation is equivalent to the original problem. Then, the discretization process will be performed in which a finite dimensional subspace of the original function space is used in the weak formulation in lieu of the original function space. The resulting formulation will yield an approximate solution to the given one-dimensional boundary value problem. It will be shown that the approximate solution arises from solving a finite linear system of equations and is unique. Finally, the linear system of equations will be developed using the basis of the subspace that is typically chosen.

In Chapter 3, an alternative to the typical basis of the finite dimensional subspace will be constructed. It will be shown that the alternative basis is, in fact, a basis of the subspace. Then, the linear system of equations, from which the unique approximate solution to the one-dimensional boundary value problem is obtained, will be developed using the alternative basis. Finally, the linear system of equations will be modified in an attempt to create a diagonally dominant matrix that will allow the system to be more efficiently solved using an iterative numerical method.

Chapter 4 will focus on the choice of iterative numerical method for solving the modified linear system of equations. Specifically, the Jacobi method will be considered, and it will be shown that the Jacobi method converges for diagonally dominant matrices. The implementation of the Jacobi method for solving the linear system developed from the typical basis and the modified linear system developed from the alternative basis in the programming language C++ will be

discussed. Finally, the results of the implementation will provide support for the use of the alternative basis in the one-dimensional case of the finite element method.

Chapter 2

Development of the One-Dimensional Case

2.1 Weak Formulation

Consider the following one-dimensional boundary value problem

$$\begin{cases} -u''(x) = f(x) & \text{in } (0,1) \\ u(0) = u'(1) = 0 \end{cases} \tag{2.1.1}$$

where f is given, u is the solution, and u'' is the second derivative of u with respect to x.

Let $v(x)$ be any function such that $v(0) = 0$, so multiplication by $v(x)$ and integration over $[0,1]$ yield

$$\int_0^1 f(x)\, v(x)\, dx = \int_0^1 -u''(x)\, v(x)\, dx$$

Integration by parts on the right-hand-side produces

$$\int_0^1 -u''(x)\, v(x)\, dx = -u'(x)\, v(x)|_0^1 + \int_0^1 u'(x)\, v'(x)\, dx$$

However, $v(0) = 0$ and $u'(1) = 0$, so

$$u'(x)\, v(x)|_0^1 = u'(1)\, v(1) - u'(0)\, v(0) = 0$$

Therefore,

$$(f, v) := \int_0^1 f(x)\, v(x)\, dx = \int_0^1 u'(x)\, v'(x)\, dx =: B(u, v)$$

Formally, let $V = \left\{v : [0,1] \to \mathbb{R} \mid v(0) = 0, v \in C^2([0,1])\right\}$. Then, the proposed *weak* formulation is

$$(f, v) = B(u, v) \, \forall v \in V \text{where the solution } u \in V \tag{2.1.2}$$

and it must be shown that Equation 2.1.2 is equivalent to Equation 2.1.1.
Note that Equation 2.1.2 may also be referred to as the *variational* formulation because the function $v(x)$ is permitted to vary arbitrarily.

Definition 1. A function $f : X \to Y$ is **continuous** at a point $c \in X$ if $\forall \epsilon > 0 \; \exists \; \delta > 0 \; s.t.$
$\forall x \in X \; |x - c| < \delta \Rightarrow |f(x) - f(c)| < \epsilon$

Lemma 2.1.1. *Let* $g : [a,b] \to \mathbb{R}, g \in C^0([a,b])$.
Then, $g(x) = 0 \iff \int_a^b g(x) h(x) \, dx = 0 \; \forall h : [a,b] \to \mathbb{R}, h \in C^0([a,b])$.

Proof.

$$\int_a^b g(x) h(x) \, dx = 0 \; \forall h : [a,b] \to \mathbb{R}, h \in C^0([a,b]$$

$g : [a,b] \to \mathbb{R}, g \in C^0([a,b])$ so substitution of $g(x)$ for $h(x)$ yields

$$\int_a^b g^2(x) \, dx = 0$$

$g^2(x) \in C^0([a,b])$ since the product of two continuous functions is continuous. Also, $g^2(x) \geq 0$.
Assume $g^2(x)$ is not identically zero over the interval $[a,b]$.
Then, there exists $d \in [a,b]$ s.t. $g^2(d) = 2\epsilon > 0$.
From the continuity of $g^2(x)$, there is an interval $(d - \delta, d + \delta)$ where

$$\forall x \in (d - \delta, d + \delta) \Rightarrow |g^2(x) - 2\epsilon| < \epsilon$$

$$\forall x \in (d - \delta, d + \delta) \Rightarrow \epsilon < g^2(x) < 3\epsilon$$

In other words, if $g^2(x) > 0$ at some point $d \in [a,b]$, then $g^2(x) > 0$ for an interval around $x = d$.
Also, recall that $g^2(x) \geq 0$. Therefore, by the definition of an integral

$$\int_a^b g^2(x) \, dx > 0$$

However, this is a contradiction with $\int_a^b g^2(x) \, dx = 0$. Therefore, $g^2(x) = 0$, and it follows that $g(x) = 0$.
Finally, it is clear that $g(x) = 0 \Rightarrow \int_a^b g(x) h(x) \, dx = \int_a^b 0 \cdot h(x) \, dx = \int_a^b 0 dx = 0$. □

Theorem 2.1.2. *Assume* $u \in V$ *and* $f \in C^0([0,1])$ *satisfy Equation* 2.1.2.
Then, u *solves Equation* 2.1.1.

Proof. By definition,

$$\int_0^1 u'(x) v'(x) \, dx = B(u, v)$$

Integration by parts on the left-hand-side produces

$$u'(x)\,v(x)|_0^1 - \int_0^1 v(x)\,u''(x)\,dx = B(u,v)$$

Impose the condition that $v(1) = 0$. Furthermore, $v \in V$, so $v(0) = 0$. Then,

$$u'(x)\,v(x)|_0^1 = u'(1)\,v(1) - u'(0)\,v(0) = 0$$

Combining the above result with Equation 2.1.2 yields

$$-\int_0^1 v(x)\,u''(x)\,dx = B(u,v) = (f,v)$$

By definition,

$$(f,v) = \int_0^1 f(x)\,v(x)\,dx$$

Therefore,

$$-\int_0^1 v(x)\,u''(x)\,dx = \int_0^1 f(x)\,v(x)\,dx$$

$$\int_0^1 f(x)\,v(x)\,dx + \int_0^1 v(x)\,u''(x)\,dx = 0$$

$$\int_0^1 [f(x)\,v(x) + v(x)\,u''(x)]\,dx = 0$$

$$\int_0^1 v(x)\,[f(x) + u''(x)]dx = 0$$

$f(x)$ and $u''(x)$ are both continuous, so their sum $f(x) + u''(x)$ is continuous. $v(x)$ is also continuous. Therefore, according to Lemma 2.1.1,

$$f(x) + u''(x) = 0$$

$$f(x) = -u''(x)$$

Now, return to $u'(x)\,v(x)|_0^1 - \int_0^1 v(x)\,u''(x)\,dx = B(u,v) = (f,v)$ with $v(x) = x$.
Elimination of the $u'(0)\,v(0)$ term on the left-hand-side since $v(0) = 0$ and substitution of the definition of (f,v) on the right-hand-side yield

$$u'(1)\,v(1) - \int_0^1 v(x)\,u''(x)\,dx = \int_0^1 f(x)\,v(x)\,dx$$

It has already been shown that $f(x) = -u''(x)$, so

$$u'(1)\,v(1) + \int_0^1 v(x)\,f(x)\,dx = \int_0^1 f(x)\,v(x)\,dx$$

Therefore,

$$u'(1)\,v(1) = 0$$

$$u'(1) = 0$$

Finally, it is obvious that $u(0) = 0$ since $u \in V$. □

Terminology. $u(0) = 0$ is called an *essential* boundary condition because it appears directly in the definition of V. $u'(1) = 0$ is called a *natural* boundary condition because it is included implicitly. Generally, the boundary condition $u(x) = 0$ is referred to as *Dirichlet*, and the boundary condition $u'(x) = 0$ is referred to as *Neumann*.

2.2 Alternative Boundary Conditions

Now, consider the following one-dimensional boundary value problem with slightly different boundary conditions than Equation 2.1.1.

$$\begin{cases} -u''(x) = f(x) & \text{in } (0,1) \\ u(0) = u(1) = 0 \end{cases} \tag{2.2.1}$$

Revise the function space V so that $V = \left\{v : [0,1] \to \mathbb{R} \mid v(0) = v(1) = 0, v \in C^2([0,1])\right\}$.
Then, the weak formulation expressed in Equation 2.1.2 for Equation 2.1.1 can be derived in a similar manner for Equation 2.2.1 with its alternative boundary conditions and revised function space V. The natural boundary condition $u'(1) = 0$ is no longer necessary because now the solution u exists in the function space V where $u(x)$ must equal 0 when $x = 1$. Notice that $u(1) = 0$ is an essential boundary condition of the Dirichlet type.

Equation 2.2.1 and its corresponding revised function space V will be henceforth considered in lieu of Equation 2.1.1 for the purposes of implementation.

2.3 Discretization

Consider the weak formulation expressed in Equation 2.1.2, where the function space V is replaced by a finite dimensional subspace $S \subset V$.

$$(f, v_s) = B(u_s, v_s)\,\forall v_s \in S \text{ where the solution } u_s \in S \tag{2.3.1}$$

The solution u_s in Equation 2.3.1 approximates the exact solution u in Equation 2.2.1. Increasing the dimension of the finite dimensional subspace S results in a better approximation.
It will be shown next that solving Equation 2.3.1 for u_s is equivalent to solving a system of equations.

Definition 2. A **basis** $\{e_1, e_2, ..., e_n\}$ of a function space G is a linearly independent subset of G that spans G.

Linear Independence Property:
$\sum_{j=1}^{n} c_j e_j(x) = 0 \Rightarrow c_j = 0$, where c_j are scalars.

Spanning Property:
Any $g \in G$ can be expressed as $g(x) = \sum_{j=1}^{n} c_j e_j(x)$.

Lemma 2.3.1. *Let* $\{\varphi_1, \varphi_2, ..., \varphi_n\}$ *be a basis of* S.
Then, Equation 2.3.1 $\Leftrightarrow (f, \varphi_i) = B(u_s, \varphi_i) \forall i = 1, ..., n$

Proof. From the definition of a basis, each $\varphi_i \in S$, so it is clearly true that

$$(f, v_s) = B(u_s, v_s) \forall v_s \in S \Rightarrow (f, \varphi_i) = B(u_s, \varphi_i) \forall i = 1, ..., n$$

Next, show that $(f, \varphi_i) = B(u_s, \varphi_i) \forall i = 1, ..., n \Rightarrow (f, v_s) = B(u_s, v_s) \forall v_s \in S$

$$\int_0^1 f(x) \varphi_i(x) dx = \int_0^1 u_s'(x) \varphi_i'(x) dx \ \forall i = 1, ..., n$$

$$c_i \int_0^1 f(x) \varphi_i(x) dx = c_i \int_0^1 u_s'(x) \varphi_i'(x) dx \ \forall i = 1, ..., n$$

$$\sum_{i=1}^{n} c_i \int_0^1 f(x) \varphi_i(x) dx = \sum_{i=1}^{n} c_i \int_0^1 u_s'(x) \varphi_i'(x) dx$$

$$\int_0^1 \sum_{i=1}^{n} c_i f(x) \varphi_i(x) dx = \int_0^1 \sum_{i=1}^{n} c_i u_s'(x) \varphi_i'(x) dx$$

$$\int_0^1 \left[\sum_{i=1}^{n} c_i \varphi_i(x)\right] f(x) dx = \int_0^1 \left[\sum_{i=1}^{n} c_i \varphi_i'(x)\right] u_s'(x) dx$$

From the definition of a basis, $v_s(x) = \sum_{i=1}^{n} c_i \varphi_i(x)$, so subsitution yields

$$\int_0^1 v_s(x) f(x) dx = \int_0^1 v_s'(x) u_s'(x) dx$$

$$(f, v_s) = B(u_s, v_s) \forall v_s \in S$$

□

Theorem 2.3.2. *Solving Equation 2.3.1 is equivalent to solving a square system of equations and results in a unique solution.*

Proof. From Lemma 2.3.1, it is known that Equation 2.3.1 $\Leftrightarrow (f, \varphi_i) = B(u_s, \varphi_i) \forall i = 1, ..., n$ where $\{\varphi_1, \varphi_2, ..., \varphi_n\}$ is a basis of S.

$$\int_0^1 f(x) \varphi_i(x) dx = \int_0^1 u_s'(x) \varphi_i'(x) dx \ \forall i = 1, ..., n$$

From the definition of a basis, $u_s(x) = \sum_{j=1}^{n} c_j \varphi_j(x)$. Therefore, substitution for $u_s'(x)$ yields

$$\int_0^1 f(x)\,\varphi_i(x)\,dx = \int_0^1 \left[\sum_{j=1}^{n} c_j \varphi_j'(x)\right] \varphi_i'(x)\,dx \;\forall i = 1, ..., n$$

$$\int_0^1 f(x)\,\varphi_i(x)\,dx = \sum_{j=1}^{n} c_j \int_0^1 \varphi_j'(x)\,\varphi_i'(x)\,dx \;\forall i = 1, ..., n$$

which can be rewritten as the following to more clearly indicate the system of equations:

$$\int_0^1 f(x)\,\varphi_1(x)\,dx = c_1 \int_0^1 \varphi_1'(x)\,\varphi_1'(x)\,dx + c_2 \int_0^1 \varphi_2'(x)\,\varphi_1'(x)\,dx + ... + c_n \int_0^1 \varphi_n'(x)\,\varphi_1'(x)\,dx$$

$$\int_0^1 f(x)\,\varphi_2(x)\,dx = c_1 \int_0^1 \varphi_1'(x)\,\varphi_2'(x)\,dx + c_2 \int_0^1 \varphi_2'(x)\,\varphi_2'(x)\,dx + ... + c_n \int_0^1 \varphi_n'(x)\,\varphi_2'(x)\,dx$$

$$.....$$

$$\int_0^1 f(x)\,\varphi_n(x)\,dx = c_1 \int_0^1 \varphi_1'(x)\,\varphi_n'(x)\,dx + c_2 \int_0^1 \varphi_2'(x)\,\varphi_n'(x)\,dx + ... + c_n \int_0^1 \varphi_n'(x)\,\varphi_n'(x)\,dx$$

Finally, the system can be written in matrix form.

$$\mathbf{A}=[a_{ij}] = [B(\varphi_i, \varphi_j)] = \begin{bmatrix} \int_0^1 \varphi_1'(x)\,\varphi_1'(x)\,dx & \int_0^1 \varphi_1'(x)\,\varphi_2'(x)\,dx & \cdots & \int_0^1 \varphi_1'(x)\,\varphi_n'(x)\,dx \\ \int_0^1 \varphi_2'(x)\,\varphi_1'(x)\,dx & \int_0^1 \varphi_2'(x)\,\varphi_2'(x)\,dx & \cdots & \int_0^1 \varphi_2'(x)\,\varphi_n'(x)\,dx \\ \vdots & \vdots & \ddots & \vdots \\ \int_0^1 \varphi_n'(x)\,\varphi_1'(x)\,dx & \int_0^1 \varphi_n'(x)\,\varphi_2'(x)\,dx & \cdots & \int_0^1 \varphi_n'(x)\,\varphi_n'(x)\,dx \end{bmatrix}$$

$$\mathbf{c} = [c_i] = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \quad , \quad \mathbf{b} = [b_i] = [(f, \varphi_i)] = \begin{bmatrix} \int_0^1 f(x)\,\varphi_1(x)\,dx \\ \int_0^1 f(x)\,\varphi_2(x)\,dx \\ \vdots \\ \int_0^1 f(x)\,\varphi_n(x)\,dx \end{bmatrix}$$

$$\mathbf{Ac} = \mathbf{b} \tag{2.3.2}$$

It is known that an n-by-n (square) linear system of equations, such as Equation 2.3.2, has a unique solution if the matrix $\mathbf{A}$ is invertible. Furthermore, $\mathbf{A}$ is invertible $\Leftrightarrow \mathbf{Ac} = \mathbf{0}$ has only the trivial solution $\mathbf{c} = \mathbf{0}$. Therefore, it is sufficient to prove that the vector $\mathbf{d}$ must equal $\mathbf{0}$ if $\mathbf{Ad}=\mathbf{0}$. The entries of vector $\mathbf{d}$ are obtained from the fact that every $v_s \in S$ can be written as $v_s(x) = \sum_{j=1}^{n} d_j \varphi_j(x)$. If $\mathbf{b}=\mathbf{0}$,

$$(f, \varphi_j) = 0 \;\forall j = 1, ..., n$$

$$0 = B(v_s, \varphi_j) = \int_0^1 v_s'(x)\,\varphi_j'(x)\,dx \;\forall j = 1, ..., n$$

Multiplying by d_j and summing over j yields

$$\sum_{j=1}^{n} d_j \int_0^1 v_s'(x)\,\varphi_j'(x)\,dx = 0$$

$$\int_0^1 \sum_{j=1}^{n} d_j v_s'(x)\,\varphi_j'(x)\,dx = 0$$

$$\int_0^1 \left[\sum_{j=1}^{n} d_j \varphi_j'(x)\right] v_s'(x)\,dx = 0$$

Substitution of $v_s(x) = \sum_{j=1}^{n} d_j \varphi_j'(x)$ yields

$$\int_0^1 [v_s'(x)]^2\,dx = 0$$

$v_s \in C^2([0,1])$, so v_s' is continuous and Lemma 2.1.1 applies.

$$v_s'(x) = 0 \Rightarrow v_s(x) = constant$$

It is also known that $v_s(0) = 0$, so $v_s(x) = 0$.
From the definition of a basis, specifically the linear independence property, if $v_s(x) = 0$, then

$$\sum_{j=1}^{n} d_j \varphi_j(x) = 0 \Rightarrow d_j = 0 \;\forall j = 1, ..., n \Rightarrow \mathbf{d} = \mathbf{0}$$

□

Terminology. The matrix $\mathbf{A}$ is commonly referred to as the *stiffness* matrix, due to its use in solving structural problems.

2.4 Typical Choice of Basis

Partition the interval $[0,1]$ so that $0 = x_0 < x_1 < \ldots < x_{n-1} < x_n = 1$. Then, the subspace $S_n \subset V$ can be defined as the space of piecewise polynomial functions $v_s : [0,1] \to \mathbb{R}$ such that

- $v_s \in C^0([0,1])$
- $v_{s[x_{i-1},x_i]}$ is linear for $i = 1, ..., n$
- $v_s(0) = v_s(1) = 0$

Let $\{\varphi_i(x) : i = 1, ..., n-1\}$ be a basis of S_n where

$$\varphi_i(x) = \begin{cases} nx - i + 1 & \text{if } x \in \left(\frac{i-1}{n}, \frac{i}{n}\right] \\ -nx + i + 1 & \text{if } x \in \left[\frac{i}{n}, \frac{i+1}{n}\right) \\ 0 & \text{otherwise} \end{cases} \tag{2.4.1}$$

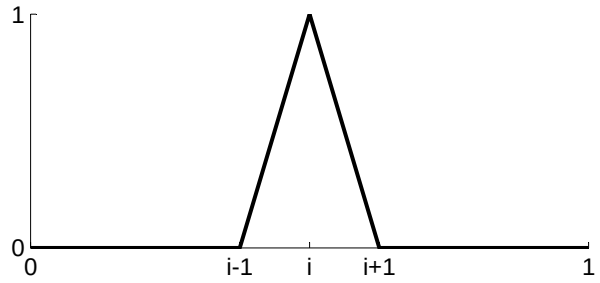

Figure 2.1. Typical Basis Function $\varphi_i(x)$

Terminology. The points $\{x_i\}$ are called the *nodes*, and $\{\varphi_i(x) : i = 1, ..., n-1\}$ is called a *nodal basis*.

Theorem 2.4.1. *$\{\varphi_i(x) : i = 1, ..., n-1\}$ as defined in Equation 2.4.1 is a basis for S_n.*

Proof. First, check the linear independence property:
If $\sum_{i=1}^{n-1} c_i\varphi_i(x) = 0$, then

$$0 + ... + 0 + c_i(nx - i + 1) + 0... + 0 = 0 \text{ for } x \in \left(\frac{i-1}{n}, \frac{i}{n}\right]$$

Therefore, either $c_i = 0$ or $x = \frac{i-1}{n}$.
However, $x = \frac{i-1}{n}$ is not included in the interval $\left(\frac{i-1}{n}, \frac{i}{n}\right]$, so all c_i must equal zero.

Next, check the spanning property:
Let $v_s(x) \in S_n$.

$$\begin{gathered} v_s(0) = 0 = \sum_{i=1}^{n-1} c_i\varphi_i(0) = 0 + ... + 0 \\ v_s(x_1) = c_1 = \sum_{i=1}^{n-1} c_i\varphi_i(x_1) = c_1(1) + 0 + ... + 0 \\ v_s(x_2) = c_2 = \sum_{i=1}^{n-1} c_i\varphi_i(x_2) = 0 + c_2(1) + 0 + ... + 0 \\ \vdots \\ v_s(x_{n-1}) = c_{n-1} = \sum_{i=1}^{n-1} c_i\varphi_i(x_{n-1}) = 0 + ... + 0 + c_{n-1}(1) \\ v_s(1) = 0 = \sum_{i=1}^{n-1} c_i\varphi_i(1) = 0 + ... + 0 \end{gathered}$$

Therefore, $v_s(x) = \sum_{i=0}^{n} c_i \varphi_i(x)$ because both are linear on each interval $[x_{i-1}, x_i]$ by definition and the endpoints are equal as shown above. □

Figure 2.2 shows an example of a function $v_s(x) \in S_8$ that is generated by a linear combination of the basis functions $\{\varphi_i(x) : i = 1, ..., 7\}$ of S_8.

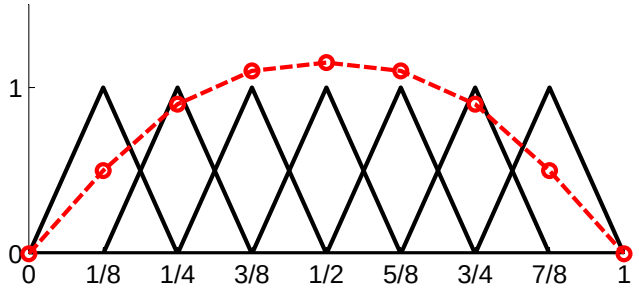

Figure 2.2. Typical Basis for the Space S_8 with Superimposed $v_s(x) \in S_8$

Definition 3. The **dimension** of a vector space S is equal to the number of vectors (in this context, functions) that constitute a basis of S.

Therefore, for the example provided in Figure 2.2, $dim(S_8) = 7$. In general, $dim(S_n) = n-1$.

Construct the stiffness matrix $\mathbf{A}$ using the basis $\{\varphi_i(x) : i = 1, ..., n-1\}$ defined in Equation 2.4.1. Recall that

$$a_{ij} = \int_0^1 \varphi_i'(x)\, \varphi_j'(x)\, dx$$

Next, the derivative of the basis function can be computed.

$$\varphi_i'(x) = \begin{cases} n & \text{for } \left(\frac{i-1}{n}, \frac{i}{n}\right] \\ -n & \text{for } \left[\frac{i}{n}, \frac{i+1}{n}\right) \\ 0 & \text{otherwise} \end{cases}$$

The resulting stiffness matrix $\mathbf{A}$ is tridiagonal with its entries defined as

$$a_{ii} = \int_{\frac{i-1}{n}}^{\frac{i+1}{n}} n^2 dx \qquad \text{(i.e. main diagonal entries)}$$

$$a_{ij} = \int_{\frac{i}{n}}^{\frac{j}{n}} -n^2 dx \quad \text{if } j = i+1 \text{ (i.e. superdiagonal entries)}$$

$$a_{ij} = \int_{\frac{j}{n}}^{\frac{i}{n}} -n^2 dx \quad \text{if } j = i-1 \text{ (i.e. subdiagonal entries)}$$

Then, the stiffness matrix $\mathbf{A}$ can be simplified to

$$\begin{bmatrix} 2n & -n & 0 & 0 & \dots & 0 \\ -n & 2n & -n & 0 & \dots & 0 \\ 0 & -n & 2n & -n & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -n & 2n & -n \\ 0 & \dots & 0 & 0 & -n & 2n \end{bmatrix}$$

Note that a more general form of Equation 2.2.1 can be considered for the one-dimensional boundary value problem. For example,

$$\begin{cases} -\Big(a\left(x\right)u'\left(x\right)\Big)' = f\left(x\right) \quad \text{in } (0,1) \\ u\left(0\right) = u\left(1\right) = 0 \end{cases} \tag{2.4.2}$$

where $a \in C^0\left([0,1]\right)$.

The tridiagonal stiffness matrix $\mathbf{A}$ corresponding to Equation 2.4.2 has the following entries:

$$a_{ii} = n^2 \textstyle\int_{\frac{i-1}{n}}^{\frac{i+1}{n}} a\left(x\right) dx \qquad \text{(i.e. main diagonal entries)}$$

$$a_{ij} = -n^2 \textstyle\int_{\frac{i}{n}}^{\frac{j}{n}} a\left(x\right) dx \quad \text{if } j = i+1 \text{ (i.e. superdiagonal entries)}$$

$$a_{ij} = -n^2 \textstyle\int_{\frac{j}{n}}^{\frac{i}{n}} a\left(x\right) dx \quad \text{if } j = i-1 \text{ (i.e. subdiagonal entries)}$$

Recall the entries of $\mathbf{b}$.

$$b_i = \int_0^1 f\left(x\right)\varphi_i\left(x\right) dx$$

Using the basis $\{\varphi_i\left(x\right) : i = 1, ..., n-1\}$ defined in Equation 2.4.1, the entries of $\mathbf{b}$ corresponding to both Equation 2.2.1 and Equation 2.4.2 become

$$b_i = \int_{\frac{i-1}{n}}^{\frac{i}{n}} f\left(x\right)\left(nx - i + 1\right) dx + \int_{\frac{i}{n}}^{\frac{i+1}{n}} f\left(x\right)\left(-nx + i + 1\right) dx$$

Chapter 3

Alternative Choice of Basis

3.1 Construction

Consider the one-dimensional boundary value problem in Equation 2.2.1. An alternative to the typical basis defined in Equation 2.4.1 can be constructed for the space S_{2^m} using a selection of the typical basis functions that correspond to the subspaces S_{2^n} where $n \leq m$. Specifically, let $\varphi_k^n(x)$ correspond to the typical basis function that exists in the space S_{2^n} such that $\varphi_k^n\left(\frac{k}{2^n}\right) = 1$. Then, all $\varphi_k^n(x)$ where k is odd and $n \leq m$ comprise the alternative basis for the space S_{2^m}.

The alternative basis for the space S_8, where $m = 3$, will be constructed as an example.

Figure 3.1 shows the typical basis for the subspace S_2, which is the subspace S_{2^n} such that $n = 1 < m$. The typical basis consists of the function $\varphi_1^1(x)$.

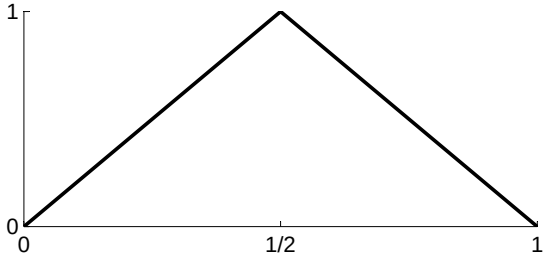

Figure 3.1. Typical Basis for the Space S_2

Figure 3.2 shows the typical basis for the subspace S_4, which is the subspace S_{2^n} such that $n = 2 < m$. The typical basis consists of the functions $\varphi_1^2(x), \varphi_2^2(x), \varphi_3^2(x)$. The basis functions with odd k are differentiated from those with even k in the figure.

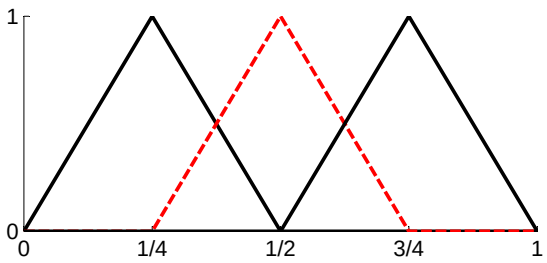

Figure 3.2. Typical Basis for the Space S_4

Figure 3.3 shows the typical basis for the subspace S_8, which is the subspace S_{2^n} such that $n = 3 = m$. The typical basis consists of the functions $\left\{\varphi_k^3(x) : k = 1, ..., 7\right\}$. The basis functions with odd k are differentiated from those with even k in the figure.

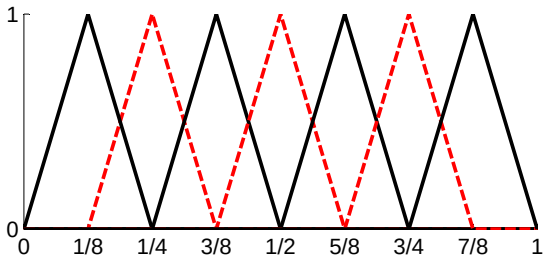

Figure 3.3. Typical Basis for the Space S_8

Figure 3.4 shows the alternative basis for the space S_8 constructed from the typical basis functions with odd k depicted in Figures 3.1, 3.2, and 3.3. Specifically, the alternative basis that has been constructed for the space S_8 is $\{\varphi_1 \in S_2\ ;\ \varphi_1, \varphi_3 \in S_4\ ;\ \varphi_1, \varphi_3, \varphi_5, \varphi_7 \in S_8\}$.

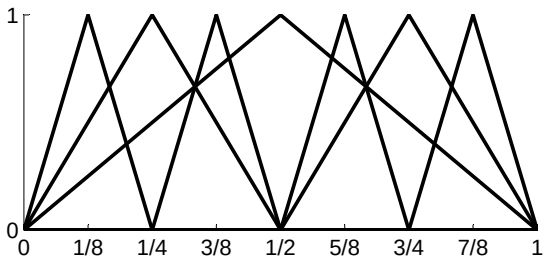

Figure 3.4. Alternative Basis for the Space S_8

Let $\varphi_k^n(x) = \psi_i(x)$ be a new notation for the alternative basis. For example, the alternative basis that was constructed for the space S_8 can be expressed as

$$\begin{aligned}
\varphi_1^1 &= \psi_1 \\
\varphi_1^2 &= \psi_2 \\
\varphi_3^2 &= \psi_3 \\
\varphi_1^3 &= \psi_4 \\
\varphi_3^3 &= \psi_5 \\
\varphi_5^3 &= \psi_6 \\
\varphi_7^3 &= \psi_7
\end{aligned}$$

Lemma 3.1.1. (*Linear Dependence Lemma*) *Let V be a finite dimensional vector space. If the set of elements $\{v_1, ..., v_n\}$ is linearly dependent in V and $v_1 \neq 0$, then there exists $i \in \{1, 2, ..., n\}$ such that*

1. $v_i \in span\{v_1, ..., v_{i-1}\}$
2. $span\{v_1, ..., v_{i-1}, v_{i+1}, ..., v_n\} = span\{v_1, ..., v_n\}$

Proof. Since $\{v_1, ..., v_n\}$ are linearly dependent, there exist scalars $c_1, ..., c_n$ not all zero such that

$$c_1 v_1 + ... + c_n v_n = 0$$

Since $v_1 \neq 0$, not all $c_2, ..., c_n$ can equal zero. Let $i \in \{2, ..., n\}$ be the largest such that $c_i = 0$. Then,

$$v_i = -\frac{c_1}{c_i} v_1 - ... - \frac{c_{i-1}}{c_i} v_{i-1} \Rightarrow Part\ 1$$

Next, let $w \in span\{v_1, ..., v_n\}$. Then, there exist scalars $d_1, ..., d_n$ such that

$$w = d_1 v_1 + ... + d_n v_n$$

Substitute v_i from Part 1 so that w is expressed as a linear combination of $\{v_1, ..., v_{i-1}, v_{i+1}, ..., v_n\}$. In other words, $w \in span\{v_1, ..., v_{i-1}, v_{i+1}, ..., v_n\}$, which implies Part 2. □

Theorem 3.1.2. (*Basis Extension Theorem*) *Every linearly independent set of elements $\{v_1, ..., v_n\}$ in a finite dimensional vector space V can be extended to a basis of V.*

Proof. Let $B = \{v_1, ..., v_n\}$. Let $\{u_1, ..., u_m\}$ be a set of elements in V that span V.

Step 1: If u_1 is in the span of B, throw u_1 away. If u_1 is not in the span of B, let $B = \{v_1, ..., v_n, u_1\}$ so that u_1 is in the span of B. B must still be linearly independent because, otherwise, there would be a contradiction with Lemma 3.1.1.

Step i: If u_i is in the spane of B, throw u_i away. If u_i is not in the span of B, add it to the

end of B. Now, $u_1, ..., u_i$ are in the span of B, and B is still linearly independent by Lemma 3.1.1.

After m steps, the span of B will include $u_1, ..., u_m$, so B spans V. Furthermore, B is still linearly independent. Therefore, B is a basis of V. □

Corollary 3.1.3. *Let V be a finite dimensional vector space such that $dim\,(V) = n$. Then, a set of n linearly independent elements in V are a basis of V.*

Proof. Let $\{v_1, ..., v_n\}$ be a linearly independent set of elements in V. Then, according to Theorem 3.1.2, $\{v_1, ..., v_n\}$ can be extended to a basis of V. However, a result of Definition 3 is that every basis of a vector space V consists of n elements where $dim\,(V) = n$. Therefore, no element needs to be added to $\{v_1, ..., v_n\}$. In other words, $\{v_1, ..., v_n\}$ is already a basis of V. □

Theorem 3.1.4. *The functions $\{\psi_i\,(x) : i = 1, 2, ..., 2^m - 1\}$ form a basis of the space S_{2^m}.*

Proof. It is known that $dim\,(S_{2^m}) = 2^m - 1$. Therefore, by Corollary 3.1.3, it is sufficient to show that the functions $\{\psi_i\,(x) : i = 1, 2, ..., 2^m - 1\}$ are linearly independent in order to prove that they form a basis of S_{2^m}.

Recall from Definition 2 that the linear independence property holds if

$$\sum_{i=1}^{2^m-1} c_i\psi_i\,(x) = 0 \Rightarrow c_i = 0$$

Evaluate the above linear combination at each node x_i as i increases from 1 to $2^m - 1$.

Step 1: Evaluate at $x_1 = \frac{1}{2}$.

$$\sum_{i=1}^{2^m-1} c_i\psi_i\left(\frac{1}{2}\right) = 0$$

$$c_1\psi_1\left(\frac{1}{2}\right) + c_2\psi_2\left(\frac{1}{2}\right) + ... + c_{2^m-1}\psi_{2^m-1}\left(\frac{1}{2}\right) = 0$$

However, based on the construction of the alternative basis, it is known that

$$\psi_2\left(\frac{1}{2}\right) = ... = \psi_{2^m-1}\left(\frac{1}{2}\right) = 0$$

Therefore,

$$c_1\psi_1\left(\frac{1}{2}\right) = 0$$

$$c_1\,(1) = 0 \Rightarrow c_1 = 0$$

Step 2: Evaluate at $x_2 = \frac{1}{4}$.

$$\sum_{i=1}^{2^m-1} c_i\psi_i\left(\frac{1}{4}\right) = 0$$

$$c_1\psi_1\left(\frac{1}{4}\right)+c_2\psi_2\left(\frac{1}{4}\right)+c_3\psi_3\left(\frac{1}{4}\right)+...+c_{2^m-1}\psi_{2^m-1}\left(\frac{1}{2}\right)=0$$

However, it is known from the previous step that $c_1 = 0$ and from the construction of the alternative basis that

$$\psi_3\left(\frac{1}{4}\right)=...=\psi_{2^m-1}\left(\frac{1}{4}\right)=0$$

Therefore,

$$c_2\psi_2\left(\frac{1}{4}\right)=0$$

$$c_2(1)=0\Rightarrow c_2=0$$

Step i: Evaluate at x_i.

$$\sum_{i=1}^{2^m-1}c_i\psi_i(x_i)=0$$

$$c_1\psi_1(x_i)+...+c_{i-1}\psi_{i-1}(x_i)+c_i\psi_i(x_i)+c_{i+1}\psi_{i+1}(x_i)+...+c_{2^m-1}\psi_{2^m-1}(x_i)=0$$

However, it is known from the previous $i-1$ steps that $c_1 = ... = c_{i-1} = 0$ and from the construction of the alternative basis that

$$\psi_{i+1}(x_i)=...=\psi_{2^m-1}(x_i)=0$$

Therefore,

$$c_i\psi_i(x_i)=0$$

$$c_i(1)=0\Rightarrow c_i=0$$

After 2^m-1 steps, it will be clear that all c_i for $i = 1, ..., 2^m-1$ equal zero. Therefore, the functions $\{\psi_i(x) : i = 1, 2, ..., 2^m-1\}$ are linearly independent, and they form a basis of S_{2^m} $\square$

3.2 Development of the Linear System

The entries of the stiffness matrix $\mathbf{A}$ corresponding to Equation 2.4.2 can be computed using the alternative basis $\{\psi_i(x) : i = 1, 2, ..., 2^m-1\}$.

$$a_{ij}=\int_0^1\psi_i'(x)\,\psi_j'(x)\,a(x)\,dx$$

It is best to work with the original notation for the alternative basis functions when simplifying the entries of $\mathbf{A}$, so a simple conversion between the indices of the different notations is necessary. Let $\{\psi_i(x)=\varphi_k^n(x) : i = 1, 2, ..., 2^m-1\}$ be basis of S_{2^m}. Then,

$$n=\left\lceil\frac{ln(i+1)}{ln2}\right\rceil$$

$$k = 2\left(i - 2^{n-1}\right) + 1$$

This ability to easily convert between the different notations allows the alternative basis functions to be more concisely defined, in a manner similar to Equation 2.4.1 for the typical basis functions.

$$\psi_i(x) = \varphi_k^n(x) = \begin{cases} 2^n x - k + 1 & \text{if } x \in (x_{i_0}, x_i] \\ -2^n x + k + 1 & \text{if } x \in \left[x_i, x_{i_f}\right) \\ 0 & \text{otherwise} \end{cases} \tag{3.2.1}$$

$$\text{where } x_{i_0} = \frac{k-1}{2^n}, \quad x_i = \frac{k}{2^n}, \quad x_{i_f} = \frac{k+1}{2^n}$$

Then, the derivative of the alternative basis function can be computed in terms of n.

$$\psi_i'(x) = \varphi_k^{n\prime}(x) = \begin{cases} 2^n & \text{for } (x_{i_0}, x_i] \\ -2^n & \text{for } \left[x_i, x_{i_f}\right) \\ 0 & \text{otherwise} \end{cases}$$

When simplifying the entries of the stiffness matrix $\mathbf{A}$ using the alternative basis defined in Equation 3.2.1, the result is not a tridiagonal matrix. Rather, several cases must be analyzed.

Consider ψ_i and ψ_j with corresponding n_i and n_j.

First, let $i = j$. Then, $n_i = n_j$, and

$$a_{ii} = 2^{2n_i} \int_{x_{i_0}}^{x_{i_f}} a(x)\, dx \quad \text{(i.e. main diagonal entries)}$$

Second, let $i \neq j$, and assume that $i > j$. Then, either $n_i = n_j$ or $n_i > n_j$. If $n_i = n_j$, then there is no overlap between the intervals $\left(x_{i_0}, x_{i_f}\right)$ and $\left(x_{j_0}, x_{j_f}\right)$. If $n_i > n_j$, then the interval $\left(x_{i_0}, x_{i_f}\right)$ is either entirely contained in $\left(x_{j_0}, x_{j_f}\right)$ or there is no overlap between the two intervals. Furthermore, if $\left(x_{i_0}, x_{i_f}\right) \subset \left(x_{j_0}, x_{j_f}\right)$, then either $\left(x_{i_0}, x_{i_f}\right) \subseteq (x_{j_0}, x_j)$ or $\left(x_{i_0}, x_{i_f}\right) \subseteq \left(x_j, x_{j_f}\right)$. These observations lead to the following entries of $\mathbf{A}$:

- If $n_i > n_j$, $x_{i_0} \geq x_{j_0}$, and $x_{i_f} \leq x_j$, then

$$a_{ij} = 2^{n_i} 2^{n_j} \left[\int_{x_{i_0}}^{x_i} a(x)\, dx - \int_{x_i}^{x_{i_f}} a(x)\, dx \right]$$

  For example, ψ_1 and ψ_5 would result in the above case. See Figure 3.5, and notice that $n_i = 3 > 1 = n_j$, $x_{i_0} = 1/4 \geq 0 = x_{j_0}$, and $x_{i_f} = 1/2 \leq 1/2 = x_j$.

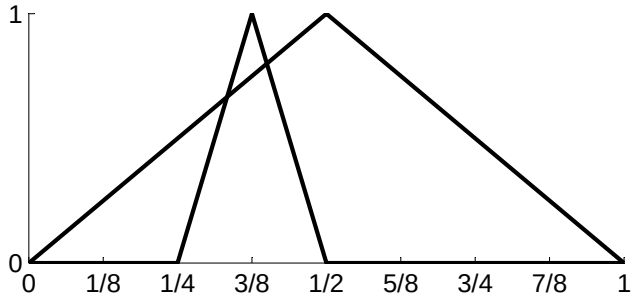

Figure 3.5. Alternative Basis Functions $\psi_1(x)$ and $\psi_5(x)$

- If $n_i > n_j$, $x_{i_0} \geq x_j$, and $x_{i_f} \leq x_{j_f}$, then

$$a_{ij} = 2^{n_i} 2^{n_j} \left[-\int_{x_{i_0}}^{x_i} a(x)\, dx + \int_{x_i}^{x_{i_f}} a(x)\, dx \right]$$

  For example, ψ_1 and ψ_3 would result in the above case. See Figure 3.6, and notice that $n_i = 2 > 1 = n_j$, $x_{i_0} = 1/2 \geq 1/2 = x_j$, and $x_{i_f} = 1 \leq 1 = x_j$.

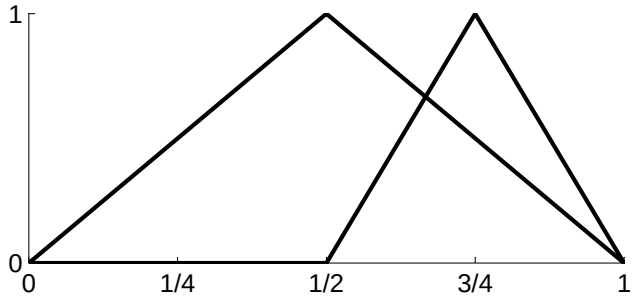

Figure 3.6. Alternative Basis Functions $\psi_1(x)$ and $\psi_3(x)$

- If $n_i = n_j$; or if $n_i > n_j$ and $x_{i_f} \leq x_{j_0}$; or if $n_i > n_j$ and $x_{i_0} \geq x_{j_f}$, then

$$a_{ij} = 0$$

  For example, ψ_4 and ψ_5 would result in the above case. See Figure 3.7, and notice that $n_i = 3 = n_j$.

  Another example of the above case is the alternative basis functions ψ_3 and ψ_4. See Figure 3.8, and notice that $n_i = 3 > 2 = n_j$, $x_{i_f} = 1/4 \leq 1/2 = x_{j_0}$,

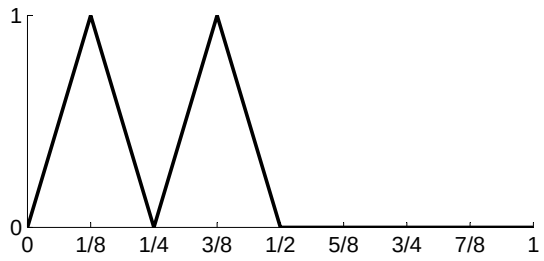

Figure 3.7. Alternative Basis Functions $\psi_4(x)$ and $\psi_5(x)$

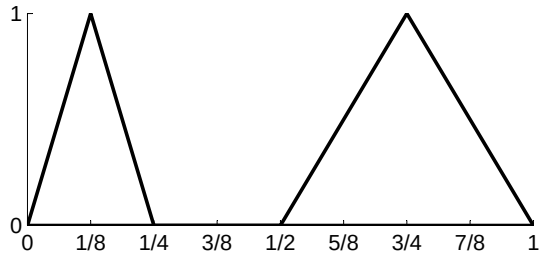

Figure 3.8. Alternative Basis Functions $\psi_3(x)$ and $\psi_4(x)$

The stiffness matrix $\mathbf{A}$ for the space S_8 will be computed using the alternative basis as an example, and the polynomial $a(x) = x^2 + 1$ will be considered in the calculations. The following sample calculations of entries in $\mathbf{A}$ illustrate the various cases.

$$a_{11} = 2^{2(1)} \int_0^1 \left(x^2 + 1\right) dx = 5.33$$

$$a_{13} = 2^{(1)}2^{(2)} \left[-\int_{\frac{1}{2}}^{\frac{3}{4}} \left(x^2 + 1\right) dx + \int_{\frac{3}{4}}^{1} \left(x^2 + 1\right) dx \right] = 0.75$$

$$a_{15} = 2^{(1)}2^{(3)} \left[\int_{\frac{1}{4}}^{\frac{3}{8}} \left(x^2 + 1\right) dx - \int_{\frac{3}{8}}^{\frac{1}{2}} \left(x^2 + 1\right) dx \right] = -0.1875$$

$$\mathbf{A} = \begin{bmatrix} 5.33 & -0.25 & 0.75 & -0.0625 & -0.1875 & 0.3125 & 0.4375 \\ -0.25 & 8.67 & 0 & -0.125 & 0.375 & 0 & 0 \\ 0.75 & 0 & 12.67 & 0 & 0 & -0.625 & 0.875 \\ -0.0625 & -0.125 & 0 & 16.33 & 0 & 0 & 0 \\ -0.1875 & 0.375 & 0 & 0 & 18.33 & 0 & 0 \\ 0.3125 & 0 & -0.625 & 0 & 0 & 22.33 & 0 \\ 0.4375 & 0 & 0.875 & 0 & 0 & 0 & 28.33 \end{bmatrix}$$

It is clear that the stiffness matrix $\mathbf{A}$ obtained from the alternative basis is symmetric.

Suppose $a(x)$ is a constant function such that $a(x) = C$. Then, the stiffness matrix $\mathbf{A}$ can be simplified to a diagonal matrix with the entries

$$a_{ii} = 2^{2n_i} C\left(x_{i_f} - x)i_0\right) = 2^{n_{i+1}} C$$

If $C = 1$, then the diagonal stiffness matrix $\mathbf{A}$ obtained from the alternative basis corresponds to Equation 2.2.1.

Next, the entries of $\mathbf{b}$ corresponding to Equation 2.4.2 can be computed using the alternative basis $\{\psi_i(x) : i = 1, 2, ..., 2^m - 1\}$.

$$b_i = \int_0^1 f(x)\,\psi_i(x)\,dx$$

$$b_i = \int_{x_{i_0}}^{x_i} f(x)\,\psi_i(x)\,dx + \int_{x_i}^{x_{i_f}} f(x)\,\psi_i(x)\,dx$$

$$b_i = \int_{x_{i_0}}^{x_i} f(x)\left[2^{n_i}x - 2^{n_i}x_i + 1\right]dx + \int_{x_i}^{x_{i_f}} f(x)\left[-2^{n_i}x + 2^{n_i}x_i + 1\right]dx$$

3.3 Modified Stiffness Matrix

Consider the stiffness matrix $\mathbf{A}$ that was developed from the alternative basis $\{\psi_i(x) : i = 1, 2, ..., 2^m - 1\}$, and let

$$\mathbf{D}^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{\sqrt{a_{1,1}}} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\sqrt{a_{2,2}}} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \frac{1}{\sqrt{a_{2^m-2,2^m-2}}} & 0 \\ 0 & \dots & 0 & 0 & \frac{1}{\sqrt{a_{2^m-1,2^m-1}}} \end{bmatrix}$$

In other words, the diagonal entries $d_{ii} = (a_{ii})^{-\frac{1}{2}}$, and the remaining entries $d_{ij} = 0$.

Then, let $\mathbf{B}$ be the modified stiffness matrix such that

$$\mathbf{B} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} = [B_{ij}] = [d_{ii}a_{ij}d_{jj}]$$

Therefore, solving Equation 2.3.2 is equivalent to solving the system

$$\left(\mathbf{D}^{\frac{1}{2}}\mathbf{B}\mathbf{D}^{\frac{1}{2}}\right)\mathbf{c} = \mathbf{b}$$

$$\mathbf{B}\mathbf{D}^{\frac{1}{2}}\mathbf{c} = \mathbf{D}^{-\frac{1}{2}}\mathbf{b}$$

Let $\widehat{\mathbf{b}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{b} = \left[\widehat{b_i}\right] = [d_{ii}b_i]$, and let $\mathbf{y} = \mathbf{D}^{\frac{1}{2}}\mathbf{c}$.
The resulting procedure for solving Equation 2.3.2 is to first solve Equation 3.3.1 for $\mathbf{y}$ and then solve Equation 3.3.2 for $\mathbf{c}$.

$$\mathbf{B}\mathbf{y} = \widehat{\mathbf{b}} \tag{3.3.1}$$

$$\mathbf{D}^{\frac{1}{2}}\mathbf{c} = \mathbf{y} \Rightarrow \mathbf{c} = \mathbf{D}^{-\frac{1}{2}}\mathbf{y} = [c_i] = [d_{ii}y_i] \tag{3.3.2}$$

The indirect numerical method that will be utilized to solve Equation 3.3.1 will be discussed in the next chapter. It is useful to first analyze the modified stiffness matrix $\mathbf{B}$.

Definition 4. The matrix $\mathbf{B}$ is **diagonally dominant** if

$$|b_{ii}| \geq \sum_{i \neq j} |b_{ij}| \text{ for all i}$$

Lemma 3.3.1. *Let a_{ij} be a nonzero entry of the stiffness matrix $\boldsymbol{A}$ where $i > j$. Then,*

$$|a_{ij}| \leq 2^{n_j - n_i + 1} \left|a'(x)\right|_{max}$$

Proof. The Mean Value Theorem for Integrals states that if a function $f(x)$ is continuous on the closed interval $[a, b]$, then there exists a number c in $[a, b]$ such that

$$\int_a^b f(x)\,dx = f(c)(b-a)$$

First, consider

$$\left| \int_{c-\epsilon}^{c} a(x)\,dx - \int_{c}^{c+\epsilon} a(x)\,dx \right| \text{ where } \epsilon > 0$$

By the Mean Value Theorem for Integrals,

$$\left| \int_{c-\epsilon}^{c} a(x)\,dx - \int_{c}^{c+\epsilon} a(x)\,dx \right| = |a(x_1)(c-(c-\epsilon)) - a(x_2)(c+\epsilon-c)| = |a(x_1)\epsilon - a(x_2)\epsilon|$$

where $x_1 \in [c-\epsilon, c]$ and $x_2 \in [c, c+\epsilon]$.

$$\epsilon\,|a(x_1) - a(x_2)| \leq \epsilon\left(2\epsilon\left|a'(x)\right|_{max}\right) = 2\epsilon^2\left|a'(x)\right|_{max}$$

Next, specifically consider the nonzero entries a_{ij} of the stiffness matrix $\mathbf{A}$ where $i > j$.
Then, $n_i > n_j$ and

$$|a_{ij}| = 2^{n_i}2^{n_j}\left| \int_{x_i-\epsilon}^{x_i} a(x)\,dx - \int_{x_i}^{x_i+\epsilon} a(x)\,dx \right| \text{ where } \epsilon = \frac{1}{2^{n_i}}$$

Therefore,

$$|a_{ij}| \leq 2^{n_i} 2^{n_j} 2 \left(\frac{1}{2^{n_i}}\right)^2 |a'(x)|_{max}$$

$$|a_{ij}| \leq 2^{n_j - n_i + 1} |a'(x)|_{max}$$

□

Theorem 3.3.2. *Beginning at some entry b_{kk}, the modified stiffness matrix $\boldsymbol{B}$ is diagonally dominant.*

Proof. Consider the stiffness matrix $\mathbf{A}$. According to Lemma 3.3.1, if $j > i$, then

$$|a_{ij}| \leq 2^{n_i - n_j + 1} |a'(x)|_{max}$$

Recall that the diagonal entries of $\mathbf{A}$ can be computed as

$$a_{ii} = 2^{2n_i} \int_{x_{i_0}}^{x_{i_f}} a(x)\, dx$$

and let

$$m_i = \int_{x_{i_0}}^{x_{i_f}} a(x)\, dx$$

By the Mean Value Theorem for Integrals,

$$m_i = a(c) \left[x_{i_f} - x_{i_0}\right] \text{ for some } c \in (0,1)$$

$$m_i = a(c)\, 2 \left(\frac{1}{2^{n_i}}\right)$$

Let the function $a(x)$ be bounded by the minimum m_0 and the maximum M_0 over the domain $[0,1]$. In other words, let $0 < m_0 \leq a(x) \leq M_0$.

Recall that the entries of the modified stiffness matrix $\mathbf{B}$ can be computed as

$$b_{ij} = d_{ii} a_{ij} d_{jj} = \frac{a_{ij}}{(a_{ii} a_{jj})^{\frac{1}{2}}}$$

Then,

$$|b_{ij}| \leq \frac{2^{n_i - n_j + 1} |a'(x)|_{max}}{(2^{2n_i} m_i 2^{2n_j} m_j)^{\frac{1}{2}}}$$

Notice that $m_i > m_j$, so the inequality can be simplified to

$$|b_{ij}| \leq \frac{2^{n_i - n_j + 1} |a'(x)|_{max}}{\left(2^{2n_i} 2^{2n_j} m_j^2\right)^{\frac{1}{2}}}$$

$$= \frac{2^{n_i - n_j + 1} \left| a'(x) \right|_{max}}{2^{n_i} 2^{n_j} m_j}$$

$$= \frac{2^{n_i - n_j + 1} \left| a'(x) \right|_{max}}{2^{n_i + n_j} a(c) \, 2\left(\frac{1}{2^{n_j}}\right)}$$

Substitution of the minimum m_o for $a(c)$ yields

$$|b_{ij}| \leq \frac{2^{n_i - n_j + 1} \left| a'(x) \right|_{max} 2^{n_j - 1}}{2^{n_i + n_j} m_0}$$

$$|b_{ij}| \leq \frac{\left| a'(x) \right|_{max}}{m_0 2^{n_j}}$$

Next, fix i and sum for all $j > i$.

$$\sum_{j=i+1}^{\infty} |b_{ij}| \leq \sum_{n_j=n_i+1}^{\infty} \left[\frac{\left| a'(x) \right|_{max}}{m_0 2^{n_j}} \right] \left(2^{n_j - n_i}\right)$$

$$\sum_{j=i+1}^{\infty} |b_{ij}| \leq \left[\frac{\left| a'(x) \right|_{max}}{m_0} \right] \left(\frac{1}{2^{n_i}} \right)$$

Now, consider the case when $j < i$. According to Lemma 3.3.1,

$$|a_{ij}| \leq 2^{n_j - n_i + 1} \left| a'(x) \right|_{max}$$

A similar procedure, in which m_i is substituted for m_j because $m_i < m_j$, yields the following inequality for $|b_{ij}|$.

$$|b_{ij}| \leq \frac{2^{n_j - n_i + 1} \left| a'(x) \right|_{max} 2^{n_i - 1}}{2^{n_j + n_i} m_0}$$

$$|b_{ij}| \leq \frac{\left| a'(x) \right|_{max}}{m_0 2^{n_i}}$$

Next, fix i and sum for all $j < i$.

$$\sum_{j=1}^{i-1} |b_{ij}| \leq \sum_{n_j=1}^{n_i-1} \left[\frac{\left| a'(x) \right|_{max}}{m_0 2^{n_i}} \right] \left(2^{n_i - n_j}\right)$$

$$\sum_{j=1}^{i-1} |b_{ij}| \leq \left[\frac{\left| a'(x) \right|_{max}}{m_0} \right] \sum_{n_j=1}^{n_i-1} \left(\frac{1}{2^{n_j}} \right)$$

$$\sum_{j=1}^{i-1} |b_{ij}| \leq \left[\frac{\left| a'(x) \right|_{max}}{m_0} \right] \left(1 - \frac{1}{2^{n_i - 1}} \right)$$

In order to show that the modified stiffness matrix $\mathbf{B}$ becomes diagonally dominant beginning

with some entry b_{kk}, consider the following sum of off-diagonal entries.

$$\sum_{j=k}^{i-1} |b_{ij}| + \sum_{j=i+1}^{\infty} |b_{ij}| \leq \left[\frac{|a'(x)|_{max}}{m_0} \right] \left[\left(\frac{1}{2^{n_k - 1}} - \frac{1}{2^{n_i - 1}} \right) + \frac{1}{2^{n_i}} \right]$$

Refer to the modified stiffness matrix $\mathbf{B}$ below for a visual representation of the entry b_{kk} where it is expected that $\mathbf{B}$ will begin to exhibit diagonal dominance.

$$\left[\begin{array}{cccc|ccc} b_{1,1} & b_{1,2} & \dots & b_{1,k-1} & b_{1,k} & \dots & b_{1,2^m-1} \\ b_{2,1} & b_{2,2} & \dots & b_{2,k-1} & b_{2,k} & \dots & b_{2,2^m-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ b_{k-1,1} & b_{k-1,2} & \dots & b_{k-1,k-1} & b_{k-1,k} & \dots & b_{k-1,2^m-1} \\ \hline b_{k,1} & \dots & & b_{k,k-1} & b_{k,k} & b_{k,k+1} & \dots & b_{k,2^{m-1}} \\ b_{k+1,1} & \dots & & b_{k+1,k-1} & b_{k+1,k} & b_{k+1,k+1} & \dots & b_{k+1,2^m-1} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{2^m-1,1} & \dots & & b_{2^m-1,k-1} & b_{2^m-1,k} & b_{2^m-1,k+1} & \dots & b_{2^m-1,2^m-1} \end{array}\right]$$

Then,

$$\sum_{j=k, j\neq i}^{\infty} |b_{ij}| \leq \left[\frac{|a'(x)|_{max}}{m_0} \right] \left(\frac{1}{2^{n_k - 1}} - \frac{1}{2^{n_i}} \right)$$

$$\lim_{n_k \to \infty} \left(\frac{1}{2^{n_k - 1}} - \frac{1}{2^{n_i}} \right) = 0$$

Therefore, there must eventually occur a row k of the modified stiffness matrix $\mathbf{B}$ such that

$$\sum_{j=k, j\neq i}^{\infty} |b_{ij}| \leq |b_{ii}| = 1$$

which is the definition of diagonal dominance. □

Chapter 4

Implementation

4.1 Numerical Methods for Linear Systems of Equations

Generally, an indirect numerical method will be implemented in the programming language C++ in order to solve Equation 2.3.2 developed from the typical basis and Equation 3.3.1, which corresponds to the alternative basis. Consider $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is an n-by-n square matrix. Addition and subtraction of the matrix $\mathbf{Q}$ yields an equivalent system.

$$(\mathbf{Q} + \mathbf{A} - \mathbf{Q})\,\mathbf{x} = \mathbf{b}$$

$$\mathbf{Qx} = (\mathbf{Q} - \mathbf{A})\,\mathbf{x} + \mathbf{b}$$

$$\mathbf{x} = \mathbf{Q}^{-1}\left[(\mathbf{Q} - \mathbf{A})\,\mathbf{x} + \mathbf{b}\right]$$

$$\mathbf{x} = \mathbf{Q}^{-1}\,(\mathbf{Q} - \mathbf{A})\,\mathbf{x} + \mathbf{Q}^{-1}\mathbf{b}$$

Next, create a fixed point iteration from the equivalent system.

$$\mathbf{x}^{(k+1)} = \mathbf{Q}^{-1}\,(\mathbf{Q} - \mathbf{A})\,\mathbf{x}^{(k)} + \mathbf{Q}^{-1}\mathbf{b} \text{ where } k = 0, 1, 2, ...$$

If $\mathbf{Q}$ equals the n-by-n identity matrix $\mathbf{I}$, then this iterative method for solving systems of linear equations is known as the Richardson Method. If $\mathbf{Q}$ equals the diagonal matrix $\mathbf{D}$ corresponding to matrix $\mathbf{A}$, then this iterative method is known as the Jacobi Method. If $\mathbf{Q}$ equals the lower triangular matrix $\mathbf{L}$ corresponding to matrix $\mathbf{A}$ in addition to the diagonal matrix $\mathbf{D}$ corresponding to matrix $\mathbf{A}$, then this iterative method is known as the Gauss-Seidel Method. The Jacobi Method is the indirect numerical method that has been chosen for this implementation because it converges to the solution vector $\mathbf{x}$ for all diagonally dominant matrices, and the modified stiffness matrix $\mathbf{B}$ must eventually become diagonally dominant according to Theorem 3.3.2. Note that the Gauss-Seidel Method also converges for diagonally dominant matrices and could have been

implemented in lieu of the Jacobi Method. Before proving the convergence of the Jacobi Method for diagonally dominant matrices, consider the following definitions:

Definition 5. Consider the vector $\mathbf{v}$. A **vector norm** $\|\mathbf{v}\|$ has the following properties:

1. $\|\mathbf{v}\| > 0$ when $\mathbf{v} \neq \mathbf{0}$ and $\mathbf{v} = 0 \Leftrightarrow \mathbf{v} = \mathbf{0}$
2. $\|\lambda\mathbf{v}\| = |\lambda| \|\mathbf{v}\|$ for any scalar λ.
3. $\|\mathbf{v} + \mathbf{w}\| \leq \|\mathbf{v}\| + \|\mathbf{w}\|$

The following vector norms will be used to check the accuracy of the results of the implementation:

$$\|\mathbf{v}\|_1 = \sum_i |v_i|$$

$$\|\mathbf{v}\|_2 = \sqrt{\sum_i |v_i|^2}$$

$$\|\mathbf{v}\|_\infty = \max_i |v_i|$$

Definition 6. Consider the square n-by-$n$matrix $\mathbf{A}$. A **matrix norm** $\|\mathbf{A}\|$ has the following properties:

1. $\|\mathbf{A}\| > 0$ when $\mathbf{A} \neq \mathbf{0}$ and $\mathbf{A} = 0 \Leftrightarrow \mathbf{A} = \mathbf{0}$
2. $\|\lambda\mathbf{A}\| = |\lambda| \|\mathbf{A}\|$ for any scalar λ.
3. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$
4. $\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\|$

In general, the matrix norm $\|\mathbf{A}\|$ is derived from the corresponding vector norm. The vector norm $\|\mathbf{Ax}\|$ is divided by the vector norm $\|\mathbf{x}\|$ for all n-dimensional vectors $\mathbf{x}$, and the supremum is the matrix norm $\|\mathbf{A}\|$. In other words,

$$\|\mathbf{A}\| = \sup \left\{ \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|}, \text{ where } \mathbf{x} \text{ is any } n\text{-dimensional vector} \right\}$$

Therefore, the matrix norm $\|\mathbf{A}\|$ has an additional property:

$$\|\mathbf{Ax}\| \leq \|\mathbf{A}\| \|\mathbf{x}\|$$

The following matrix norm will be used in the proof that the Jacobi Method converges for all diagonally dominant matrices:

$$\|\mathbf{A}\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|$$

Theorem 4.1.1. *Consider $\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A}$ is an n-by-n diagonally dominant matrix. Then, for every vector $\boldsymbol{b}$ and every starting guess $\boldsymbol{x}^{(0)}$, the Jacobi Method applied to $\boldsymbol{Ax} = \boldsymbol{b}$ converges to the solution $\boldsymbol{x}$.*

Proof. Recall the equations for the exact solution $\mathbf{x}$ and the approximate solution $\mathbf{x}^{(k+1)}$ in the derivation of the Jacobi Method.

$$\mathbf{x} = \mathbf{Q}^{-1}(\mathbf{Q} - \mathbf{A})\mathbf{x} + \mathbf{Q}^{-1}\mathbf{b}$$

$$\mathbf{x}^{(k+1)} = \mathbf{Q}^{-1}(\mathbf{Q} - \mathbf{A})\mathbf{x}^{(k)} + \mathbf{Q}^{-1}\mathbf{b}$$

Let $\mathbf{e}^{(k)}$ be the error between the exact solution $\mathbf{x}$ and the approximate solution $\mathbf{x}^{(k)}$. In other words,

$$e^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}$$

Then,

$$\mathbf{e}^{(k+1)} = \mathbf{x}^{(k+1)} - \mathbf{x}$$

$$\mathbf{e}^{(k+1)} = \mathbf{Q}^{-1}(\mathbf{Q} - \mathbf{A})\mathbf{x}^{(k)} + \mathbf{Q}^{-1}\mathbf{b} - \left[\mathbf{Q}^{-1}(\mathbf{Q} - \mathbf{A})\mathbf{x} + \mathbf{Q}^{-1}\mathbf{b}\right]$$

$$\mathbf{e}^{(k+1)} = \mathbf{Q}^{-1}(\mathbf{Q} - \mathbf{A})\mathbf{x}^{(k)} - \mathbf{Q}^{-1}(\mathbf{Q} - \mathbf{A})\mathbf{x}$$

$$\mathbf{e}^{(k+1)} = \mathbf{Q}^{-1}(\mathbf{Q} - \mathbf{A})\left(\mathbf{x}^{(k)} - \mathbf{x}\right)$$

$$\mathbf{e}^{(k+1)} = \mathbf{Q}^{-1}(\mathbf{Q} - \mathbf{A})\,e^{(k)}$$

Let $\mathbf{G} = \mathbf{Q}^{-1}(\mathbf{Q} - \mathbf{A})$, so

$$\mathbf{e}^{(k+1)} = \mathbf{G}\mathbf{e}^{(k)} = \mathbf{G}\mathbf{G}\mathbf{e}^{(k-1)} = \ldots = \mathbf{G}^{k+1}\mathbf{e}^{(0)}$$

In general, for the k^{th} iteration

$$\mathbf{e}^{(k)} = \mathbf{G}^k\mathbf{e}^{(0)}$$

$$\left\|\mathbf{e}^{(k)}\right\| = \left\|\mathbf{G}^k\mathbf{e}^{(0)}\right\|$$

By the additional property in Definition 6,

$$\left\|\mathbf{e}^{(k)}\right\| \leq \left\|\mathbf{G}^k\right\| \left\|\mathbf{e}^{(0)}\right\|$$

By the fourth property in Definition 6,

$$\left\|\mathbf{G}^k\right\| \leq \underbrace{\|\mathbf{G}\|\,\|\mathbf{G}\|\ldots\|\mathbf{G}\|}_{k \text{ times}}$$

If $\|\mathbf{G}\| < 1$, then

$$\left\|\mathbf{G}^k\right\| \leq \|\mathbf{G}\|^k \to 0$$

Therefore,

$$\left\|\mathbf{G}^k\right\| \to 0 \Rightarrow \left\|\mathbf{e}^k\right\| \to 0 \Rightarrow \mathbf{e}^k \to 0$$

which shows that the Jacobi Method is convergent. It remains to check that $\|\mathbf{G}\| < 1$. As previously indicated, the matrix norm that will be used is

$$\|\mathbf{G}\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |g_{ij}|$$

Also, recall that the matrix $\mathbf{Q}$ for the Jacobi Method is the diagonal matrix $\mathbf{D}$ corresponding to matrix $\mathbf{A}$.

$$\mathbf{G} = \mathbf{Q}^{-1}(\mathbf{Q} - \mathbf{A}) = \mathbf{I} - \mathbf{Q}^{-1}\mathbf{A}$$

$$\mathbf{G} = \mathbf{I} - \begin{bmatrix} (a_{1,1})^{-1} & 0 & \dots & 0 \\ 0 & (a_{2,2})^{-1} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & (a_{n-1,n-1})^{-1} & 0 \\ 0 & \dots & 0 & (a_{n,n})^{-1} \end{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \dots & a_{n-1,n} \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix}$$

$$\mathbf{G} = \mathbf{I} - \begin{bmatrix} 1 & \frac{a_{1,2}}{a_{1,1}} & \dots & \frac{a_{1,n}}{a_{1,1}} \\ \frac{a_{2,1}}{a_{2,2}} & 1 & \dots & \frac{a_{2,n}}{a_{2,2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{n-1,1}}{a_{n-1,n-1}} & \frac{a_{n-1,2}}{a_{n-1,n-1}} & \dots & \frac{a_{n-1,n}}{a_{n-1,n-1}} \\ \frac{a_{n,1}}{a_{n,n}} & \frac{a_{n,2}}{a_{n,n}} & \dots & 1 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 0 & \frac{a_{1,2}}{a_{1,1}} & \dots & \frac{a_{1,n}}{a_{1,1}} \\ \frac{a_{2,1}}{a_{2,2}} & 0 & \dots & \frac{a_{2,n}}{a_{2,2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{n-1,1}}{a_{n-1,n-1}} & \frac{a_{n-1,2}}{a_{n-1,n-1}} & \dots & \frac{a_{n-1,n}}{a_{n-1,n-1}} \\ \frac{a_{n,1}}{a_{n,n}} & \frac{a_{n,2}}{a_{n,n}} & \dots & 0 \end{bmatrix}$$

For row i,

$$\sum_{j=1}^{n} |g_{ij}| = \sum_{j=1}^{n} \left|\frac{a_{ij}}{a_{ii}}\right| = \frac{1}{|a_{ii}|} \sum_{j=1}^{n} |a_{ij}|$$

In order to satisfy the inequality $\|\mathbf{G}\|_\infty < 1$, the following inequality must be true for each row i:

$$\frac{1}{|a_{ii}|} \sum_{j=1}^{n} |a_{ij}| < 1$$

$$\sum_{j=1}^{n} |a_{ij}| < |a_{ii}|$$

Therefore, the Jacobi Method is convergent if the matrix $\mathbf{A}$ is strictly diagonally dominant. $\square$

The algorithm for implementing the Jacobi Method is the following:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}}$$

Furthermore, the Jacobi Method can be ordered to cease iterations once a specified tolerance has been satisfied. For this implementation, the following tolerance will be utilized:

$$\left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\|_2 < \text{ tol}$$

4.2 Results

Recall Equation 2.4.2, which is a general one-dimensional boundary value problem. Let $a(x) = x^2 + 1$ and $f(x) = 6x^2 - 2x + 2$ so that the solution is known to be $u(x) = x(1-x)$.

$$\begin{cases} -\Big(\left(x^2+1\right) u'(x) \Big)' = 6x^2 - 2x + 2 \quad \text{in } (0,1) \\ u(0) = u(1) = 0 \end{cases} \tag{4.2.1}$$

Equation 4.2.1 is the boundary value problem that was considered for the entirety of the implementation. It is possible to consider other boundary value problems, but the functions $a(x)$ and $f(x)$ have been limited to polynomials in order to simplify the necessary methods in the programming language C++. For example, nested multiplication, also known as Horner's Method, is the numerical method that was utilized for evaluating the polynomials at specified x-values. Furthermore, polynomials can be easily differentiated and integrated by direct means in the programming language C++.

Results have been compiled for the spaces S_8, S_{16}, S_{32}, and S_{64} using both the typical basis and the alternative basis for each space. In addition, tolerances of 0.01, 0.001, 0.0001, and 0.00001 were considered, in order to view the Jacobi Method's rate of convergence. Figure 4.1 illustrates the approximate solution approaching the exact solution as the tolerance is decreased from 0.01 to 0.001 to 0.0001 for the space S_8 using the typical basis. Figures 4.2, 4.3, and 4.4 provide a visual representation of the approximate solution aligning with the exact solution at the nodes $\{x_i\}$ as the tolerance approaches zero for the spaces S_8, S_{16}, and S_{32} respectively.

As expected, the number of iterations of the Jacobi Method increases as the tolerance decreases. Refer to Figures 4.5 and 4.6 in order to view the number of Jacobi Method iterations needed to solve Equation 2.3.2 using the typical basis and Equation 3.3.1 corresponding to the alternative basis respectively. The rather remarkable result that can be gleaned from these two figures is that the number of Jacobi Method iterations for the typical basis increases rapidly as the dimension of the space increases. However, the number of Jacobi Method iterations for the alternative basis and modified stiffness matrix appears to stabilize as the dimension of the space increases.

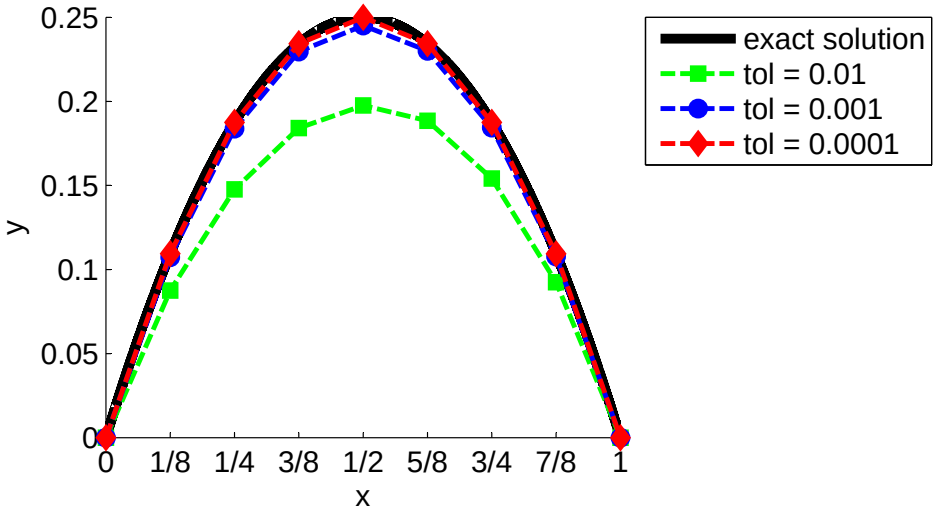

Figure 4.1. Approximate Solution for the Space S_8 Using the Typical Basis for Various Tolerances

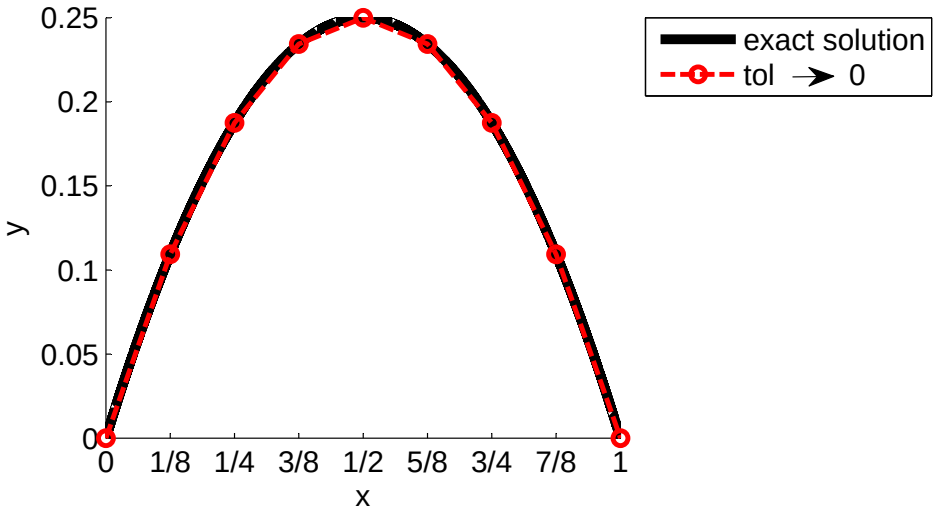

Figure 4.2. Approximate Solution for the Space S_8 as the Tolerance Approaches 0

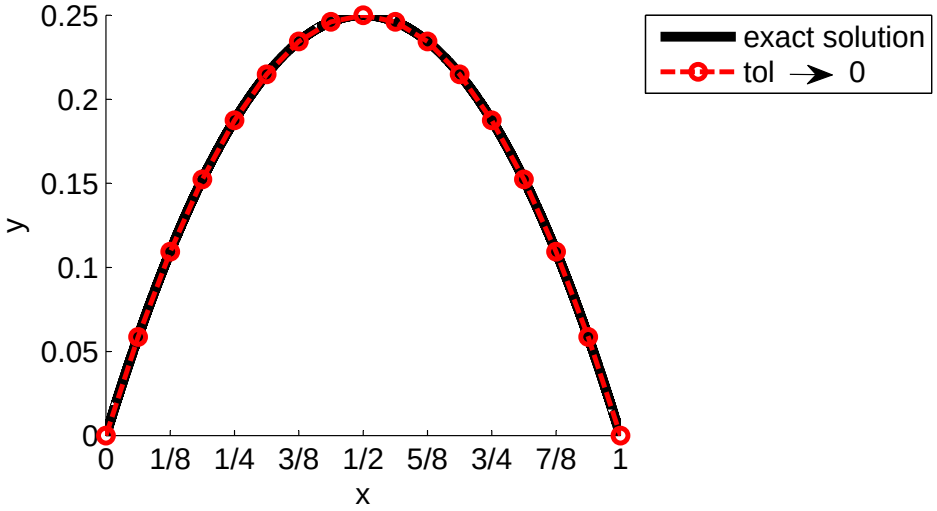

Figure 4.3. Approximate Solution for the Space S_{16} as the Tolerance Approaches 0

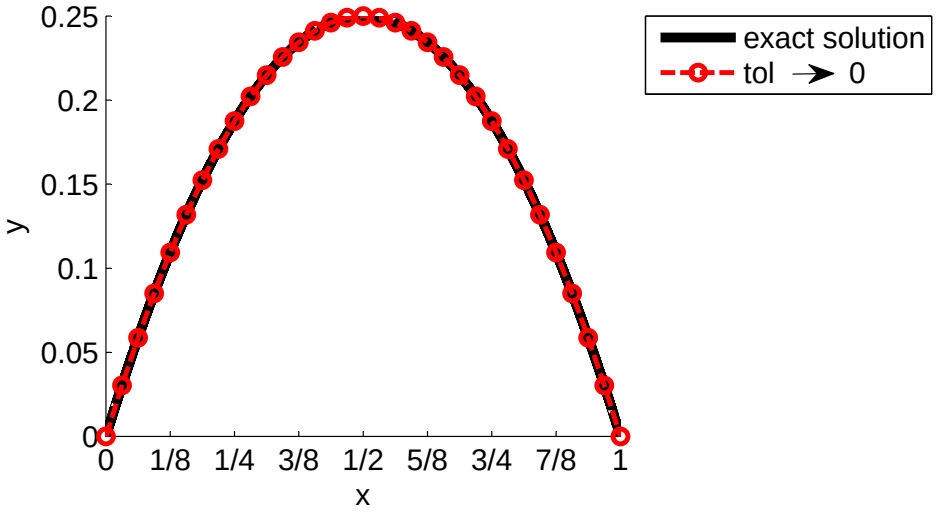

Figure 4.4. Approximate Solution for the Space S_{32} as the Tolerance Approaches 0

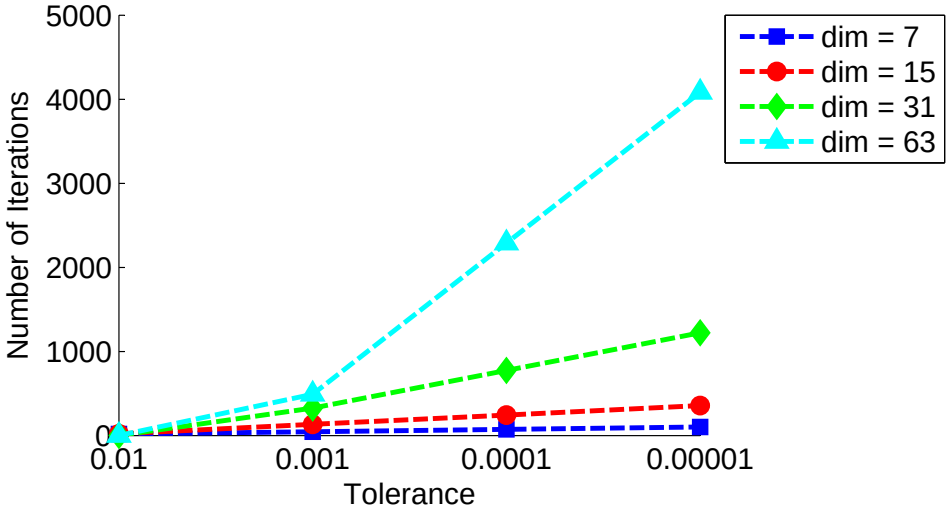

Figure 4.5. Computation of the Approximate Solution Using the Typical Basis: Number of Iterations of the Jacobi Method versus Tolerance

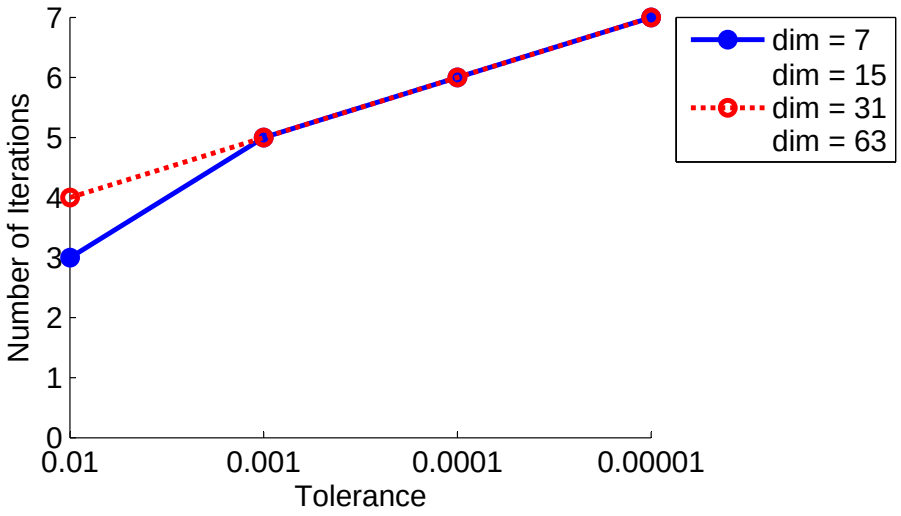

Figure 4.6. Computation of the Approximate Solution Using the Alternative Basis: Number of Iterations of the Jacobi Method versus Tolerance

Because the exact solution is known in the implementation, it is possible to compare the approximate solutions that have been computed for a variety of spaces and tolerances to the exact solution. All three vector norms that were provided in Definition 5 have been used to compute errors between the approximate solutions and the exact solution.

Let $\{m_i\}$ be the midpoints between the nodes $\{x_i\}$. In other words,

$$\{m_j\} = \left\{ \frac{x_1}{2}, \frac{x_i + x_{i+1}}{2}, \frac{x_{n-1} + 1}{2} \right\}$$

where $i = 1, ..., n-1$ and $n-1$ equals the dimension of the subspace S_n

Then, let the vector **diff** be the difference between the exact solution $u(x)$ and the approximate solution $u_s(x)$ at the midpoints $\{m_j\}$.

$$\mathbf{diff} = \begin{bmatrix} u(m_1) - u_s(m_1) \\ u(m_2) - u_s(m_2) \\ \vdots \\ u(m_n) - u_s(m_n) \end{bmatrix}$$

Then, $\|\mathbf{diff}\|_1$, $\|\mathbf{diff}\|_2$, and $\|\mathbf{diff}\|_\infty$ are the errors.

In addition to computing errors between the approximate solutions and the exact solution, the backward error of the Jacobi Method can be computed. Let the vector $\mathbf{c}_s$ be the solution to Equation 2.3.2 that corresponds to a specified tolerance for the Jacobi Method. Let the vector **residual** equal $\mathbf{A}\mathbf{c}_s - \mathbf{b}$. Then, the backward error that was considered in the implementation is $\|\mathbf{residual}\|_2$.

Refer to Tables 4.1 through 4.8 for the errors between the approximate solutions and the exact solution, the backward errors, and the specific number of Jacobi Method iterations according to specified tolerance. The tables are separated according to subspace and basis.

It is clear from the compiled results that the alternative basis and modified stiffness matrix configuration outperforms the typical basis in terms of efficiently obtaining an accurate approximate solution to Equation 4.2.1. However, it should be noted that the boundary value problem that was used in the implementation represents a somewhat idealized scenario. Each modified stiffness matrix $\mathbf{B}$ that was created for the problem was diagonally dominant from the very beginning. It is expected that the results might not be as good for a boundary value problem in which the modified stiffness matrix $\mathbf{B}$ does not become diagonally dominant until entry b_{kk}, where k is a rather large number.

Space S_8: Typical Basis					
Tolerance	# of Iterations	$\|\mathbf{residual}\|_2$	$\|\mathbf{diff}\|_1$	$\|\mathbf{diff}\|_2$	$\|\mathbf{diff}\|_\infty$
0.01	19	0.17380680	0.26072481	0.10215886	0.05133978
0.001	46	0.01790170	0.02459118	0.00964411	0.00485510
0.0001	73	0.00184380	0.00027218	0.00012113	0.00007154
0.00001	101	0.00017460	0.00226753	0.00087817	0.00043147

Table 4.1. Approximate Solution for the Space S_8 Using the Typical Basis: Tolerance and Number of Iterations for the Jacobi Method, Backward Error, and Errors when Compared with the Exact Solution

Space S_8: Alternative Basis					
Tolerance	# of Iterations	$\|\mathbf{residual}\|_2$	$\|\mathbf{diff}\|_1$	$\|\mathbf{diff}\|_2$	$\|\mathbf{diff}\|_\infty$
0.01	3	0.00424648	0.00346984	0.00134427	0.00065948
0.001	5	0.00006658	0.00254799	0.00098720	0.00048526
0.0001	6	0.00000761	0.00252891	0.00098053	0.00048297
0.00001	7	0.00000105	0.00253232	0.00098180	0.00048358

Table 4.2. Approximate Solution for the Space S_8 Using the Alternative Basis: Tolerance and Number of Iterations for the Jacobi Method, Backward Error, and Errors when Compared with the Exact Solution

Space S_{16}: Typical Basis					
Tolerance	# of Iterations	$\|\mathbf{residual}\|_2$	$\|\mathbf{diff}\|_1$	$\|\mathbf{diff}\|_2$	$\|\mathbf{diff}\|_\infty$
0.01	21	0.40241440	1.69944230	0.47281968	0.16836087
0.001	133	0.03928330	0.16717948	0.04656459	0.01663263
0.0001	244	0.00396680	0.01573126	0.00438592	0.00156987
0.00001	356	0.00039240	0.00040262	0.00011803	0.00004534

Table 4.3. Approximate Solution for the Space S_{16} Using the Typical Basis: Tolerance and Number of Iterations for the Jacobi Method, Backward Error, and Errors when Compared with the Exact Solution

Space S_{16}: Alternative Basis					
Tolerance	# of Iterations	$\|\mathbf{residual}\|_2$	$\|\mathbf{diff}\|_1$	$\|\mathbf{diff}\|_2$	$\|\mathbf{diff}\|_\infty$
0.01	4	0.00064891	0.00080497	0.00023170	0.00008774
0.001	5	0.00009248	0.00132236	0.00036258	0.00012496
0.0001	6	0.00001160	0.00127188	0.00035006	0.00012124
0.00001	7	0.00000163	0.00128088	0.00035240	0.00012211

Table 4.4. Approximate Solution for the Space S_{16} Using the Alternative Basis: Tolerance and Number of Iterations for the Jacobi Method, Backward Error, and Errors when Compared with the Exact Solution

Space S_{32}: Typical Basis					
Tolerance	# of Iterations	$\|\mathbf{residual}\|_2$	$\|\mathbf{diff}\|_1$	$\|\mathbf{diff}\|_2$	$\|\mathbf{diff}\|_\infty$
0.01	1	0.53237620	5.29550487	1.02626112	0.24854675
0.001	327	0.08086930	0.97983645	0.19321674	0.04876279
0.0001	775	0.00808120	0.09737176	0.01920298	0.00484752
0.00001	1223	0.00080780	0.00915559	0.00180718	0.00045694

Table 4.5. Approximate Solution for the Space S_{32} Using the Typical Basis: Tolerance and Number of Iterations for the Jacobi Method, Backward Error, and Errors when Compared with the Exact Solution

Space S_{32}: Alternative Basis					
Tolerance	# of Iterations	$\|\mathbf{residual}\|_2$	$\|\mathbf{diff}\|_1$	$\|\mathbf{diff}\|_2$	$\|\mathbf{diff}\|_\infty$
0.01	4	0.00071665	0.00042786	0.00009153	0.00002612
0.001	5	0.00010236	0.00073216	0.00014121	0.00003418
0.0001	6	0.00001316	0.00062430	0.00012175	0.00002995
0.00001	7	0.00000186	0.00064353	0.00012528	0.00003074

Table 4.6. Approximate Solution for the Space S_{32} Using the Alternative Basis: Tolerance and Number of Iterations for the Jacobi Method, Backward Error, and Errors when Compared with the Exact Solution

Space S_{64}: Typical Basis					
Tolerance	# of Iterations	$\|\mathbf{residual}\|_2$	$\|\mathbf{diff}\|_1$	$\|\mathbf{diff}\|_2$	$\|\mathbf{diff}\|_\infty$
0.01	1	0.39085720	10.64745050	1.45827856	0.24963528
0.001	492	0.16450400	5.59767800	0.78040230	0.13919330
0.0001	2290	0.01624500	0.55693879	0.07768510	0.01387446
0.00001	4083	0.00162600	0.05545555	0.00773583	0.00138181

Table 4.7. Approximate Solution for the Space S_{64} Using the Typical Basis: Tolerance and Number of Iterations for the Jacobi Method, Backward Error, and Errors when Compared with the Exact Solution

Space S_{64}: Alternative Basis					
Tolerance	# of Iterations	$\|\mathbf{residual}\|_2$	$\|\mathbf{diff}\|_1$	$\|\mathbf{diff}\|_2$	$\|\mathbf{diff}\|_\infty$
0.01	4	0.00074340	0.00167311	0.00024007	0.00004790
0.001	5	0.00010605	0.00050483	0.00006969	0.00001234
0.0001	6	0.00001374	0.00028556	0.00003963	0.00000698
0.00001	7	0.00000194	0.00032467	0.00004466	0.00000773

Table 4.8. Approximate Solution for the Space S_{64} Using the Alternative Basis: Tolerance and Number of Iterations for the Jacobi Method, Backward Error, and Errors when Compared with the Exact Solution

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The basic mathematical theory of the finite element method has been developed for the one-dimensional case. It has been shown that the weak formulation given by Equation 2.1.2 is equivalent to the one-dimensional boundary value problem presented in Equation 2.1.1. The original function space was replaced with a finite dimensional subspace in order to transform the problem into a linear system of equations that produces a unique, approximate solution. Refer to Equation 2.3.2 for the system. Furthermore, a more general version of the original one-dimensional boundary value problem with different boundary conditions was considered in Equation 2.4.2. The basis that is typically utilized in the discretization process was discussed and presented in Equation 2.4.1; it consists of shifted and equally scaled triangle functions.

The common usage of the basis in Equation 2.4.1 does not preclude the development of an alternative basis. Refer to Equation 3.2.1 for the alternative basis as well as Figure 3.4 for a visual representation of the alternative basis for the subspace S_8. It is clear that the alternative basis was constructed from subsets of the typical basis. Then, the stiffness matrix that was constructed from the alternative basis was modified in an effort to create a diagonally dominant matrix; it was shown that the modified stiffness matrix must eventually become diagonally dominant. The modification resulted in Equations 3.3.1 and 3.3.2, which are equivalent to the system in Equation 2.3.2.

The iterative Jacobi Method was implemented in the programming language C++ in order to solve Equation 2.3.2 corresponding to the typical basis and Equation 3.3.1 corresponding to the alternative basis and modified stiffness matrix. The Jacobi Method was chosen because it is convergent for diagonally dominant matrices. The one-dimensional boundary value problem in Equation 4.2.1 was considered for the implementation. Then the subspaces S_8, S_{16}, S_{32}, and

S_{64} as well as a variety of tolerances were used to compute approximate solutions corresponding to both the typical and alternative bases. An analysis of the required number of Jacobi Method iterations, backward error, and errors between the approximate solutions and the known solution, provided support for the alternative basis as a more efficient basis without compromising accuracy.

5.2 Future Work

As previously discussed, another indirect numerical method for solving linear systems of equations that also converges for diagonally dominant matrices is the Gauss-Seidel Method. It is expected that the Gauss-Seidel Method will converge even more quickly than the Jacobi Method, so future work might involve the implementation of the Gauss-Seidel Method in the programming language C++.

It was also noted that the one-dimensional boundary value problem presented in Equation 4.2.1 and used for the implementation represents a somewhat idealized scenario. In addition, the program created in the programming language C++ limits functions in the boundary value problem to be polynomials. Therefore, future work could involve the expansion of the program to accept boundary value problems with more difficult functions. Then, numerical methods, such as quadrature formulas, would need to be utilized for the integration of the more difficult functions instead of direct means.

Finally, the ultimate goal of any future work should be to extend the concepts developed in this thesis with regard to the one-dimensional case to the two-dimensional case.

Bibliography

[1] Sheldon Axler. *Linear Algebra Done Right*. Springer Science+Business Media, 2nd edition, 1997.

[2] Susanne C. Brenner and L. Ridgeway Scott. *The Mathematical Theory of Finite Element Methods*. Springer Verlag New York, Inc., 1994.

[3] Kenneth H. Huebner. *The Finite Element Method for Engineers*. John Wiley and Sons, Inc., 2001.

[4] Darrell W. Pepper and Juan C. Heinrich. *The Finite Element Method: Basic Concepts and Applications*. Hemisphere Publishing Corporation, 1992.

[5] Timothy Sauer. *Numerical Analysis*. Addison Wesley, 2005.

[6] James Stewart. *Calculus*. Brooks Cole, 2002.

[7] William Gilbert Strang. *An Analysis of the Finite Element Method*. Prentice-Hall, 1973.

Vita

Carol Lynn Gaertner

443 Nedham Court
Wexford, PA 15090

clg5017@psu.edu
carolgaertner@gmail.com

Education:

The Pennsylvania State University, Summer 2010

Bachelor of Science in Mathematics, General Option
Honors in Mathematics
Thesis Title: More Efficient Choices of Linear Bases in Numerical Methods
Thesis Supervisor: Victor Nistor

Bachelor of Architectural Engineering, Structural Option
Honors in Architectural Engineering
Thesis Title: A Study of the Engineering Systems of the 8th Street Office Building in Richmond, Virginia, with an Emphasis on Structural Design
Thesis Supervisor: M. Kevin Parfitt

Related Experience:

Penn State Learning
Math Tutor Coordinator
Spring 2008 - Summer 2010

Rathgeber | Goss Associates
Engineering Intern
Summer 2009

CLT Efficient Technologies Group
Engineering Intern
Summer 2007

Pennsylvania Department of Transportation
Engineering, Scientific, and Technical Intern
Summer 2006

Awards:

Schreyer Honors College Scholar
Dean's List
Engineer-In-Training
Phi Kappa Phi National Honor Society
Tau Beta Pi Engineering Honor Society
Phi Alpha Epsilon Architectural Engineering Honor Society
Gladys M. Baird Scholarship in Architectural Engineering

Professor Vincent L. Pass Scholarship in Architectural Engineering
H. Thomas and Dorothy Willits Hallowell Scholarship
C. Melville, Jr. and Kenneth Barr Scholarship

Activities:

Schreyer Honors College Mentoring Program
Mentor: Art Glenn
Schreyer Honors College THON Team, Finance Officer
Architectural Engineering Envoy, Tour Guide and Liaison
SCUBA: Kinesiology 45, Teaching Assistant; Nittany Divers Club, Member