CUBIC SPLINE FINITE ELEMENT METHOD FOR SOLVING POISSON'S EQUATION ON A SQUARE

by

Stephanie Beck

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Applied Mathematics and Statistics).

Golden, Colorado

Date ____________________

Signed: ____________________

Stephanie Beck

Signed: ____________________

Dr. Bernard Bialecki

Thesis Advisor

Golden, Colorado

Date ____________________

Signed: ____________________

Dr. Willy Hereman

Professor and Head

Department of Applied Mathematics and Statistics

ABSTRACT

We solve a simple 2-point boundary value problem with Dirichlet boundary conditions on $[0, 1]$ using a finite element method with cubic splines. We obtain explicit forms for the mass and stiffness matrices that arise from the method. We then solve Poisson's equation with zero Dirichlet boundary conditions in the unit square using a finite element method with basis functions that are tensor products of cubic splines. The resulting linear systems are solved in Matlab using Gauss elimination without pivoting, and then more efficiently using a Matrix Decomposition Algorithm. Cost of the matrix decomposition algorithm is $\mathcal{O}(N^3)$, where $N+1$ is the number of subintervals in each coordinate direction. We improve the method still by using an Alternating Direction Implicit Method that reduces the cost of solving the resulting linear systems to $\mathcal{O}(N^2 \ln^2 N)$.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to my advisor Dr. Bernard Bialecki, who has demonstrated continued patience and extensive knowledge in his field. Without his supervision and constant guidance, this thesis would not have been possible. I would also like to thank my committee members, Dr. Paul Martin and Dr. Mahadevan Ganesh, for their patience and constructive comments.

Special thanks to my fellow graduate students, Greg Johnson and Jason Gates, for their assistance proof-reading and debugging and their willingness to listen. I would like to offer sincere appreciation for the AMS department administrators Roxanne Aungst and Virginia Barbier for their help with the logistics of defending and submitting my thesis. I would like to also thank Lyndsey Wright for the use of her code and guidance while formatting.

My deepest appreciation goes to my husband Charles for putting up with the long nights, early mornings, and stress and for always encouraging me every step of the way. Also, without the constant support and love from my family, I would have never made it this far.

TABLE OF CONTENTS

LIST OF TABLES

1 INTRODUCTION

We solve Poisson's equation with zero Dirichlet boundary conditions on the unit square using a cubic spline finite element method (FEM). Poisson's equation in two dimensions models steady-state heat flow in a plane, and our problem in particular holds the temperature at zero on the boundaries. We independently obtain explicit forms for the stiffness and mass matrices that arise. Soliman [14] gives the same formulas for the matrix entries on the diagonal and three superdiagonals, but ignores the Dirichlet boundary conditions, which change the corner entries. Standard finite element methods use piecewise linear basis functions that are second-order convergent [6]. Our selection of cubic spline basis functions improves accuracy to fourth-order. We use a Matrix Decomposition Algorithm (MDA) [2] and then an Alternate Direction Implicit method (ADI) [15] to solve the resulting linear system.

MDAs are used to solve linear systems of the form

$$(A \otimes B + B \otimes A)\boldsymbol{\alpha} = \mathbf{F}$$

that arise when finding numerical solutions to certain elliptic boundary value problems. Formulation of an MDA requires that there exist nonsingular matrices Y and Z such that

$$YAZ = \Lambda_A, \quad YBZ = \Lambda_B,$$

where Λ_A and Λ_B are diagonal matrices. The first MDA appeared in 1960 by Bickley and McNamee [3]. The results of Bickley and McNamee were reviewed by Osborne in 1965 [11]. Their main interest was in a numerical solution to elliptic partial differential equations. They were using a finite difference scheme and wanted to find a direct method for solving the resulting linear systems instead of using successive approximations formally used by Southwell and Thom. Lynch, Rice, and Thomas in [9] claim that another mathematician Egervary was the first to use an MDA in 1960 to analyze the five-point difference approximation of Poisson's equation, but his results were inefficient.

MDAs are typically used with finite difference methods (FDMs), FEMs with piecewise linear quadratic and Hermite cubic basis functions, orthogonal spline collocation (OSC) methods, and nodal cubic spline collocation (NCSC) methods to solve Poisson's equation with Dirichlet boundary conditions on the unit square [2]. Systems and matrices arising from these methods have properties that allow us to find Λ and Z explicitly. In all these cases there are nonsingular Zs and Ys that are made up of sines and cosines [2]. So, the first and third steps in the MDA can be carried out with FFTs. The cost of MDA for FDM, FEM, OSC, and NCSC with FFTs is $\mathcal{O}(N^2 \ln N)$ [2]. We have not used FFTs because we do not have Z in terms of sines and cosines.

An MDA is also applied to a FEM for solving separable elliptic equations on a rectangle in 1984 by Kaufman and Warner [7]. Their algorithm is based on the fact that $Z^T AZ = \Lambda$ and $Z^T BZ = I$, with A and B symmetric and positive definite. Also, Λ and Z are not known explicitly and must be computed. These are the same conditions we have in our MDA, but we are concerned only with Poisson's equation and our problem has the same total cost as theirs, $\mathcal{O}(N^3)$. They extend their results in [8] by creating a program SERRG2 to solve a system arising from a Rayleigh-Ritz-Galerkin method with basis functions that are the tensor products of B-splines. The program allows for high-order discretizations, variable meshes and multiple knots.

ADI methods were first developed by Peaceman and Rachford in 1955 [12]. They were concerned with numerical approximations of the heat equation and Laplace's equation in a square. They found that their procedure for the heat equation was "stable for any size time step" and "required much less work than other methods that had been studied." However, for Laplace's equation, they made no mention of optimum acceleration parameters, like the ones we use. The first sets of optimum acceleration parameter were developed by Wachpress in 1962 [15]. The power of the Peaceman-Rachford scheme, as described in [10], is that it can be exact; however, with the right choice of acceleration parameters, satisfactory results can be obtained with fewer iterations. Lynch, Rice, and Thomas analyze both a MDA and an ADI for Laplace's equation on a square in [9] and [10].

The rest of the thesis is organized as follows. In Section 2 we set up and examine the FEM for the simple one-dimensional 2-point boundary value problem with Dirichlet boundary conditions on $[0, 1]$. We test our problem formulation with a known solution and observe fourth-order convergence as expected. Section 3 deals with the two-dimensional formulation of the method, where we again observe fourth-order convergence. In Section 4 we apply a MDA for solving the large system of linear equations that arises from the two-dimensional problem. The MDA has total cost $\mathcal{O}(N^3)$, where $N + 1$ is the number of subintervals in the x and y directions. To further reduce the cost of solving the linear system in Section 5, we introduce the ADI method whose total cost is $\mathcal{O}(N^2 \ln^2 N)$. The appendices give the details of how we obtained explicit forms for the stiffness and mass matrices.

2 ONE-DIMENSIONAL PROBLEM

2.1 Statement of the Problem

Consider the following elliptic Dirichlet boundary value problem:

$$-u'' + cu = f(x), \quad x \in (0,1) \tag{2.1}$$

$$u(0) = 0, \;\; u(1) = 0, \tag{2.2}$$

where constant $c > 0$. The weak formulation of (2.1) and (2.2) is: Find $u \in H_0^1(0,1)$ such that

$$a(u,v) = l(v), \; v \in H_0^1(0,1), \tag{2.3}$$

where $a(u,v)$ is a symmetric bilinear form and $l(v)$ is a linear functional given by

$$a(u,v) = \int_0^1 u'(x)v'(x)dx + c\int_0^1 u(x)v(x)dx, \tag{2.4}$$

$$l(v) = \int_0^1 f(x)v(x)dx. \tag{2.5}$$

For positive integer N, let $\{x_k\}_{k=0}^{N+1}$ be a uniform partition of $[0,1]$ in the x direction, such that $x_k = kh$, where $h = 1/(N+1)$ is the stepsize. Define the space

$$S_3 = \{v : v \in C^2([0,1]), v\mid_{[x_{k-1},x_k]} \in P_3, 1 \le k \le N+1\},$$

where P_3 is the space of all polynomials of degree ≤ 3. Extend the partition $\{x_k\}_{k=0}^{N+1}$ using $x_k = kh$, $k = -3, -2, -1, N+2, N+3, N+4$. As a basis for S, we choose the B-splines, $\{B_m\}_{m=-1}^{N+2}$, where

$$B_m(x) = \begin{cases} h^{-3} g_1(x - x_{m-2}), & x \in [x_{m-2}, x_{m-1}], \\ g_2(\frac{x - x_{m-1}}{h}), & x \in [x_{m-1}, x_m], \\ g_2(\frac{x_{m+1} - x}{h}), & x \in [x_m, x_{m+1}], \\ h^{-3} g_1(x_{m+2} - x), & x \in [x_{m+1}, x_{m+2}], \\ 0, & \text{otherwise}, \end{cases} \tag{2.6}$$

and $g_1(x) = x^3$, $g_2(x) = 1 + 3x + 3x^2 - 3x^3$. We seek to construct a basis for the space

$$S^D = \{v \in S_3 : v(0) = v(1) = 0\}. \tag{2.7}$$

This basis consists of cubic splines $\left\{B_m^D\right\}_{m=0}^{N+1}$ defined by:

$$
\begin{cases} B_0^D = B_0 - 4B_{-1}, & B_1^D = B_1 - B_{-1}, \\ B_m^D = B_m, & m = 2, ..., N-1 \\ B_N^D = B_N - B_{N+2}, & B_{N+1}^D = B_{N+1} - 4B_{N+2}. \end{cases} \tag{2.8}
$$

Like the cubic splines B_m, the cubic splines B_m^D have support of at most 4 subintervals.

We seek to find an approximation $u_h \in S^D$ such that

$$
a(u_h, v) = l(v), \quad v \in S^D. \tag{2.9}
$$

Since $u_h \in S^D$,

$$
u_h = \sum_{m=0}^{N+1} \alpha_m B_m^D. \tag{2.10}
$$

Substituting (2.10) into (2.9) and choosing $v = B_i^D,\ i = 0, \ldots, N+1$, we obtain

$$
\sum_{m=0}^{N+1} \left(\int_0^1 [B_i^D]'(x)[B_m^D]'(x)dx + c\int_0^1 B_i^D(x)B_m^D(x)dx \right) \alpha_m = F_i \quad i = 0, \ldots, N+1 \tag{2.11}
$$

where

$$
F_i = \int_0^1 f(x)B_i^D(x)dx. \tag{2.12}
$$

Define the following matrices:

$$
A = (a_{i,m})_{i,m=0}^{N+1}, \;\; a_{i,m} = \int_0^1 \left[B_i^D\right]'(x)\left[B_m^D\right]'(x)dx, \tag{2.13}
$$

$$
B = (b_{i,m})_{i,m=0}^{N+1}, \;\; b_{i,m} = \int_0^1 B_i^D(x)B_m^D(x)dx. \tag{2.14}
$$

Equation (2.11) is now a $(N+2) \times (N+2)$ linear system

$$
(A + cB)\,\boldsymbol{\alpha} = \mathbf{F}, \tag{2.15}
$$

where

$$
\boldsymbol{\alpha} = [\alpha_0, \ldots, \alpha_{N+1}]^T \tag{2.16}
$$

is a vector of unknown coefficients and

$$
\mathbf{F} = [F_0, \ldots, F_{N+1}]^T. \tag{2.17}
$$

Each F_i is approximated using a composite 3-point Guass quadrature rule on $[0, 1]$ with the partition

$\{x_k\}_{k=0}^{N+1}$. This quadrature rule yields an exact result for definite integrals of polynomials of degree ≤ 5 and is derived as follows [4].

The basic 3-point Gauss quadrature is defined by

$$\int_{-1}^{1} r(t)dt \approx \sum_{p=1}^{3} w_p r(\xi_p), \tag{2.18}$$

where the nodes ξ_p and weights w_p are

$$\xi_1 = -\sqrt{\frac{3}{5}}, \quad \xi_2 = 0, \quad \xi_3 = \sqrt{\frac{3}{5}}, \tag{2.19}$$

$$w_1 = \frac{5}{9}, \quad w_2 = \frac{8}{9}, \quad w_3 = \frac{5}{9}. \tag{2.20}$$

For $g(x)$ on $[0,1]$, with substitution

$$x = x_k + \frac{h}{2}(1+t), \quad t \in [-1,1], \tag{2.21}$$

using (2.18) we have

$$\begin{aligned} \int_0^1 g(x)dx = \sum_{k=0}^{N} \int_{x_k}^{x_{k+1}} g(x)dx = \frac{h}{2} \sum_{k=0}^{N} \int_{-1}^{1} g\left(x_k + \frac{h}{2}(1+t)\right) dt \\ \approx \frac{h}{2} \sum_{k=0}^{N} \sum_{p=1}^{3} w_p g(\xi_{k,p}) \end{aligned} \tag{2.22}$$

where

$$\xi_{k,p} = \frac{h}{2}\xi_p + \frac{x_k + x_{k+1}}{2}, \quad k = 0, \ldots, N, \quad p = 1,2,3. \tag{2.23}$$

Using (2.23) and (2.19), we have

$$\xi_{k,1} = x_k + \frac{1-\sqrt{3/5}}{2}h, \quad \xi_{k,2} = x_k + \frac{h}{2}, \quad \xi_{k,3} = x_k + \frac{1+\sqrt{3/5}}{2}h, \quad k = 0, \ldots, N. \tag{2.24}$$

Using (2.22), the F_i given in (2.12) can be approximated by

$$F_i \approx \frac{h}{2} \sum_{k=0}^{N} \sum_{p=1}^{3} w_p f(\xi_{k,p}) B_i^D(\xi_{k,p}), \quad i = 0, \ldots, N+1. \tag{2.25}$$

We can explicitly write the entries of the matrices A and B in (2.13), (2.14). Derivation of entries may be found in Appendix A and Appendix B. The stiffness matrix is

$$A = \frac{3}{10h}\begin{bmatrix} 80 & 43 & -20 & -1 & 0 & \cdots & \cdots & \cdots & 0 \\ 43 & 104 & -14 & -24 & -1 & \ddots & & & \vdots \\ -20 & -14 & 80 & -15 & -24 & -1 & \ddots & & \vdots \\ -1 & -24 & -15 & 80 & -15 & -24 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & -24 & -15 & 80 & -15 & -24 & -1 \\ \vdots & & \ddots & -1 & -24 & -15 & 80 & -14 & -20 \\ \vdots & & & \ddots & -1 & -24 & -14 & 104 & 43 \\ 0 & \cdots & \cdots & \cdots & 0 & -1 & -20 & 43 & 80 \end{bmatrix}, \tag{2.26}$$

and the mass matrix is

$$B = \frac{h}{140}\begin{bmatrix} 496 & 773 & 116 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 773 & 2296 & 1190 & 120 & 1 & \ddots & & & \vdots \\ 116 & 1190 & 2416 & 1191 & 120 & 1 & \ddots & & \vdots \\ 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 \\ \vdots & & \ddots & 1 & 120 & 1191 & 2416 & 1190 & 116 \\ \vdots & & & \ddots & 1 & 120 & 1190 & 2296 & 773 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & 116 & 773 & 496 \end{bmatrix}. \tag{2.27}$$

2.2 Rate of Convergence

The error in the maximum norm at all x_k is defined by:

$$e_N = \max_{0\le k\le N+1} |u(x_k) - u_h(x_k)|. \tag{2.28}$$

If we assume $e_N = ch^p$, where positive c is independent of h, then

$$\frac{e_N}{e_{2N}} = \frac{c\left(\frac{1}{N+1}\right)^p}{c\left(\frac{1}{2N+1}\right)^p} = \left(\frac{2N+2-1}{N+1}\right)^p = \left(2 - \frac{1}{N+1}\right)^p.$$

Thus we estimate the rate of convergence p of the method using the formula

$$p = \frac{\log \frac{e_N}{e_{2N}}}{\log(2 - \frac{1}{N+1})} \ . \tag{2.29}$$

2.3 Numerical Results

For a specific case of (2.1) and (2.2), we assume $c = 2$ and a solution is $u = x(x-1)e^x$. Then, $f(x) = x^2e^x - 5xe^x$. The entries of the right-hand side vector $\mathbf{F}$ were computed using (2.25). We solve system (2.15) and obtain u_h of (2.10), which gives

$$u_h(x_k) = \sum_{m=0}^{N+1} B_m^D(x_k)\alpha_m, \quad k = 0, \ldots, N+1.$$

Hence for

$$\mathbf{u}_h = [u_h(x_0), \ldots, u_h(x_N)]^T \tag{2.30}$$

we have

$$\mathbf{u}_h = C\boldsymbol{\alpha},$$

where $\boldsymbol{\alpha}$ is as in (2.15) and

$$C = (c_{km})_{k,m=0}^{N+1}, \quad c_{km} = B_m^D(x_k). \tag{2.31}$$

The basis functions are such that

$$B_{k-1}(x_k) = 1, \quad B_k(x_k) = 4, \quad B_{k+1}(x_k) = 1. \tag{2.32}$$

Using (2.8) and (2.32), we have

$$C = \begin{bmatrix} 0 & \cdots & & & & & \cdots & 0 \\ 1 & 4 & 1 & 0 & \cdots & & & \vdots \\ 0 & 1 & 4 & 1 & 0 & & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & & 0 \\ \vdots & & & 0 & 1 & 4 & 1 & 0 \\ 0 & & & & 0 & 1 & 4 & 1 \\ 0 & \cdots & & & & & \cdots & 0 \end{bmatrix}. \tag{2.33}$$

In Table 1, for different values of N, we present the following errors and estimated rates of convergence.

Table 1: Errors e_N and convergence rates p for 1D test problem

N	4	8	16	32	64
e_N	6.96–06	6.44–07	4.96–08	3.45–09	2.27–10
p		3.7428	3.8670	3.933	3.9662

We observe fourth-order convergence as expected.

3 TWO-DIMENSIONAL PROBLEM

3.1 Statement of the Problem

Consider the following two-dimensional, elliptic problem

$$-\Delta u(x,y) = f(x,y),\ (x,y) \in \Omega = (0,1) \times (0,1), \tag{3.1}$$

with zero Dirichlet boundary conditions

$$u(x,0) = u(x,1) = 0,\ x \in [0,1], \quad u(0,y) = u(1,y) = 0,\ y \in [0,1]. \tag{3.2}$$

The weak formulation of (3.1) and (3.2) is: Find $u \in H_0^1(\Omega)$ such that

$$a(u,v) = l(v), \ \ v \in H_0^1(\Omega), \tag{3.3}$$

where $a(u,v)$ is a symmetric, bilinear form and $l(v)$ is a linear functional given by

$$a(u,v) = \iint_\Omega \bigtriangledown u(x,y) \cdot \bigtriangledown v(x,y) dxdy,$$

$$l(v) = \iint_\Omega f(x,y) v(x,y) dxdy.$$

With the space S^D defined in (2.7), we seek an approximate solution $u_h \in S^D \otimes S^D$ such that

$$a(u_h, v) = l(v),\ v \in S^D \otimes S^D. \tag{3.4}$$

where $\otimes$ denotes the tensor product of vector spaces[1].

For positive integer, N, let $\{x_k\}_{k=0}^{N+1}$ be a uniform partition of $[0,1]$ in the x direction, such that $x_k = kh$, where $h = 1/(N+1)$ is the stepsize. Similarly, in the y direction, let $\{y_l\}_{l=0}^{N+1}$ be a uniform partition of $[0,1]$, such that $y_l = lh$. The approximate solution $u_h \in S^D \otimes S^D$ is of the form

$$u_h(x,y) = \sum_{m=0}^{N+1} \sum_{n=0}^{N+1} \alpha_{m,n} B_m^D(x) B_n^D(y). \tag{3.5}$$

Substituting (3.5) in (3.4) and choosing

$$v(x,y) = B_i^D(x) B_j^D(y),\ i,j = 0, \ldots, N+1,$$

[1]For two spaces V and W of functions, $V \otimes W$ denotes the space of functions consisting of all finite linear combinations of products $\psi_1(x)\psi_2(y)$ with $\psi_1 \in V$ and $\psi_2 \in W$.

with $B_i^D(x)$ and $B_j^D(y)$ as in (2.8), we obtain

$$\sum_{m=0}^{N+1}\sum_{n=0}^{N+1}\Big[\iint_{\Omega}[B_m^D]'(x)B_n^D(y)[B_i^D]'(x)B_j^D(y)+B_m^D(x)[B_n^D]'(y)B_i^D(x)[B_j^D]'(y)\Big]\alpha_{m,n}dxdy$$
$$=F_{i,j},\ i,j=0,\dots,N+1,$$

where

$$F_{i,j}=\int_0^1\int_0^1 f(x,y)B_i^D(x)B_j^D(y)dxdy \quad i,j=0,1,\dots,N+1. \tag{3.6}$$

Expressing double integrals in terms of iterated integrals, we get

$$\begin{aligned}\sum_{m=0}^{N+1}\sum_{n=0}^{N+1}\Big[\int_0^1[B_m^D]'(x)[B_i^D]'(x)dx\int_0^1 B_n^D(y)B_j^D(y)dy+\int_0^1 B_m^D(x)B_i^D(x)dx\\ \int_0^1[B_n^D]'(y)[B_j^D]'(y)dy\Big]\alpha_{m,n}=F_{i,j},\quad i,j=0,\dots,N+1,\quad m,n=0,\dots,N+1.\end{aligned} \tag{3.7}$$

Using (2.13) and (2.14) in (3.7), we have

$$\sum_{m=0}^{N+1}a_{i,m}\sum_{n=0}^{N+1}b_{j,m}\alpha_{m,n}\ +\ \sum_{m=0}^{N+1}b_{i,m}\sum_{n=0}^{N+1}a_{j,n}\alpha_{m,n}=F_{i,j},\quad i,j=0,\dots,N+1. \tag{3.8}$$

Suppose $\mathcal{I},\mathcal{J},\mathcal{M},\mathcal{N}$ are finite sets of increasing indices. We assume that

$$\mathcal{I}=\{0,\dots,I'\},\quad \mathcal{J}=\{0,\dots,J'\},\quad \mathcal{M}=\{0,\dots,M'\}\quad \mathcal{N}=\{0,\dots,N'\}.$$

Then the matrix form of

$$\phi_{i,j}=\sum_{m\in\mathcal{M}}c_{i,m}^{(1)}\sum_{n\in\mathcal{N}}c_{j,n}^{(2)}\psi_{m,n},\ i\in\mathcal{I},\ j\in\mathcal{J}, \tag{3.9}$$

is

$$\mathbf{\Phi}=(C_1\otimes C_2)\mathbf{\Psi}, \tag{3.10}$$

where $\otimes$ in this case denotes the tensor product of matrices[2] and

$$C_1=\Big[c_{i,m}^{(1)}\Big]_{i\in\mathcal{I},m\in\mathcal{M}},C_2=\Big[c_{j,n}^{(2)}\Big]_{j\in\mathcal{J},n\in\mathcal{N}}, \tag{3.11}$$

and

$$\mathbf{\Phi}=\Big[\phi_{0,0},\dots,\phi_{0,M'},\dots,\phi_{I',0},\dots,\phi_{I',J'}\Big]^T,$$
$$\mathbf{\Psi}=\Big[\psi_{0,0},\dots,\psi_{0,N'},\dots,\psi_{M',0},\dots,\psi_{M',N'}\Big]^T.$$

[2]If the matrix $A=(a_{i,j})$ is $M_A\times N_A$ and B is $M_B\times N_B$, then the matrix $A\otimes B$ is the $M_AM_B\times N_AN_B$ block matrix whose (i,j) block is $a_{i,j}B$.

Thus, using this result [2], (3.8) can be written as an $(N+2)^2 \times (N+2)^2$ linear system

$$\Big(A \otimes B + B \otimes A\Big)\boldsymbol{\alpha} = \mathbf{F} \tag{3.12}$$

where A and B are as in (2.13) and (2.14),

$$\boldsymbol{\alpha} = \Big[\alpha_{0,0}, \dots, \alpha_{0,N+1}, \alpha_{1,0}, \dots, \alpha_{1,N+1}, \dots, \alpha_{N+1,0} \dots, \alpha_{N+1,N+1}\Big]^T \tag{3.13}$$

is a vector of unknown coefficients and

$$\mathbf{F} = \Big[F_{0,0}, \dots, F_{0,N+1}, F_{1,0}, \dots, F_{1,N+1}, \dots, F_{N+1,0} \dots, F_{N+1,N+1}\Big]^T. \tag{3.14}$$

Using (2.22) with respect to x in (3.6), we get

$$\begin{aligned} F_{i,j} &= \int_0^1 \Big[\int_0^1 f(x,y)B_i^D(x)dx\Big]B_j^D(y)dy \\ &\approx \frac{h}{2}\sum_{k=0}^{N}\sum_{p=0}^{3} w_p B_i^D(\xi_k)\int_0^1 f(\xi_{k,p},y)B_j^D(y)dy. \end{aligned}$$

Then using (2.22) with respect to y, we get

$$\begin{aligned} F_{i,j} &\approx \frac{h}{2}\sum_{k=0}^{N}\sum_{p=1}^{3} w_p B_i^D(\xi_{k,p})\frac{h}{2}\sum_{l=0}^{N}\sum_{q=1}^{3}\big[w_q f(\xi_{k,p},\xi_{l,q})B_j^D(\xi_{l,q})\big] \\ &= \left(\frac{h}{2}\right)^2 \sum_{k=0}^{N}\sum_{l=0}^{N}\sum_{p=1}^{3}\sum_{q=1}^{3} w_p w_q f(\xi_{k,p},\xi_{l,q})B_i^D(\xi_{k,p})B_j^D(\xi_{l,q}) \end{aligned}$$

for $i,j = 0,\dots,N+1$.

3.2 Numerical Results

Consider a specific case of the two-dimensional problem (3.1)-(3.2) with the solution $u(x,y) = x(x-1)e^x y(y-1)e^y$. Thus,

$$f(x,y) = -2xye^{x+y}(xy + x + y - 3). \tag{3.15}$$

We solve (3.12) to obtain u_h in (3.5) which gives

$$u_h(x_k,y_l) = \sum_{m=0}^{N+1} B_m^D(x_k) \sum_{n=0}^{N+1} B_n^D(y_l)\alpha_{m,n}, \quad k,l = 0,\dots,N+1. \tag{3.16}$$

Applying (3.10) and (3.11) to (3.16) we obtain

$$\mathbf{u}_h = (C \otimes C)\,\boldsymbol{\alpha}, \tag{3.17}$$

where C is defined in (2.31), $\boldsymbol{\alpha}$ is as in (3.13), and

$$\mathbf{u}_h = \left[\; u_h(x_0, y_0), \ldots, u_h(x_0, y_{N+1}), \ldots, u_h(x_{N+1}, y_0), \ldots, u_h(x_{N+1}, y_{N+1}) \;\right]^T. \tag{3.18}$$

The error in the discrete maximum norm at all x_k, y_l is defined by

$$e_N = \max_{\substack{0 \le k \le N+1 \\ 0 \le l \le N+1}} |u(x_k, y_l) - u_h(x_k, y_l)|. \tag{3.19}$$

In Table 2, for different values of N, we present the errors and estimated convergence rates, p of (2.29).

Table 2: Errors e_N and convergence rates p for 2D test problem

N	4	8	16	32
e_N	3.21–04	3.77–05	3.47–06	2.81–07
p		3.6400	3.7523	3.7907

Again, fourth-order convergence is observed.

4 MATRIX DECOMPOSITION ALGORITHM

4.1 Explanation of the Method

Matrix decomposition algorithms have been proposed to produce efficient solutions to systems of linear algebraic equations of the form

$$(A_1 \otimes B_2 + B_1 \otimes A_2)\boldsymbol{\alpha} = \mathbf{f}, \tag{4.1}$$

where A_1, B_1 are $M_1 \times M_1$ matrices and A_2, B_2 are $M_2 \times M_2$ matrices. These MDAs provide a direct method for solving (4.1) as a set of independent one-dimensional problems rather than one large costly system. As a result of our two-dimensional finite element method formulation, we see that (3.12) is a system of linear algebraic equations of the same form as (4.1) with $A_1 = A_2 = A$ as in (2.13) and $B_1 = B_2 = B$ as in (2.14).

Since B is a symmetric and positive definite matrix, B has a Cholesky factorization,

$$B = R^T R \tag{4.2}$$

where R is an upper triangular matrix. Since A is a symmetric matrix, so is

$$\widetilde{A} = R^{-T} A R^{-1}. \tag{4.3}$$

Any real symmetric matrix is orthogonally similar to a real diagonal matrix, so we have

$$V^T \widetilde{A} V = \Lambda \tag{4.4}$$

where Λ is a real diagonal matrix and V is an orthogonal matrix, that is, V is a real matrix and

$$V V^T = I, \quad V^T V = I. \tag{4.5}$$

Substituting (4.3) into (4.4), we get

$$V^T R^{-T} A R^{-1} V = \Lambda. \tag{4.6}$$

Multiplying (4.6) on the left by $R^T V$ and using (4.5) and (4.2), we obtain

$$AZ = BZ\Lambda \tag{4.7}$$

where the real matrix

$$Z = R^{-1} V. \tag{4.8}$$

It follows from (4.2), (4.8), and (4.5) that

$$Z^T B Z = V^T R^{-T} R^T R R^{-1} V = V^T V = I. \tag{4.9}$$

Multiplying (4.7) by Z^T and using (4.9), we have

$$Z^T A Z = Z^T B Z \Lambda = \Lambda. \tag{4.10}$$

(4.9) implies that Z is nonsingular. Hence system (3.12) is equivalent to

$$(Z^T \otimes I)(A \otimes B + B \otimes A)(Z \otimes I)(Z^{-1} \otimes I)\boldsymbol{\alpha} = (Z^T \otimes I)\mathbf{F}, \tag{4.11}$$

where we used a property of matrix tensor product,

$$(P \otimes R)(S \otimes T) = (PS) \otimes (RT), \tag{4.12}$$

and $\boldsymbol{\alpha}$ and $\mathbf{F}$ are as in (3.13) and (3.14), respectively. Using again (4.12) in (4.11), we obtain

$$[(Z^T A Z) \otimes B + (Z^T B Z) \otimes A]\mathbf{v} = \mathbf{g}, \tag{4.13}$$

where

$$\mathbf{v} = (Z^{-1} \otimes I)\boldsymbol{\alpha}, \quad \mathbf{g} = (Z^T \otimes I)\mathbf{F} \tag{4.14}$$

and the vectors $\mathbf{v}$ and $\mathbf{g}$ have the form

$$\boldsymbol{v} = \Big[v_{0,0}, \ldots, v_{0,N+1}, v_{1,0}, \ldots, v_{1,N+1}, \ldots, v_{N+1,0} \ldots, v_{N+1,N+1}\Big]^T,$$

$$\boldsymbol{g} = \Big[g_{0,0}, \ldots, g_{0,N+1}, g_{1,0}, \ldots, g_{1,N+1}, \ldots, g_{N+1,0} \ldots, g_{N+1,N+1}\Big]^T.$$

Using (4.13), (4.10) and (4.9), we obtain

$$(\Lambda \otimes B + I \otimes A)\mathbf{v} = \mathbf{g}. \tag{4.15}$$

Since $\Lambda = \text{diag}(\lambda_i)_{i=0}^{N+1}$, (4.15) reduces to the solution of $N+2$ independent systems

$$(\lambda_i B + A)\mathbf{v}_i = \mathbf{g}_i, \;\; i = 0, \ldots, N+1, \tag{4.16}$$

where

$$\mathbf{v}_i = \Big[v_{i,0}, \ldots, v_{i,N+1}\Big], \quad \mathbf{g}_i = \Big[g_{i,0}, \ldots, g_{i,N+1}\Big], \; i = 0, \ldots, N+1.$$

Thus, we have the following algorithm for solving (3.12)

MDA

Step 1: Compute $\mathbf{g} = (Z^T \otimes I)\mathbf{F}$

Step 2: Solve $(\lambda_i B + A)\mathbf{v}_i = \mathbf{g}_i, \; i = 0, \ldots, N+1$

Step 3: Compute $\boldsymbol{\alpha} = (Z \otimes I)\mathbf{v}$.

Steps 1 and 3 can be computed more effeciently taking advantage of the block structure of $(Z^T \otimes I)$ and $(Z \otimes I)$.

Assume $Z = (z_{i,j})_{i,j=0}^{N+1}$ and consider Step 3,

$$\boldsymbol{\alpha} = (Z \otimes I)\mathbf{v}, \tag{4.17}$$

where

$$Z \otimes I = \left[\begin{array}{c|c|c|c} z_{0,0}I & z_{0,1}I & \dots & z_{0,N+1}I \\ \hline z_{1,0}I & z_{1,1}I & \dots & z_{1,N+1}I \\ \hline \vdots & \vdots & \dots & \vdots \\ \hline z_{N+1,0}I & z_{N+1,1}I & \dots & z_{N+1,N+1}I \end{array}\right]. \tag{4.18}$$

Using (4.17), (4.18), (3.13), for $j = 0, \dots, N+1$, we have

$$\begin{bmatrix} \alpha_{0,j} \\ \alpha_{1,j} \\ \vdots \\ \alpha_{N+1,j} \end{bmatrix} = \begin{bmatrix} z_{0,0} & z_{0,1} & \cdots & z_{0,N+1} \\ z_{1,0} & z_{1,1} & \cdots & z_{1,N+1} \\ \vdots & \vdots & & \vdots \\ z_{N+1,0} & z_{N+1,1} & \cdots & z_{N+1,N+1} \end{bmatrix} \begin{bmatrix} v_{0,j} \\ v_{1,j} \\ \vdots \\ v_{N+1,j} \end{bmatrix}. \tag{4.19}$$

Hence Step 3 becomes:

For $j = 0, \dots, N+1$, compute $\tilde{\boldsymbol{\alpha}}_j = Z\tilde{\mathbf{v}}_j$ where

$$\tilde{\boldsymbol{\alpha}}_j = \left[\alpha_{0,j}, \dots, \alpha_{N+1,j}\right]^T, \quad \tilde{\mathbf{v}}_j = \left[v_{0,j}, \dots, v_{N+1,j}\right]^T.$$

Step 1 can be carried out in a similar way because $Z^T \otimes I$ has the same structure as $Z \otimes I$. So Step 1 becomes,

For $j = 0, \dots, N+1$, compute $\tilde{\mathbf{g}}_j = Z^T\tilde{\mathbf{F}}_j$, where

$$\tilde{\mathbf{g}}_j = \left[g_{0,j}, \dots, g_{N+1,j}\right]^T, \quad \tilde{\mathbf{F}}_j = \left[F_{0,j}, \dots, F_{N+1,j}\right]^T.$$

4.2 Cost Analysis

The cost of an algorithm is the number of required arithmetic operations. To determine Z and Λ, the Crawford algorithm can be used at a cost of $\mathcal{O}(N^2)$ [5]. Step 1 and Step 3 require $N+2$ multiplications of Z^T with $\tilde{\mathbf{F}}$ and Z with $\tilde{\mathbf{v}}$, respectively. Thus, Step 1 and Step 3 can be performed at a cost of $\mathcal{O}(N^3)$. Step 2 is comprised of $N+2$ independent systems that can each be solved, assuming that we use Gauss elimination without pivoting within the heptadiagonal band, at a cost proportional to the number of unknowns, $\mathcal{O}(N)$. We need to compute those solutions $N+2$ times, so the total cost of Step 2 is $\mathcal{O}(N^2)$. Thus the total cost of MDA is $\mathcal{O}(N^3)$.

4.3 Numerical Results

Consider the same specific case of the two-dimensional problem (3.1)-(3.2) with the symmetric $u(x,y) = x(x-1)e^x y(y-1)e^y$ that we used in the general formulation of the two-dimensional problem with (3.15). We now solve (3.12) using the MDA to obtain $\boldsymbol{\alpha}$ which gives $\mathbf{u_h}$ of (3.17).

The error in the discrete maximum norm at all (x_k, y_l) is defined in (3.19). In Table 3, for different values of N, we present the errors, e_N of (3.19) and estimated convergence rates, p of (2.29):

Table 3: Errors e_N and convergence rates p using MDA

N	4	8	16	32	64	128
e_N	4.73–05	5.26–06	3.91–07	2.72–08	1.81-09	0.00-10
p		3.7373	4.0858	4.0213	3.9994	3.9984

We notice that the numerical results using the MDA are almost the same as when we used Gauss elimination without pivoting to solve the resulting linear systems. The rate of convergence is approaching 4 as we expect. Because the cost of MDA is significantly less than that of Gauss elimination, we can increase the number of subintervals and in turn improve our rate of convergence.

We also considered a non-separable test problem, $u(x,y) = x(x-1)y(y-1)e^{xy}$ and in Table 4, for different values of N, we present the errors e_N of (3.19) and estimated convergence rates p of (2.29) where we again observe fourth order convergence.

Table 4: Errors e_N and convergence rates p using MDA, non-separable u

N	4	8	16	32	64
e_N	1.06–05	1.20–06	9.25–08	6.39–09	4.25–10
p		3.7050	4.0253	4.0298	3.9997

We also computed the maximum error norm defined

$$e_M = \max_{(x,y)\in\Omega} |u(x,y) - u_h(x,y)|, \tag{4.20}$$

where we approximated the norm using 201 equally spaced points in the x and y directions, for non-separable $u(x,y)$. In Table 5, for different values of N, we present the maximum errors e_M of (4.20) and convergence rates p of (2.29), and we again observe fourth order convergence.

Table 5: Errors e_M and convergence rates p using MDA, non-separable u

N	4	8	16	32	64
e_N	1.06–05	1.24–06	9.16–08	5.85–09	4.18–10
p		3.6450	4.0965	4.1474	3.8943

5 ALTERNATING DIRECTION IMPLICIT METHOD

5.1 Explanation of the Method

The Alternating Direction Implicit (ADI) method, developed by Peaceman and Rachford [12], is a technique for solving the linear systems that arise from using finite difference, finite element, and collocation methods to solve elliptic partial differential equations.

Consider the system

$$(C \otimes I + I \otimes C)\mathbf{v} = \mathbf{g}. \tag{5.1}$$

Note that $C \otimes I$ and $I \otimes C$ commute. If C is a symmetric and positive definite matrix, then the ADI method for solving (5.1) is as follows [13]:

$$[(C + \omega_{k+1} I) \otimes I]\mathbf{v}^{(k+\frac{1}{2})} = \mathbf{g} - [I \otimes (C - \omega_{k+1} I)]\mathbf{v}^{(k)}, \tag{5.2}$$

$$[I \otimes (C + \omega_{k+1} I)]\mathbf{v}^{(k+1)} = \mathbf{g} - [(C - \omega_{k+1} I) \otimes I]\mathbf{v}^{(k+\frac{1}{2})}, \tag{5.3}$$

for $k = 0, 1, \ldots$ where $\mathbf{v}^{(0)}$ is an initial approximation (typically $\mathbf{v}^{(0)} = \mathbf{0}$) and ω_{k+1} is a Jordan acceleration parameter. To solve (3.12) using the ADI method, we first put (3.12) in the form (5.1). Multiplying (3.12) on the left by $B^{-1/2} \otimes B^{-1/2}$ and recognizing by (4.12) that $I = (B^{-1/2} \otimes B^{-1/2})(B^{1/2} \otimes B^{1/2})$, we obtain

$$(B^{-1/2} \otimes B^{-1/2})(A \otimes B + B \otimes A)(B^{-1/2} \otimes B^{-1/2})(B^{1/2} \otimes B^{1/2})\boldsymbol{\alpha} = (B^{-1/2} \otimes B^{-1/2})\mathbf{f}. \tag{5.4}$$

It follows from the tensor product property (4.12) that

$$\begin{aligned}
&(B^{-1/2} \otimes B^{-1/2})(A \otimes B + B \otimes A)(B^{-1/2} \otimes B^{-1/2}) \\
&= (B^{-1/2} A B^{-1/2}) \otimes (B^{-1/2} B B^{-1/2}) + (B^{-1/2} B B^{-1/2}) \otimes (B^{-1/2} A B^{-1/2}) \\
&= C \otimes I + I \otimes C,
\end{aligned} \tag{5.5}$$

where

$$C = B^{-1/2} A B^{-1/2}. \tag{5.6}$$

Introducing

$$\mathbf{v} = (B^{1/2} \otimes B^{1/2})\boldsymbol{\alpha}, \tag{5.7}$$

$$\mathbf{g} = (B^{-1/2} \otimes B^{-1/2})\mathbf{f}, \tag{5.8}$$

and using (5.5), we see that (5.4) becomes (5.1).

In order to avoid having to compute C of (5.6), we want to show that (5.2) and (5.3) can be written in terms of A and B. We introduce

$$(B^{1/2} \otimes B^{1/2})\boldsymbol{\alpha}^{(k)} = \mathbf{v}^{(k)}, \tag{5.9}$$

$$(B^{1/2} \otimes B^{1/2})\boldsymbol{\alpha}^{(k+\frac{1}{2})} = \mathbf{v}^{(k+\frac{1}{2})}, \tag{5.10}$$

and

$$(B^{1/2} \otimes B^{1/2})\boldsymbol{\alpha}^{(k+1)} = \mathbf{v}^{(k+1)}. \tag{5.11}$$

Multiplying (5.2) on the left by $B^{1/2} \otimes B^{1/2}$ and using (5.10), (5.9), (5.8) the tensor product property (4.12), and (5.6), we obtain

$$\begin{aligned}
&(B^{1/2} \otimes B^{1/2})[(C + \omega_{k+1}I) \otimes I](B^{1/2} \otimes B^{1/2})\boldsymbol{\alpha}^{(k+\frac{1}{2})} \\
&\qquad = (B^{1/2} \otimes B^{1/2}) \left[(B^{-1/2} \otimes B^{-1/2})\mathbf{f} - [I \otimes (C - \omega_{k+1}I)]\right] (B^{1/2} \otimes B^{1/2})\boldsymbol{\alpha}^{(k)}, \\
&(B^{1/2}CB^{1/2} + \omega_{k+1}B^{1/2}B^{1/2}) \otimes (B^{1/2}B^{1/2})]\boldsymbol{\alpha}^{(k+\frac{1}{2})} \\
&\qquad = \mathbf{f} - [(B^{1/2}B^{1/2}) \otimes (B^{1/2}CB^{1/2} - \omega_{k+1}B^{1/2}B^{1/2})]\boldsymbol{\alpha}^{(k)}, \\
&[(A + \omega_{k+1}^{(1)}B) \otimes B]\boldsymbol{\alpha}^{(k+\frac{1}{2})} = \mathbf{f} - [B \otimes (A - \omega_{k+1}B)]\boldsymbol{\alpha}^{(k)}, \quad k = 0, 1, \ldots.
\end{aligned} \tag{5.12}$$

Clearly, (5.12) is entirely in terms of A and B. Similarly, multiplying (5.3) on the left by $B^{1/2} \otimes B^{1/2}$ and using (5.10), (5.11), (5.8), (5.6), the tensor product property (4.12), and (5.6), we obtain

$$\begin{aligned}
&(B^{1/2} \otimes B^{1/2})[I \otimes (C + \omega_{k+1}I)](B^{1/2} \otimes B^{1/2})\boldsymbol{\alpha}^{(k+1)} \\
&\qquad = (B^{1/2} \otimes B^{1/2}) \left[(B^{-1/2} \otimes B^{-1/2})\mathbf{f} - [(C - \omega_{k+1}I) \otimes I]\right] (B^{1/2} \otimes B^{1/2})\boldsymbol{\alpha}^{(k+\frac{1}{2})}, \\
&(B^{1/2}B^{1/2}) \otimes [(B^{1/2}CB^{1/2} + \omega_{k+1}B^{1/2}B^{1/2})]\boldsymbol{\alpha}^{(k+1)} \\
&\qquad = \mathbf{f} - [(B^{1/2}CB^{1/2} - \omega_{k+1}B^{1/2}B^{1/2}) \otimes (B^{1/2}B^{1/2})]\boldsymbol{\alpha}^{(k+\frac{1}{2})}, \\
&[B \otimes (A + \omega_{k+1}B)]\boldsymbol{\alpha}^{(k+1)} = \mathbf{f} - [(A - \omega_{k+1}B) \otimes B]\boldsymbol{\alpha}^{(k+\frac{1}{2})}, \quad k = 0, 1, \ldots.
\end{aligned} \tag{5.13}$$

Clearly, (5.13) is in terms of only A and B. We now have an iterative approach (5.12)-(5.13) for solving (3.12).

Because our itereation parameter requires the largest and smallest eigenvalue of C, but now we have our iterative approach in terms of A and B, we need to show that the eigenvalues of C are the same as those from the generalized eigenvalue problem for A and B. If any λ is an eigenvalue of C, then

$$C\mathbf{x} = \lambda\mathbf{x}, \quad \mathbf{x} \neq 0. \tag{5.14}$$

Using (5.6) and multiplying (5.14) on the left by $B^{1/2}$ gives,

$$AB^{-1/2}\mathbf{x} = \lambda B^{1/2}\mathbf{x}. \tag{5.15}$$

Take $\mathbf{y} = B^{-1/2}\mathbf{x}$, equivalently $\mathbf{x} = B^{1/2}\mathbf{y}$, and then (5.15) becomes

$$A\mathbf{y} = \lambda B\mathbf{y}. \tag{5.16}$$

Thus λ is an eigenvalue of C if and only if λ solves the generalized eigenvalue problem (5.16). Definitions of our iteration parameter requires the use of the quantities $\delta_1,\ \delta_2,\ \Delta_1,\ \Delta_2$ [13]. For our particular problem, $\delta_1 = \delta_2 = \delta$ and $\Delta_1 = \Delta_2 = \Delta$, where Δ and δ are the largest and smallest eigenvalues from the generalized eigenvalue problem, (5.16). We will need the following constant to find our iterative parameters

$$\eta = \frac{\delta}{\Delta}, \quad \eta \in [0, 1]. \tag{5.17}$$

Iterating n times will achieve accuracy ϵ if

$$n \geq n_0(\epsilon) = \frac{1}{\pi^2} \ \ln \frac{4}{\eta} \ \ln \frac{4}{\epsilon}. \tag{5.18}$$

It is natural to take

$$\epsilon = h^5.$$

Numerical results indicate that $\eta = \mathcal{O}(h^2)$ which yields $n_0(\epsilon)$ of the form $\beta \ln^2 N + \gamma$, where real numbers β, γ are independent of N. For simplicity, we take

$$n_0 = \ln^2 N + 4.$$

The parameter, ω_{k+1} is chosen using the formula

$$\omega_i = \Delta \mu_i, \quad i = 1, 2, \ldots, n. \tag{5.19}$$

where

$$\mu_i = dn \left(\frac{2i-1}{2n} K'(\eta), \eta' \right), \quad i = 1, 2, \ldots, n, \tag{5.20}$$

and dn denotes the Jacobi elliptic function. We use the following approximations for μ_i given in section 11.1.4 of [13]:

$$\mu_i = \begin{cases} \sqrt{\eta} q^{\frac{2\sigma_i - 1}{4}} \frac{1+q^{1-\sigma_i}+q^{1+\sigma_i}}{1+q^{\sigma_i}+q^{2-\sigma_i}}, & [n/2]+1 \leq i \leq n, \\ \eta/\mu_{n+1-i}, & 1 \leq i \leq [n/2], \end{cases} \tag{5.21}$$

$$\sigma_i = \frac{2i-1}{2n}, \qquad q = \eta^2 \frac{1+\eta^2/2}{16}.$$

Derivation and analysis of (5.21) is given in section 11.1.4 of [13].

5.2 Cost Analysis

Since $B \otimes (A - \omega_{k+1} B) = (B \otimes I)(I \otimes (A - \omega_{k+1} B))$ and $(A + \omega_{k+1} B) \otimes B = ((A + \omega_{k+1} B) \otimes I)(I \otimes B)$, the first half of the ADI step (5.12) requires multiplying $N + 2$ times by $A - \omega_{k+1} B$ and then multiplying $N + 2$ times by B. Then, we solve $N + 2$ linear systems with $A + \omega_{k+1} B$ and $N + 2$ with B. Similarly, the second half of the ADI step (5.13) involves $N + 2$ multiplications by B and then again by $A - \omega_{k+1} B$ and then requires solving $N + 2$ linear systems, first with B and then with $A + \omega_{k+1} B$. All the linear systems are solved using Gauss elimination without pivoting in the band with cost $\mathcal{O}(N^2)$. So the total cost of each step is $\mathcal{O}(N^2)$. We have confirmed numerically that it is sufficient to iterate $\ln^2 N + 4$ times to achieve the desired convergence rate. So the total cost of the method is $\mathcal{O}(N^2 \ln^2 N)$.

5.3 Numerical Results

We will again consider the same specific case of the two-dimensional problem (3.1)-(3.2) with the solution $u(x, y) = x(x-1)e^x y(y-1)e^y$ that we used in the general formulation of the two-dimensional problem with (3.15). We now solve (3.12) using n steps of ADI in (5.12), (5.13) to obtain $\alpha^{(n)}$ which gives $\mathbf{u_h}$ of (3.18) via (3.17) with $\alpha^{(n)}$ replacing α.

The error in the discrete maximum norm at all (x_k, y_l) is defined in (3.19). In Table 6, for different values of N, we present the errors, e_N of (3.19) and estimated convergence rates, p of (2.29).

Table 6: Errors e_N and convergence rates p using ADI

N	4	8	16	32	64
e_N	4.73–05	5.26–06	3.91–07	2.72–08	1.81–09
p		3.7373	4.0858	4.0213	3.9994

We notice that we have fourth order convergence as we expected. The results are almost the same as when we used Gauss elimination in the two dimensional test problem. Again, we are able to increase the number of subintervals because we have reduced the cost of computations.

We also considered a non-separable test problem, $u(x, y) = x(x-1)y(y-1)e^{xy}$ and in Table 7, for different values of N, we present the errors, e_N of (3.19) and estimated convergence rates, p of (2.29).

Table 7: Errors e_N and convergence rates p using ADI, non-separable u

N	4	8	16	32	64
e_N	1.17–05	1.27–06	1.02–07	7.12–09	4.82–10
p		3.7779	3.9593	4.0183	3.9722

We again observe fourth-order convergence. In Table 8, for different values of N, we present the errors e_M of (4.20) and the estimated convergence rates p of (2.29) for the non-separable test problem.

Table 8: Errors e_M and convergence rates p using ADI, non-separable u

N	4	8	16	32	64
e_N	1.17–05	1.27–06	1.02–07	6.44–09	4.82–10
p		3.7745	3.9657	4.1665	3.8254

6 CONCLUSIONS

This thesis presents a cubic spline finite element method for solving Poisson's equation with Dirichlet boundary conditions in the unit square. Our work has shown a method to obtain fourth order convergence with almost the same cost as standard finite element methods that use piecewise linear basis functions. We also obtained explicit forms for the stiffness and mass matrices that arise with Dirichlet boundary conditions, and these formulas can be used in future work with cubic splines.

We used an MDA to improve the efficiency of solving the large linear systems that we encountered. The MDA provided a direct method for solving our tensor product system as a set independent one-dimensional problems. By using a heptadiagonal solver that performs Gauss elimination only on the non-zero diagonals, we were able to further minimize our computing cost to $\mathcal{O}(N^3)$.

Employing the ADI method allowed us to obtain accurate results at even less cost than MDA. ADI methods are favorable because they provide accurate results with only a small number of iterations. Again, using Gauss elimination on the non-zero diagonals further reduced our cost to $\mathcal{O}(N^2 \ln^2 N)$.

Because it was not easy to find an explicit form for Z for use in the MDA, we were unable to determine if FFTs could be used. Finding a form of Z that involves sines and cosines could be a topic of future research. The use of FFTs would allow us to develop an MDA that has cost $\mathcal{O}(N^2 \ln N)$.

Additionally, there is still work to be done in extending this cubic spline finite element method to solve non-trivial problems. MDAs and ADIs used in finite element methods can be modified for different, more general, boundary conditions [1], [2]. We could also generalize our method for other separable elliptic partial differential equations and for higher dimensions.

REFERENCES CITED

[1] B. Bialecki. "An Alternating Direction Implicit Method for Orthogonal Spline Collocation Linear Systems." *Numerische Mathematik*, 59, (1991): 413-429.

[2] B. Bialecki, G. Fairweather, and A. Karageorghis. "Matrix Decomposition Algorithms for Elliptic boundary Value Problems: A Survey." *Numerical Algorithms*, 56 (2011): 253-295.

[3] W.S. Bickley and J. McNamee. " Matrix and Other Direct Methods for the solution of systems of Linear Difference Equations." *Philos. Trans. R. Soc. Lond., Ser. A.*, 252, (1960): 69-131.

[4] W. Cheney and D. Kincaid. *Numerical Mathematics and Computing.* 6 Ed. Thomson Learning, Inc: Belmont, CA, 2008.

[5] C. R. Crawford. " Reduction of a band-symmetric generalized eigenvalue problem." *Trans. ACM*, 16 (1973): 41-44.

[6] M. S. Gockenbach. *Understanding and Implementing the Finite Element Method,* SIAM: Philadelphia, PA, 2006.

[7] L. Kaufman and D. Warner. "High-order, fast-direct methods for separable elliptic equations." *Numer. Anal*, 21 (1984): 674-694.

[8] L. Kaufman and D. Warner. "Algorithm 685: a program for solving separable elliptic equations." *ACM Trans. Math. Softw.*, 16 (1990): 325-351.

[9] R.E. Lynch, J.R. Rice, and D.H. Thomas. "Direct solution of partial difference equations by tensor product methods." *Numer. Math.*, 6 (1964): 185-199.

[10] R.E. Lynch, J.R. Rice, and D.H. Thomas. "Tensor product analysis of partial difference equations." *Bull. Am. Math. Soc.*, 70 (1964): 378-384.

[11] M.R. Osborne. " Direct methods for the solution of finite-difference approximations to separable partial differential equations." *Comput. J.*, 8 (1965/1966): 150-156.

[12] D.W. Peaceman and H.H. Rachford. "The Numerical Solution of Parabolic and Elliptic Differential Equations." *Journal of the Society for Industrial and Applied Mathematics.*, 3 (1955): 28-41.

[13] A.A. Samarskii and E.S. Nikolaev, E.S.. "Methods for Solving Grid Equations." *Moscow: Nauka*, (1978).

[14] A.A. Soliman. "A Galerkin Solution for Burger's Equation Using Cubic B-Spline Finite Elements." *Abstract and Applied Analysis*, vol. 2012, Article ID 527467 (2012): `http://dx.doi.org/10.1155/2012/527467`.

[15] E. L. Wachpress. *The ADI Model Problem,* Windsor, CA, 1995.

APPENDICES

A Stiffness Matrix Entries

It follows from (2.6) that

$$B'_m(x) = \begin{cases} h^{-3} g'_1(x - x_{m-2}), & x \in [x_{m-2}, x_{m-1}], \\ h^{-1} g'_2(\frac{x - x_{m-1}}{h}), & x \in [x_{m-1}, x_m], \\ -h^{-1} g'_2(\frac{x_{m+1} - x}{h}), & x \in [x_m, x_{m+1}], \\ -h^{-3} g'_1(x_{m+2} - x), & x \in [x_{m+1}, x_{m+2}], \\ 0, & \text{otherwise.} \end{cases} \tag{A.1}$$

where $g'_1(x) = 3x^2$ and $g'_2(x) = 3 + 6x - 9x^2$.

With the basis functions defined in (2.8), for each $B_m(x)$, the support is $[x_{m-2}, x_{m+2}]$. For x-values outside of this range, $B_m(x) = 0$, therefore the integrals for those ranges are zero, so we need only compute entries on the heptadiagonal.

For $m = 2, ..., N - 1$ using (2.13), (2.8), (A.1) we have

$$\begin{aligned} a_{m,m} &= \int_0^1 [B^D_m]'(x)[B^D_m]'(x)dx = \int_0^1 B'_m(x)B'_m(x)dx \\ &= h^{-6} \int_{x_{m-2}}^{x_{m-1}} [g'_1(x - x_{m-2})]^2 dx \ + h^{-2} \int_{x_{m-1}}^{x_m} [g'_2(\frac{x - x_{m-1}}{h})]^2 dx \\ &+ h^{-2} \int_{x_m}^{x_{m+1}} [g'_2(\frac{x_{m+1} - x}{h})]^2 dx \ + h^{-6} \int_{x_{m+1}}^{x_{m+2}} [g'_1(x_{m+2} - x)]^2 dx. \end{aligned}$$

Using the following substitutions

$$x = x_{m-2} + hs, \quad s \in [0, 1], \tag{A.2}$$

$$x = x_{m-1} + hs, \quad s \in [0, 1], \tag{A.3}$$

$$x = x_{m+1} - hs, \quad s \in [0, 1], \tag{A.4}$$

$$x = x_{m+2} - hs, \quad s \in [0, 1], \tag{A.5}$$

for the integrals over $[x_{m-2}, x_{m-1}]$, $[x_{m-1}, x_m]$, $[x_m, x_{m+1}]$, and $[x_{m+1}, x_{m+2}]$, respectively, and using $g'_1(hs) = h^2 g'_1(s)$, we get

$$
\begin{aligned}
a_{m,m} &= h^{-1}\int_0^1 [g_1'(s)]^2 ds + h^{-1}\int_0^1 [g_2'(s)]^2 ds \\
&\quad - h^{-1}\int_1^0 [g_2'(s)]^2 ds - h^{-1}\int_1^0 [g_1'(s)]^2 ds \\
&= 2h^{-1}\int_0^1 [g_1'(s)]^2 + [g_2'(s)]^2 ds \\
&= 2h^{-1}\int_0^1 [9s^4] + [(3+6s-9s^2)^2] ds \quad = 24h^{-1} = \frac{240}{10h},
\end{aligned} \tag{A.6}
$$

where in the last step we used symbolic integration on Wolfram Alpha.

For $m = 2, ..., N-2$ using (2.13), (2.8), (A.1) we also have

$$
\begin{aligned}
a_{m,m+1} &= \int_0^1 [B_m^D]'(x)[B_{m+1}^D]'(x)dx = \int_0^1 B_m'(x)B_{m+1}'(x)dx \\
&= h^{-4}\int_{x_{m-1}}^{x_m} g_2'(\frac{x-x_{m-1}}{h})\ g_1'(x-x_{m-1})dx - h^{-2}\int_{x_m}^{x_{m+1}} g_2'(\frac{x_{m+1}-x}{h})\ g_2'(\frac{x-x_m}{h})]dx \\
&+ h^{-4}\int_{x_{m+1}}^{x_{m+2}} g_1'(x_{m+2}-x)\ g_2'(\frac{x_{m+2}-x}{h})dx.
\end{aligned}
$$

Using (A.3), (A.4), and (A.5) for the integrals $[x_{m-1}, x_m]$, $[x_m, x_{m+1}]$, and $[x_{m+1}, x_{m+2}]$, respectively, we get

$$
\begin{aligned}
a_{m,m+1} &= h^{-1}\int_0^1 g_2'(s)g_1'(s)ds + h^{-1}\int_1^0 g_2'(s)g_2'(1-s)ds + h^{-1}\int_1^0 g_1'(s)g_2'(s)ds \\
&= 2h^{-1}\int_0^1 g_1'(s)g_2'(s)ds - h^{-1}\int_0^1 g_2'(1-s)g_2'(s)ds \\
&= h^{-1}\int_0^1 2(3s^2)(3+6s-9s^2) - (3+6(1-s)-9(1-s)^2)(3+6s-9s^2)ds \\
&= -\frac{9}{2}h^{-1} = -\frac{45}{10h},
\end{aligned} \tag{A.7}
$$

where in the last step we used symbolic integration on Wolfram Alpha.

For $m = 2, ..., N-3$ using (2.13), (2.8), (A.1) we also have

$$
\begin{aligned}
a_{m,m+2} &= \int_0^1 [B_m^D]'(x)[B_{m+2}^D]'(x)dx = \int_0^1 B_m'(x)B_{m+2}'(x)dx \\
&= -h^{-4}\int_{x_m}^{x_{m+1}} g_2'(\frac{x_{m+1}-x}{h})\ g_1'(x-x_m)dx \\
&- h^{-4}\int_{x_{m+1}}^{x_{m+2}} g_1'(x_{m+2}-x)\ g_2'(\frac{x-x_{m+1}}{h})dx
\end{aligned}
$$

Using the substitution (A.4) for the integral over $[x_m, x_{m+1}]$ and (A.5) for the integral over $[x_{m+1}, x_{m+2}]$, we get

$$
\begin{aligned}
a_{m,m+2} &= -h^{-1}\int_0^1 g_2'(1-s)g_1'(s)ds + h^{-1}\int_1^0 g_2'(s)g_1'(1-s)ds \\
&= -2h^{-1}\int_0^1 g_2'(1-s)g_1'(s)ds \\
&= -2h^{-1}\int_0^1 (3+6(1-s)-9(1-s)^2)(3s^2)ds \\
&= -\frac{36}{5}h^{-1} \ = -\frac{72}{10h},
\end{aligned}
\tag{A.8}
$$

where in the last step we used symbolic integration on Wolfram Alpha.

Using (2.13), (2.8), (A.1) for $m = 2, ..., N-4$ we also have

$$
\begin{aligned}
a_{m,m+3} &= \int_0^1 [B_m^D]'(x)[B_{m+3}^D]'(x)dx = \int_0^1 B_m'(x)B_{m+3}'(x)dx \\
&= -h^{-6}\int_{x_{m+1}}^{x_{m+2}} g_1'(x_{m+2}-x)\ g_1'(x-x_{m+1})dx.
\end{aligned}
$$

Using (A.5), we get

$$
\begin{aligned}
a_{m,m+3} &= -h^{-1}\int_0^1 g_1'(1-s)g_1'(s)ds \\
&= -h^{-1}\int_0^1 (3(1-s)^2)(3s^2)ds = -\frac{3}{10}h^{-1}.
\end{aligned}
\tag{A.9}
$$

Integrals for the entries $a_{0,2},\ a_{0,3},\ a_{1,2},\ a_{1,3},\ a_{1,4}$ though we no longer have that $B_m(x) = B_m^D(x)$, reduce to integrals we have already computed, so using (A.6), (A.7), (A.8), and (A.9), we have

$$a_{0,2} = \int_0^1 [B_0^D]'(x)[B_2^D]'(x)dx = \int_0^1 B_0'(x)B_2'(x)dx - 4\int_0^1 B_{-1}'(x)B_2'(x)dx$$
$$= a_{m,m+2} - 4a_{m,m+3} = -\frac{36}{5}h^{-1} + \frac{12}{10}h^{-1} = -6h^{-1},$$

$$a_{0,3} = \int_0^1 [B_0^D]'(x)[B_3^D]'(x)dx = \int_0^1 B_0'(x)B_3'(x)dx - 4\int_0^1 B_{-1}'(x)B_3'(x)dx$$
$$= a_{m,m+3} - 0 = -\frac{3}{10}h^{-1},$$

$$a_{1,2} = \int_0^1 [B_1^D]'(x)[B_2^D]'(x)dx = \int_0^1 B_1'(x)B_2'(x)dx - \int_0^1 B_{-1}'(x)B_2'(x)dx$$
$$= a_{m,m+1} - a_{m,m+3} = -\frac{9}{2}h^{-1} - \left(-\frac{3}{10}h^{-1}\right) = -\frac{21}{5}h^{-1} = -\frac{42}{10h},$$

$$a_{1,3} = \int_0^1 [B_1^D]'(x)[B_3^D]'(x)dx = \int_0^1 B_1'(x)B_3'(x)dx - \int_0^1 B_{-1}'(x)B_3'(x)dx$$
$$= a_{m,m+2} - 0 = -\frac{36}{5}h^{-1},$$

and

$$a_{1,4} = \int_0^1 [B_1^D]'(x)[B_4^D]'(x) = dx\int_0^1 B_1'(x)B_4'(x)dx - \int_0^1 B_{-1}'(x)B_4'(x)dx$$
$$= a_{m,m+3} - 0 = -\frac{3}{10}h^{-1}.$$

The integrals for $a_{0,0}, a_{0,1}$, and $a_{1,1}$ will be more complicated since we no longer have that $B_m(x) = B_m^D(x)$ or that integrals will reduce to integrals we have already computed. Since these elements lie near edges of the matrix, some of the intervals of integration can be dropped because they lie outside of our defined interval $[x_0, x_{N+1}]$. Using (2.13), (2.8), (A.1), we have

$$
\begin{aligned}
a_{0,0} &= \int_0^1 [B_0^D]'(x)[B_0^D]'(x)dx = \int_{x_0}^{x_1} [B_0'(x) - 4B_{-1}'(x)]^2 dx + \int_{x_1}^{x_2} [B_0'(x)]^2 dx \\
&= \int_{x_0}^{x_1} [B_0'(x)]^2 dx - 8\int_{x_0}^{x_1} B_{-1}'(x)B_0'(x)dx + 16\int_{x_0}^{x_1} [B_{-1}'(x)]^2 dx + \int_{x_1}^{x_2} [B_0'(x)]^2 dx \\
&= h^{-2}\int_{x_0}^{x_1} \left[g_2'\left(\frac{x_1 - x}{h}\right)\right]^2 dx - 8h^{-4}\int_{x_0}^{x_1} g_1'(x_1 - x)g_2'\left(\frac{x_1 - x}{h}\right) dx \\
&\quad + 16h^{-6}\int_{x_0}^{x_1} [g_1'(x_1 - x)]^2 dx + h^{-6}\int_{x_1}^{x_2} [g_1'(x_2 - x)]^2 dx
\end{aligned}
\tag{A.10}
$$

$$
\tag{A.11}
$$

Using (A.4) and (A.5) for the integrals over $[x_0, x_1]$ and $[x_1, x_2]$, respectively, with $m = 0$, we get

$$
\begin{aligned}
a_{0,0} &= -h^{-1}\int_1^0 [g_2'(s)]^2 ds + 8h^{-1}\int_1^0 g_1'(s)g_2'(s)ds - 16h^{-1}\int_1^0 [g_1'(s)]^2 ds - h^{-1}\int_1^0 [g_1'(s)]^2 ds \\
&= h^{-1}\int_0^1 (3 + 6s - 9s^2)^2 - 8(3s^2)(3 + 6s - 9s^2) + 16(3s^2)^2 + (3s^2)^2 ds \\
&= 24h^{-1} = 80\left(\frac{3}{10h}\right).
\end{aligned}
$$

Using (2.13), (2.8), (A.1), we also have

$$
\begin{aligned}
a_{0,1} &= \int_0^1 [B_1^D]'(x)[B_1^D]'(x)dx = \int_{x_0}^{x_1} [B_0'(x) - 4B_{-1}'(x)][B_1'(x) - B_{-1}'(x)]dx + \int_{x_1}^{x_2} B_0'(x)B_1'(x)dx \\
&= \int_{x_0}^{x_1} B_0'(x)B_1'(x)dx - \int_{x_0}^{x_1} B_{-1}'(x)B_0'(x)dx + 4\int_{x_0}^{x_1} [B_{-1}'(x)]^2 dx - 4\int_{x_0}^{x_1} B_{-1}'(x)B_1'(x)dx + \int_{x_1}^{x_2} B_0'(x)B_1'(x)dx \\
&= -h^{-2}\int_{x_0}^{x_1} g_2'\left(\frac{x_1 - x}{h}\right) g_2'\left(\frac{x - x_0}{h}\right) dx - h^{-4}\int_{x_0}^{x_1} g_1'(x_1 - x)g_2'\left(\frac{x_1 - x}{h}\right) dx + 4h^{-6}\int_{x_0}^{x_1} [g_1'(x_1 - x)]^2 dx \\
&\quad + 4h^{-4}\int_{x_0}^{x_1} g_1'(x_1 - x)g_2'\left(\frac{x - x_0}{h}\right) dx + h^{-4}\int_{x_1}^{x_2} g_1'(x_2 - x)g_2'\left(\frac{x_2 - x}{h}\right) dx.
\end{aligned}
\tag{A.12}
$$

Using (A.4) and (A.5) for the integrals over $[x_0, x_1]$ and $[x_1, x_2]$, respectively, with $m = 0$, we get

$$
\begin{aligned}
a_{0,1} &= h^{-1}\int_1^0 g_2'(s)g_2'(1-s)ds + h^{-1}\int_1^0 g_1'(s)g_2'(s)ds - 4h^{-1}\int_1^0 [g_1'(s)]^2 ds \\
&\quad - 4h^{-1}\int_1^0 g_1'(s)g_2'(1-s)ds - h^{-1}\int_1^0 g_1'(s)g_2'(s)ds \\
&= h^{-1}\int_0^1 -g_2'(s)g_2'(1-s) + 4[g_1'(s)]^2 + 4g_1'(s)g_2'(1-s)ds \\
&= h^{-1}\int_0^1 -(3+6s-9s^2)(3+6(1-s)-9(1-s)^2) + 4(3s^2)^2 + 4(3s^2)(3+6(1-s)-9(1-s)^2)ds \\
&= \frac{129}{10h} = 43\left(\frac{3}{10h}\right).
\end{aligned}
$$

Using (2.13), (2.8), (A.1), we also have

$$
\begin{aligned}
a_{1,1} &= \int_0^1 [B_1^D]'(x)[B_1^D]'(x)dx = \int_{x_0}^{x_1} [B_1'(x) - B_{-1}'(x)]^2 dx + \int_{x_1}^{x_2} [B_1'(x)]^2 dx + \int_{x_2}^{x_3} [B_1'(x)]^2 dx \\
&= \int_{x_0}^{x_1} [B_1'(x)]^2 dx - 2\int_{x_0}^{x_1} B_{-1}'(x)B_1'(x)dx + \int_{x_0}^{x_1} [B_{-1}'(x)]^2 dx + \int_{x_1}^{x_2} [B_1'(x)]^2 dx + \int_{x_2}^{x_3} [B_1'(x)]^2 dx \\
&= h^{-2}\int_{x_0}^{x_1} [g_2'\left(\frac{x-x_0}{h}\right)]^2 dx + 2h^{-4}\int_{x_0}^{x_1} g_1'(x_1-x)g_2'\left(\frac{x-x_0}{h}\right)dx + h^{-6}\int_{x_0}^{x_1} [g_1'(x_1-x)]^2 dx \\
&\quad + h^{-2}\int_{x_1}^{x_2} [g_2'\left(\frac{x_2-x}{h}\right)]^2 dx + h^{-6}\int_{x_2}^{x_3} [g_1'(x_3-x)]^2 dx.
\end{aligned}
$$

Using (A.3), (A.4), and (A.5) for the integrals over$[x_0, x_1]$, $[x_1, x_2]$ and $[x_2, x_3]$, respectively,with $m = 1$, we get

$$
\begin{aligned}
a_{1,1} &= h^{-1}\int_0^1 [g_2'(s)]^2 ds + 2h^{-1}\int_0^1 g_1'(1-s)g_2'(s)ds + h^{-1}\int_0^1 [g_1'(1-s)]^2 ds - h^{-1}\int_1^0 [g_2'(s)]^2 ds - h^{-1}\int_1^0 [g_1'(s)]^2 ds \\
&= h^{-1}\int_0^1 2[g_2'(s)]^2 + 2g_1'(1-s)g_2'(s) + [g_1'(1-s)]^2 + [g_1'(s)]^2 ds \\
&= h^{-1}\int_0^1 2(3+6s-9s^2)^2 + 2(3(1-s)^2)(3+6s-9s^2) + [3(1-s)^2]^2 + [3s^2]^2 ds \\
&= \frac{156}{5h} = 104\left(\frac{3}{10h}\right).
\end{aligned}
$$

We seek to show that $a_{0,0} = a_{N+1,N+1}$. Using (2.13), (2.8), (A.1), we have

$$
\begin{aligned}
a_{N+1,N+1} &= \int_0^1 [B_{N+1}^D]'(x)[B_{N+1}^D]'(x)dx = \int_0^1 [B'_{N+1}(x) - 4B'_{N+2}(x)]^2 dx \\
&= \int_{x_{N-1}}^{x_N} [B'_{N+1}(x)]^2 dx + \int_{x_N}^{x_{N+1}} [B'_{N+1}(x) - 4B'_{N+2}(x)]^2 dx \\
&= \int_{x_{N-1}}^{x_N} [B'_{N+1}(x)]^2 dx + \int_{x_N}^{x_{N+1}} [B'_{N+1}(x)]^2 dx - 8\int_{x_N}^{x_{N+1}} B'_{N+1}(x)B'_{N-1}(x)dx + 16\int_{x_N}^{x_{N+1}} [B'_{N+2}(x)]^2 dx \\
&= h^{-6}\int_{x_{N-1}}^{x_N} [g'_1(x - x_{N-1})]^2 dx + h^{-2}\int_{x_N}^{x_{N+1}} [g'_2\left(\frac{x - x_N}{h}\right)]^2 dx \\
&\quad - 8h^{-4}\int_{x_N}^{x_{N+1}} g'_2\left(\frac{x - x_N}{h}\right) g'_1(x - x_N)dx + 16h^{-6}\int_{x_N}^{x_{N+1}} [g'_1(x - x_N)]^2 dx. \qquad \text{(A.13)}
\end{aligned}
$$

If we make the following substitution,

$$
x_2 - t = x - x_{N-1}, \quad t \in [0, 1], \qquad \text{(A.14)}
$$

then (A.13) becomes

$$
\begin{aligned}
a_{N+1,N+1} = h^{-6}\int_{x_1}^{x_2} [g'_1(x_2 - t)]^2 dt + h^{-2}\int_{x_0}^{x_1} [g'_2\left(\frac{x_1 - t}{h}\right)]^2 dt \\
- 8h^{-4}\int_{x_0}^{x_1} g'_1(x_1 - t)g'_2\left(\frac{x_1 - t}{h}\right) dt + 16h^{-6}\int_{x_0}^{x_1} [g'_1(x_1 - t)]^2 dt. \qquad \text{(A.15)}
\end{aligned}
$$

If we change the independent variable t to x, we notice that (A.15) is equivalent to (A.11). By examination, we can also show that $a_{0,1} = a_{N,N+1}$. Using (2.13), (2.8), (A.1), we have

$$
\begin{aligned}
a_{N,N+1} &= \int_0^1 [B_N^D]'(x)[B_{N+1}^D]'(x)dx = \int_0^1 [B'_N(x) - B'_{N+2}(x)][B'_{N+1}(x) - 4B'_{N+2}(x)]dx \\
&= \int_{x_{N-1}}^{x_N} B'_N(x)B'_{N+1}(x)dx + \int_{x_N}^{x_{N+1}} [B'_N(x) - B'_{N+2}(x)][B'_{N+1}(x) - 4B'_{N+2}(x)]dx \\
&= \int_{x_{N-1}}^{x_N} B'_N(x)B'_{N+1}(x)dx + \int_{x_N}^{x_{N+1}} B'_N(x)B'_{N+1}(x)dx + 4\int_{x_N}^{x_{N+1}} [B'_{N+2}(x)]^2 dx \\
&\quad - 4\int_{x_N}^{x_{N+1}} B'_N(x)B'_{N+2}(x)dx - \int_{x_N}^{x_{N+1}} B'_{N+1}(x)B'_{N+2}(x)dx \\
&= h^{-4}\int_{x_{N-1}}^{x_N} g'_2(\frac{x - x_{N-1}}{h})g'_1(x - x_{N-1})dx - h^{-2}\int_{x_N}^{x_{N-1}} g'_2\left(\frac{x_{N+1} - x}{h}\right) g'_2\left(\frac{x - x_N}{h}\right) dx \\
&\quad + 4h^{-6}\int_{x_N}^{x_{N+1}} [g'_1(x - x_N)]^2 dx + 4h^{-4}\int_{x_N}^{x_{N+1}} g'_2\left(\frac{x_{N+1} - x}{h}\right) g'_1(x - x_N)dx \\
&\quad - h^{-4}\int_{x_N}^{x_{N+1}} g'_2\left(\frac{x - x_N}{h}\right) g'_1(x - x_N)dx \qquad \text{(A.16)}
\end{aligned}
$$

Using the substitution (A.14) gives

$$
\begin{aligned}
a_{N,N+1} = h^{-4} \int_{x_1}^{x_2} g_2'(\frac{x_2 - t}{h}) g_1'(x_2 - t)dt - h^{-2} \int_{x_0}^{x_1} g_2' \left(\frac{t - x_0}{h}\right) g_2' \left(\frac{x_1 - t}{h}\right) dt \\
+ 4h^{-6} \int_{x_0}^{x_1} [g_1'(x_1 - t)]^2 dt + 4h^{-4} \int_{x_0}^{x_1} g_2' \left(\frac{t - x_0}{h}\right) g_1'(x_1 - t)dt \qquad \text{(A.17)} \\
- h^{-4} \int_{x_N}^{x_{N+1}} g_2' \left(\frac{x_1 - t}{h}\right) g_1'(x_1 - t)dt \qquad \text{(A.18)}
\end{aligned}
$$

If we change the independent variable t to x, we notice that (A.18) is equivalent to (A.12). We could continue and similarly show that $a_{i,N+1} = a_{0,N+1-i}$ for $i = N-2, N-1, N, N+1$ and $a_{i,N} = a_{1,N+1-i}$ for $i = N-3, N-2, N-1, N, N+1$. Thus inputting all of the entries we have computed and taking advantage of the symmetry of A, the stiffness matrix is

$$
A = \frac{3}{10h} \begin{bmatrix}
80 & 43 & -20 & -1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
43 & 104 & -14 & -24 & -1 & \ddots & & & & \vdots \\
-20 & -14 & 80 & -15 & -24 & -1 & \ddots & & & \vdots \\
-1 & -24 & -15 & 80 & -15 & -24 & -1 & \ddots & & \vdots \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\
0 & 0 & -1 & -24 & -15 & 80 & -15 & -24 & -1 & 0 \\
\vdots & & \ddots & -1 & -24 & -15 & 80 & -15 & -24 & -1 \\
\vdots & & & \ddots & -1 & -24 & -15 & 80 & -14 & -20 \\
\vdots & & & & \ddots & -1 & -24 & -14 & 104 & 43 \\
0 & \cdots & \cdots & \cdots & \cdots & 0 & -1 & -20 & 43 & 80
\end{bmatrix}.
$$

B Mass Matrix Entries

Using (2.6), (2.8), and (2.14):

$$\begin{aligned}
b_{m,m} &= \int_0^1 B_m^D(x)B_m^D(x)dx = \int_0^1 B_m(x)B_m(x)dx \\
&= h^{-6}\int_{x_{m-2}}^{x_{m-1}} [g_1(x-x_{m-2})]^2dx \ + \int_{x_{m-1}}^{x_m} [g_2(\frac{x-x_{m-1}}{h})]^2dx \\
&+ \int_{x_m}^{x_{m+1}} [g_2(\frac{x_{m+1}-x}{h})]^2dx \ + h^{-6}\int_{x_{m+1}}^{x_{m+2}} [g_1(x_{m+2}-x)]^2dx
\end{aligned} \tag{B.1}$$

To simplify, use (A.2), (A.3), (A.4), and (A.5), to get

$$\begin{aligned}
b_{m,m} &= h\int_0^1 (g_1(s))^2ds + h\int_0^1 (g_2(s))^2ds + h\int_0^1 (g_2(s))^2ds + h\int_0^1 (g_1(s))^2ds \\
&= 2h\int_0^1 (s^3)^2ds + (1+3s+3s^2-3s^3)^2ds = \frac{604}{35}h = \frac{2416h}{140}.
\end{aligned}$$

Using (2.6), (2.8), and (2.14), we also have

$$\begin{aligned}
b_{m,m+1} &= \int_0^1 B_m^D(x)B_{m+1}^D(x)dx = \int_{x_{m-1}}^{x_m} g_2\left(\frac{x-x_{m-1}}{h}\right)\ h^{-3}g_1(x-x_{m-1})dx \\
&+ \int_{x_m}^{x_{m+1}} g_2\left(\frac{x_{m+1}-x}{h}\right)\ g_2\left(\frac{x-x_m}{h}\right)dx + \int_{x_{m+1}}^{x_{m+2}} h^{-3}g_1(x_{m+2}-x)\ g_2\left(\frac{x_{m+2}-x}{h}\right)dx
\end{aligned}$$

Use the substitutions (A.3), (A.4), (A.5), we get

$$\begin{aligned}
b_{m,m+1} &= h\int_0^1 g_2(s)g_1(s)ds + h\int_0^1 g_2(s)g_2(1-s)ds + h\int_0^1 g_1(s))g_2(s)ds \\
&= 2h\int_0^1 g_2(s)g_1(s)ds + h\int_0^1 g_2(s)g_2(1-s)ds \\
&= h\int_0^1 2(1+3s+3s^2-3s^3)(s^3) + (1+3s+3s^2-3s^3)(1+3(1-s)+3(1-s)^2-3(1-s)^3)ds \\
&= \frac{1191h}{140}.
\end{aligned}$$

Using (2.6), (2.8), and (2.14), we also have

$$
\begin{aligned}
b_{m,m+2} &= \int_0^1 B_m^D(x) B_{m+2}^D(x)dx = \int_0^1 B_m(x)B_{m+2}(x)dx \\
&= \int_{x_m}^{x_{m+1}} g_2(\frac{x_{m+1}-x}{h})\ h^{-3} g_1(x-x_m)dx + \int_{x_{m+1}}^{x_{m+2}} h^{-3} g_1(x_{m+2}-x)\ g_2(\frac{x-x_{m+1}}{h})dx
\end{aligned}
$$

Use the substitutions (A.4) and (A.5) to get

$$
\begin{aligned}
b_{m,m+2} &= h\int_0^1 g_2(s)g_1(1-s)ds + h\int_0^1 g_1(s)g_2(1-s)ds \\
&= h\int_0^1 (1+3s+3s^2-3s^2)(1-s)^3 + (s^3)(1+3(1-s)+3(1-s)^2-3(1-s)^3)ds \\
&= \frac{6h}{7} = \frac{120h}{140}
\end{aligned}
$$

where in the last step we used symbolic integration on Wolfram Alpha.

Using (2.6), (2.8), and (2.14), we also have

$$
\begin{aligned}
B_{m,m+3} &= \int_0^1 B_m^D(x)B_{m+3}^D(x)dx = \int_0^1 B_m(x)B_{m+3}(x)dx \\
&= \int_{x_{m+1}}^{x_{m+2}} h^{-3} g_1(x_{m+2}-x)\ h^{-3} g_1(x-x_{m+1})dx.
\end{aligned}
$$

Make the substitution $x = x_{m+1} + hs,\ s \in [0,1]$, to get

$$
\begin{aligned}
b_{m,m+3} &= h\int_0^1 g_1(1-s)g_1(s)ds \\
&= h\int_0^1 ((1-s)^3)(s^3)ds = \frac{h}{140},
\end{aligned}
$$

where in the last step we used symbolic integration on Wolfram Alpha.

The entries $b_{0,2}, b_{0,3}, b_{1,2}, b_{1,3}$, and $b_{1,4}$ reduce to integrals that have already been computed, so we have

$$
\begin{aligned}
b_{0,2} &= \int_0^1 B_0^D(x)B_2^D(x)dx = \int_0^1 B_0(x)B_2(x)dx - 4\int_0^1 B_{-1}(x)B_2(x)dx \\
&= b_{m,m+2} - 4b_{m,m+3} = -\frac{6h}{7} - 4\frac{h}{140} = \frac{116h}{140},
\end{aligned}
$$

$$
\begin{aligned}
b_{0,3} &= \int_0^1 B_0^D(x)B_3^D(x)dx = \int_0^1 B_0(x)B_3(x)dx - 4\int_0^1 B_{-1}(x)B_3(x)dx \\
&= b_{m,m+3} - 0 = \frac{h}{140},
\end{aligned}
$$

$$
\begin{aligned}
b_{1,2} &= \int_0^1 B_1^D(x)B_2^D(x)dx = \int_0^1 B_1(x)B_2(x)dx - \int_0^1 B_{-1}(x)B_2(x)dx \\
&= b_{m,m+1} - b_{m,m+3} = -\frac{1191h}{140} - \frac{h}{140} = \frac{1190h}{140},
\end{aligned}
$$

$$
\begin{aligned}
b_{1,3} &= \int_0^1 B_1^D(x)B_3^D(x)dx = \int_0^1 B_1(x)B_3(x)dx - \int_0^1 B_{-1}(x)B_3(x)dx \\
&= b_{m,m+2} - 0 = \frac{6h}{7},
\end{aligned}
$$

and

$$
\begin{aligned}
b_{1,4} &= \int_0^1 B_1^D(x)B_4^D(x)dx = \int_0^1 B_1(x)B_4(x)dx - \int_0^1 B_{-1}(x)B_4(x)dx \\
&= b_{m,m+3} - 0 = \frac{h}{140}.
\end{aligned}
$$

The integrals for $b_{0,0}, b_{0,1}, \text{and} b_{1,1}$ will be more complicated since we no longer have that $B_m(x) = B_m^D(x)$. Since these elements lie near edges of the matrix, some of the intervals of integration can be dropped because they lie outside of our defined interval $[x_0, x_{N+1}]$.

$$
\begin{aligned}
b_{0,0} &= \int_0^1 B_0^D(x) B_0^D(x) dx = \int_{x_0}^{x_1} [B_0(x) - 4B_{-1}(x)]^2 dx + \int_{x_1}^{x_2} [B_0(x)]^2 dx \\
&= \int_{x_0}^{x_1} [B_0(x)]^2 - 8B_{-1}(x)B_0(x) + 16[B_{-1}(x)]^2 + \int_{x_1}^{x_2} [B_0(x)]^2 dx \\
&= \int_{x_0}^{x_1} [g_2\left(\frac{x_1 - x}{h}\right)]^2 dx - 8h^{-3} \int_{x_0}^{x_1} g_1(x_1 - x) g_2\left(\frac{x_1 - x}{h}\right) dx \\
&\quad + 16h^{-6} \int_{x_0}^{x_1} [g_1(x_1 - x)]^2 dx + h^{-6} \int_{x_1}^{x_2} [g_1(x_2 - x)]^2 dx
\end{aligned}
$$

Using (A.4) and (A.5) for the integrals over $[x_0, x_1]$ and $[x_1, x_2]$, respectively, with $m = 0$, and noticing that (A.4) is equivalent to $x = (x_2 - h) - hs$, we get

$$
\begin{aligned}
b_{0,0} &= -h \int_1^0 [g_2(s)]^2 ds + 8h \int_1^0 g_1(s) g_2(s) ds - 16h \int_1^0 [g_1(s)]^2 ds - h \int_1^0 [g_1(1-s)]^2 ds \\
&= h \int_0^1 (1 + 3s + 3s^2 - 3s^3)^2 - 8(s^3)(1 + 3s + 3s^2 - 3s^3) + 16(s^3)^2 + ((1-s)^3)^2 ds \\
&= \frac{124h}{35} = \frac{496h}{140},
\end{aligned}
$$

and

$$
\begin{aligned}
b_{0,1} &= \int_0^1 B_0^D(x) B_1^D(x) dx = \int_{x_0}^{x_1} [B_0(x) - 4B_{-1}(x)][B_1(x) - B_{-1}(x)] dx + \int_{x_1}^{x_2} B_0(x) B_1(x) dx \\
&= \int_{x_0}^{x_1} B_0(x) B_1(x) dx - \int_{x_0}^{x_1} B_{-1}(x) B_0(x) dx - 4 \int_{x_0}^{x_1} B_{-1}(x) B_1(x) dx + 4 \int_{x_0}^{x_1} [B_{-1}(x)]^2 dx + \int_{x_1}^{x_2} B_0(x) B_1(x) dx \\
&= \int_{x_0}^{x_1} g_2\left(\frac{x_1 - x}{h}\right) g_2\left(\frac{x - x_0}{h}\right) dx - h^{-3} \int_{x_0}^{x_1} g_1(x_1 - x) g_2\left(\frac{x_1 - x}{h}\right) dx - 4h^{-3} \int_{x_0}^{x_1} g_1(x_1 - x) g_2\left(\frac{x - x_0}{h}\right) dx \\
&\quad + 4h^{-6} \int_{x_0}^{x_1} [g_1(x_1 - x)]^2 dx + h^{-3} \int_{x_1}^{x_2} g_1(x_2 - x) g_2\left(\frac{x_2 - x}{h}\right) dx
\end{aligned}
$$

Using (A.4) and (A.5) with $m = 0$ and recognizing (A.4) can be rewritten, $x = x_1 - hs = x_0 + h(1 - s)$, we get

$$
\begin{aligned}
b_{0,1} &= -h\int_1^0 g_2(s)g_2(1-s)ds + h\int_1^0 g_1(s)g_2(s)ds + 4h\int_1^0 g_1(s)g_2(1-s)ds - 4h\int_1^0 [g_1(s)]^2 ds - h\int_1^0 g_1(s)g_2(s)ds \\
&= h\int_0^1 g_2(s)g_2(1-s) - 4g_1(s)g_2(1-s) + 4[g_1(s)]^2 ds \\
&= h\int_0^1 (1+3s+3s^2-3s^3)(1+3(1-s)+3(1-s)^2-3(1-s)^3) \\
&\quad - 4(s^3)(1+3(1-s)+3(1-s)^2-3(1-s)^3) + 4(s^3)^2 ds \\
&= \frac{773h}{140},
\end{aligned}
$$

and

$$
\begin{aligned}
b_{1,1} &= \int_0^1 B_1^D(x)B_1^D(x)dx = \int_{x_0}^{x_1} [B_1(x) - B_{-1}(x)]^2 dx + \int_{x_1}^{x_2} [B_1(x)]^2 dx + \int_{x_2}^{x_3} [B_1(x)]^2 dx \\
&= \int_{x_0}^{x_1} [B_1(x)]^2 dx - 2\int_{x_0}^{x_1} B_{-1}(x)B_1(x)dx + \int_{x_0}^{x_1} [B_{-1}(x)]^2 dx + \int_{x_1}^{x_2} [B_1(x)]^2 dx + \int_{x_2}^{x_3} [B_1(x)]^2 dx \\
&= \int_{x_0}^{x_1} [g_2\left(\frac{x-x_0}{h}\right)]^2 dx - 2h^{-3}\int_{x_0}^{x_1} g_1(x_1-x)g_2\left(\frac{x-x_0}{h}\right)dx + h^{-6}\int_{x_0}^{x_1} [g_1(x_1-x)]^2 dx \\
&\quad + \int_{x_1}^{x_2} [g_2\left(\frac{x_2-x}{h}\right)]^2 dx + h^{-6}\int_{x_2}^{x_3} [g_1(x_3-x)]^2 dx
\end{aligned}
$$

Using (A.3), (A.4), and (A.5) with m=1 and recognizing (A.3) is equivalent to $x = x_1 - h(1-s)$, we get

$$
\begin{aligned}
b_{1,1} &= h\int_0^1 [g_2(s)]^2 ds - 2h\int_0^1 g_1(1-s)g_2(s)ds + h\int_0^1 [g_1(1-s)]^2 ds - h\int_1^0 [g_2(s)]^2 ds - h\int_1^0 [g_1(s)]^2 ds \\
&= h\int_0^1 2[g_2(s)]^2 - 2g_1(1-s)g_2'(s) + [g_1(1-s)]^2 + [g_1(s)]^2 ds \\
&= h\int_0^1 2(1+3s+3s^2-3s^3)^2 - 2((1-s)^3)(1+3s+3s^2-3s^3) + [(1-s)^3]^2 + [s^3]^2 ds \\
&= \frac{2296h}{140}
\end{aligned}
$$

Just as we showed $a_{i,N+1} = a_{0,N+1-i}$ for $i = N-2, N-1, N, N+1$ and $a_{i,N} = a_{1,N+1-i}$ for $i = N-3, N-2, N-1, N, N+1$, we could similarly show $b_{i,N+1} = b_{0,N+1-i}$ for $i = N-2, N-1, N, N+1$ and $b_{i,N} = b_{1,N+1-i}$ for $i = N-3, N-2, N-1, N, N+1$. Thus inputting all of the entries we have computed and taking advantage of the symmetry of B, the mass matrix is

$$B = \frac{h}{140}\begin{bmatrix} 496 & 773 & 116 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 773 & 2296 & 1190 & 120 & 1 & \ddots & & & & \vdots \\ 116 & 1190 & 2416 & 1191 & 120 & 1 & \ddots & & & \vdots \\ 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 & 0 \\ \vdots & & \ddots & 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 \\ \vdots & & & \ddots & 1 & 120 & 1191 & 2416 & 1190 & 116 \\ \vdots & & & & \ddots & 1 & 120 & 1190 & 2296 & 773 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 1 & 116 & 773 & 496 \end{bmatrix}.$$