Spring 5-2017

Krylov Subspace Spectral Methods for PDEs in Polar and Cylindrical Geometries

Megan Richardson
University of Southern Mississippi

KRYLOV SUBSPACE SPECTRAL METHODS FOR PDES IN POLAR AND CYLINDRICAL GEOMETRIES

by

Megan Richardson

A Dissertation
Submitted to the Graduate School
and the Department of Mathematics
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Approved:

Dr. James V. Lambers, Committee Chair
Associate Professor, Mathematics

Dr. Ching-Shyang Chen, Committee Member
Professor, Mathematics

Dr. Haiyan Tian, Committee Member
Associate Professor, Mathematics

Dr. Huiqing Zhu, Committee Member
Associate Professor, Mathematics

Dr. Karen S. Coats
Dean of the Graduate School

May 2017

ABSTRACT

KRYLOV SUBSPACE SPECTRAL METHODS FOR PDES IN POLAR AND CYLINDRICAL GEOMETRIES

by Megan Richardson

May 2017

As a result of stiff systems of ODEs, difficulties arise when using time stepping methods for PDEs. Krylov subspace spectral (KSS) methods get around the difficulties caused by stiffness by computing each component of the solution independently. In this dissertation, we extend the KSS method to a circular domain using polar coordinates. In addition to using these coordinates, we will approximate the solution using Legendre polynomials instead of Fourier basis functions. We will also compare KSS methods on a time-independent PDE to other iterative methods. Then we will shift our focus to three families of orthogonal polynomials on the interval $(-1,1)$, with weight function $\omega(x) \equiv 1$. These families of polynomials satisfy the boundary conditions (1) $p(1) = 0$, (2) $p(-1) = p(1) = 0$, and (3) $p(1) = p'(1) = 0$. The first two boundary conditions arise naturally from PDEs defined on a disk with Dirichlet boundary conditions and the requirement of regularity in Cartesian coordinates. The third boundary condition includes both Dirichlet and Neumann boundary conditions for a higher-order PDE. The families of orthogonal polynomials are obtained by orthogonalizing short linear combinations of Legendre polynomials that satisfy the same boundary conditions. Then, the three-term recurrence relations are derived. Finally, it is shown that from these recurrence relations, one can efficiently compute the corresponding recurrences for generalized Jacobi polynomials (GJPs) that satisfy the same boundary conditions.

ACKNOWLEDGMENTS

Although the process of writing this dissertation has been long journey, I could not have completed it by myself. I would like to thank all of those who have assisted me in this effort. First, I would like to express my deepest gratitude to my adviser, Dr. James Lambers. This dissertation would not have been possible without his time, patience, and guidance.

In addition, I am deeply grateful to Dr. C.S. Chen, Dr. Haiyan Tian, and Dr. Huiqing Zhu for serving on my committee and for their suggestions and guidance during the course of my research. I would also like to thank my fellow colleagues in the Department of Mathematics for their support and friendship over the years.

I would like to thank my family and friends for their continuous love and support. Last, but definitely not least, a special thank you goes to my parents for their endless love, support, and encouragement. I appreciate their sacrifices and I would not have been able to get through this without them.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure

LIST OF TABLES

Table

LIST OF ABBREVIATIONS

KSS	-	Krylov Subspace Spectral
GJP	-	Generalized Jacobi Polynomials
ODE	-	Ordinary Differential Equation
PDE	-	Partial Differential Equation
IBVP	-	Initial Boundary Value Problem

NOTATION AND GLOSSARY

General Usage and Terminology

The notation used in this text represents fairly standard mathematical and computational usage. In many cases these fields tend to use different preferred notation to indicate the same concept, and these have been reconciled to the extent possible, given the interdisciplinary nature of the material. In particular, the notation for partial derivatives varies extensively, and the notation used is chosen for stylistic convenience based on the application. While it would be convenient to utilize a standard nomenclature for this important symbol, the many alternatives currently in the published literature will continue to be utilized.

The capital letters, $A, B, \ldots$ are used to denote matrices, including capital Greek letters, e.g., Λ for a diagonal matrix. Functions which are denoted in boldface type typically represent vector valued functions, and real valued functions usually are set in lower case Roman or Greek letters. Lower case letters such as i, j, k, l, m, n and sometimes p and d are used to denote indices.

Vectors and matrices are typeset in square brackets, e.g., $[\cdot]$. The standard notation $\mathbf{e}_j$ refers to the standard basis vectors. In general the norms are typeset using double pairs of lines, e.g., $||\cdot||$, and the absolute value of numbers is denoted using a single pair of lines, e.g., $|\cdot|$.

Chapter 1
INTRODUCTION

1.1 Introduction

The time-dependent reaction-diffusion equation arises in numerous mathematical models. To solve most of these, we use numerical methods. However, the time-dependent ordinary differential equations (ODEs) that are derived from partial differential equations (PDEs) in time and space become extremely stiff [1]. As a result, both explicit and implicit time-stepping methods have problems solving the system.

Normally, explicit time-stepping methods such as Euler's method, or higher-order methods like Runge-Kutta, as well as the Lanczos method solve the system of ODE $u' = Au$ by approximating the matrix exponential e^{At} using a polynomial. But as A gets larger due to the increased spatial resolution, the eigenvalues of A tend to spread out. As a result, the polynomial must be higher-degree to attain sufficient accuracy which can be computationally expensive. Implicit time-stepping methods require a system of equations to be solved at each time step, but generally using iterative methods. These methods also rely on polynomial approximations.

To reduce the computational expense, we can use the Krylov Subspace Spectral (KSS) method. This method can compute the same approximation, but with low-degree approximations by using different interpolating polynomials for each component of the solution independently. That is, as the number of grid points increases, the efficiency of KSS becomes an advantage over the other time-stepping methods.

Previously, the KSS method has only been used on a rectangular domain with Fourier basis functions. However, using polar coordinates and Legendre polynomials, we will extend the KSS method to a circular domain. When mapping polar or cylindrical geometries to rectangular domains using polar coordinates, it makes sense to use spectral methods [25]. Numerous algorithms based on spectral-collocation and spectral-tau methods already exist. See, for example, [4, 6, 8, 14, 18].

After applying separation of variables in polar coordinates, the resulting PDEs that depend on the radial coordinate r and time t can be solved numerically using the Legendre-Galerkin formulation similar to that used for the steady-state problem [25]. It is natural to use bases of polynomials that satisfy the boundary conditions for each PDE, and these can easily be obtained by taking short linear combinations of Legendre polynomials.

However, the bases used in [25] are not orthogonal with respect to the weight function $\omega \equiv 1$. This weight function is used in the variational formulation of the PDE. In [26] orthogonal bases were introduced that satisfy these same boundary conditions. These bases are known as generalized Jacobi polynomials (GJPs) that have indices $\alpha, \beta \leq -1$ that are orthogonal with respect to the weight function $\omega^{\alpha,\beta}(x) \equiv (1-x)^{\alpha}(1-x)^{\beta}$. GJPs corresponding to specific indices were introduced in [26] for the purpose of solving differential equations of odd higher order. Generalization to other (non-integer) indices was carried out in [15] to obtain families of orthogonal polynomials for Chebyshev spectral methods or problems with singular coefficients. However, although these GJPs can be described in terms of short linear combinations of Legendre polynomials, at least for certain index pairs of interest [15, 26] the three-term recurrence relations characteristic of families of orthogonal polynomials are unknown.

In this dissertation, we develop families of orthogonal polynomials that satisfy the requisite boundary conditions, to facilitate transformation between physical and frequency space without using functions such as the Legendre polynomials that lie outside of the solution space. These families can also be modified to work with alternative weight functions, thus leading to the development of new numerical methods.

The outline of this dissertation is as follows. In Chapter 2, we provide context for the use of KSS methods and these families of polynomials by adapting the variational formulation employed in [25] to the time-independent and time-dependent PDE. In Chapter 3, we describe the KSS method and how it is implemented. In Chapter 4, we develop orthogonal polynomials with a weight function, $\omega \equiv 1$, that satisfy the boundary conditions (1) $p(1) = 0$, (2) $p(1) = p(1) = 0$, and (3) $p(1) = p'(1) = 0$ and we describe how these polynomials can be modified to obtain three-term recurrence relations for the GJPs described in [15, 26]. In Chapter 5, we provide numerical results. Chapter 6 contains the conclusions and possible paths for future work.

Chapter 2

SPATIAL DISCRETIZATION

On the circular domain, the unit disk, we will use short linear combinations of Legendre polynomials instead of Fourier basis functions to approximate the solution of various PDEs. In [25], Shen describes two problems that arise when using spectral methods on polar and cylindrical geometries. Specifically, transforming to polar coordinates results in singularity at the pole(s), and constant coefficients in Cartesian coordinates will have variable coefficients of the form $r^{\pm k}$ in polar coordinates [25]. To work around these problems, we will need to choose suitable basis functions and use variational formulations that include the pole conditions.

2.1 Conversion to Polar Coordinates

We consider the elliptic equation on a unit disk

$$\Delta U - \alpha U = F \text{ in } \Omega = \left\{(x,y) : x^2 + y^2 < 1\right\}, \tag{2.1}$$
$$U = 0 \text{ on } \partial\Omega, \tag{2.2}$$

where α is a constant. Following the approach used in [25], we can convert the initial boundary value problem (IBVP) in (2.1) – (2.2) to polar coordinates by applying the polar transformation $x = r\cos\theta$, $y = r\sin\theta$ and letting $u(r,\theta) = U(r\cos\theta, r\sin\theta)$, $f(r,\theta) = F(r\cos\theta, r\sin\theta)$. The resulting problem in polar coordinates is as follows:

$$u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta} - \alpha u = f, \quad (r,\theta) \in Q = (0,1) \times [0,2\pi), \tag{2.3}$$
$$u(1,\theta) = 0, \quad \theta \in [0,2\pi), \quad u \; 2\pi\text{-periodic in } \theta,$$

The solution is represented using the Fourier series

$$u(r,\theta) = \sum_{|m|=0}^{\infty} \left[u_{1,m}(r)\cos(m\theta) + u_{2,m}(r)\sin(m\theta)\right]. \tag{2.4}$$

The Fourier coefficients $u_{1,m}(r,\theta)$, $u_{2,m}(r,\theta)$ must satisfy the boundary conditions $u_{1,m}(1,\theta) = u_{2,m}(1,\theta) = 0$ for $m = 0,1,2,\ldots$. Due to the singularity at the pole $r = 0$, we must impose additional pole conditions on (2.4) to have regularity in Cartesian coordinates. For $u(r,\theta)$ to be infinitely differentiable in the Cartesian plane, the additional pole conditions are [25]

$$u_{1,m}(0) = u_{2,m}(0) = 0 \text{ for } m \neq 0. \tag{2.5}$$

By substituting in the series in (2.4) into (2.3) and applying the pole conditions in (2.5), we obtain the following ODEs, for each nonnegative integer m:

$$\begin{aligned} u_{rr} + \frac{1}{r}u_r - \left(\frac{m^2}{r^2} + \alpha\right)u &= f(r), \quad 0 < r < 1, & (2.6)\\ u(0) &= 0 \text{ if } m \neq 0, & (2.7)\\ u(1) &= 0, & (2.8) \end{aligned}$$

where u and f are generic functions.

2.1.1 Weighted Formulation

We will extend (2.6) to the interval $(-1,1)$ using a coordinate transformation as in [25]. Using the coordinate transformation $r = \frac{s+1}{2}$ in (2.6) and setting $v(s) = u\left(\frac{s+1}{2}\right)$, we obtain

$$\begin{aligned} v_{ss} + \frac{1}{s+1}v_s - \left(\frac{m^2}{(s+1)^2} + \frac{\alpha}{4}\right)v &= \frac{1}{4}f\left(\frac{s+1}{2}\right), \quad s \in I = (-1,1), & (2.9)\\ v(-1) &= 0, \text{ if } m \neq 0,\\ v(1) &= 0. \end{aligned}$$

To obtain a weighted variational formulation for (2.9), we must find $v \in X(m)$ such that

$$((s+1)v_s, (w\omega)_s) + \left(\frac{m^2}{s+1}v, w\right)_\omega + \frac{\alpha}{4}((s+1)v, w)_\omega = (g(s), w)_\omega \tag{2.10}$$

where $X(m) = H^1_{0,\omega}(I)$ if $m \neq 0$, $X(0) = \left\{v \in H^1_\omega(I) : u(1) = 0\right\}$, ω is a weight function, and $g(s) = \frac{1}{4}(s+1)f\left(\frac{s+1}{2}\right)$.

2.1.2 Legendre-Galerkin

To approximate (2.10) using the Legendre-Galerkin method, we let $\omega = 1$ and we have to find $v_N \in X_N(m)$ such that $\forall w \in X_N(m)$,

$$((s+1)(v_N)_s, w_s) + \left(\frac{m^2}{s+1}v_N, w\right) + \frac{\alpha}{4}((s+1)v_N, w) = (I_N g(s), w), \tag{2.11}$$

where I_N is the interpolation operator based on the Legendre-Gauss-Lobatto points. That is, $(I_N g)(t_i) = g(t_i)$, $i = 0, 1, \ldots, N$, where $\{t_i\}$ are the roots of $\left(1-t^2\right)L'_N(t)$ and L_N is the Legendre polynomial of degree N.

The Case $m \neq 0$

We let $L_k(t)$ be the kth degree Legendre polynomial. Then

$$X_N(m) = \text{span}\{\phi_i(t) = L_i(t) - L_{i+2}(t) : i = 0, 1, \ldots, N-2\}$$

to satisfy the boundary conditions $v(-1) = v(1) = 0$ in (2.7) and (2.8). Therefore, setting

$$m_{ij} = \int_I (t+1)\,\phi_j'\phi_i'\,dt, \quad M = (m_{ij}) \quad i,j = 0,1,\ldots,N-2, \tag{2.12}$$

$$b_{ij} = \int_I \frac{1}{t+1}\phi_j\phi_i\,dt, \quad B = (B_{ij}) \quad i,j = 0,1,\ldots,N-2, \tag{2.13}$$

$$c_{ij} = \int_I (t+1)\,\phi_j\phi_i\,dt, \quad C = (C_{ij}) \quad i,j = 0,1,\ldots,N-2, \tag{2.14}$$

$$f_i = \int_I I_N g\phi_i\,dt, \qquad \boldsymbol{f} = (f_i) \quad i,j = 0,1,\ldots,N-2, \tag{2.15}$$

$$v_N = \sum_{i=0}^{N-2} x_i\phi_i(t), \qquad \boldsymbol{x} = (x_i) \quad i = 0,1,\ldots,N-2, \tag{2.16}$$

and applying (2.12) – (2.14) to (2.11), we obtain the following matrix equation

$$\left(M + m^2 B + \frac{\alpha}{4}C\right)\boldsymbol{x} = \boldsymbol{f}. \tag{2.17}$$

The matrices M and B in (2.17) are both symmetric tri-diagonal with

$$m_{ij} = \begin{cases} 2i+4, & j = i+1, \\ 4i+6, & j = i, \end{cases} \qquad b_{ij} = \begin{cases} -\frac{2}{i+2}, & j = i+1, \\ \frac{2(2i+3)}{(i+1)(i+2)}, & j = i, \end{cases}$$

and the matrix C is symmetric seven-diagonal with

$$c_{ij} = \begin{cases} -\frac{2(i+3)}{(2i+5)(2i+7)}, & j = i+3, \\ -\frac{2}{2i+5}, & j = i+2, \\ \frac{2}{(2i+1)(2i+5)} + \frac{2(i+3)}{(2i+5)(2i+7)}, & j = i+1, \\ \frac{2}{2i+1} + \frac{2}{2i+5}, & j = i. \end{cases}$$

The Case $m = 0$

In the case where $m = 0$, (2.11) reduces to

$$((s+1)(v_N)_s, w_s) + \frac{\alpha}{4}((s+1)v_N, w) = (I_N g(s), w), \quad \forall w \in X_N(0).$$

As before, we let $L_k(t)$ be the kth degree Legendre polynomial, and define $X_N(0)$ to be the space of all polynomials of degree less than or equal to N that vanish at 1. This space can be described as [25]

$$X_N(m) = \text{span}\{\phi_i(t) = L_i(t) - L_{i+1}(t) : i = 0, 1, \ldots, N-1\}$$

to satisfy the boundary conditions $v(1) = 0$ in (2.8). Similarly, extending the indexes i and j to $N-1$ in the definitions found in (2.17), we obtain the following matrix equation

$$\left(M + \frac{\alpha}{4}C\right)\boldsymbol{x} = \boldsymbol{f}. \tag{2.18}$$

The matrix M in (2.18) is a diagonal matrix with $m_{ii} = 2i+2$ and the matrix C is a symmetric penta-diagonal matrix with

$$c_{ij} = \begin{cases} -\frac{2(i+2)}{(2i+3)(2i+5)}, & j = i+2, \\ \frac{4}{(2i+1)(2i+3)(2i+5)}, & j = i+1, \\ \frac{4(i+1)}{(2i+1)(2i+3)}, & j = i. \end{cases}$$

2.2 Time-Dependent Case

This problem can also be extended to the time-dependent reaction-diffusion equation on a unit disk

$$\begin{aligned} \Delta U - \alpha U &= \frac{\partial U}{\partial t} \text{ in } \Omega = \left\{(x,y) : x^2 + y^2 < 1\right\},\ t > 0 & (2.19) \\ U &= 0 \text{ on } \partial\Omega, & (2.20) \\ U(x,y,0) &= F(x,y) \text{ on } \Omega, & (2.21) \end{aligned}$$

where α is a constant. As a result of the spatial discretization of (2.19) – (2.21), we can rewrite the system in terms of ODEs

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} &= A\mathbf{u}, \\ \mathbf{u}(0) &= \mathbf{f}, \end{aligned} \tag{2.22}$$

where A is an $N \times N$ matrix.

After applying the polar transformations found in Section 2.1, we can convert (2.19) – (2.20) to polar coordinates. The resulting problem is as follows:

$$\begin{aligned} u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta} - \alpha u &= \frac{\partial u}{\partial t}, \quad (r,\theta) \in Q = (0,1) \times [0,2\pi), & (2.23) \\ u(1,t) &= 0, \quad \theta \in [0,2\pi), \quad u\ 2\pi\text{-periodic in } \theta, \\ u(r,\theta,0) &= \frac{\partial u}{\partial t}. \end{aligned}$$

Using the Fourier series, the solution can be represented as

$$u(r,t) = \sum_{|m|=0}^{\infty} \left[u_{1,m}(r,t)\cos(m\theta) + u_{2,m}(r,t)\sin(m\theta)\right]. \tag{2.24}$$

For $u(r,\theta)$ to be infinitely differentiable in the Cartesian plane, the additional pole conditions are

$$u_{1,m}(0,t) = u_{2,m}(0,t) = 0 \text{ for } m \neq 0 \tag{2.25}$$

By substituting the series in (2.24) into (2.23) and applying the pole conditions in (2.25), we obtain the following ODEs, for each nonnegative integer m:

$$\begin{aligned} u_{rr} + \frac{1}{r}u_r - \left(\frac{m^2}{r^2} + \alpha\right)u &= \frac{\partial u}{\partial t}, \quad 0 < r < 1, \\ u(r,0) &= f(r), \\ u(0,t) &= 0 \text{ if } m \neq 0, \\ u(1,t) &= 0, \end{aligned} \tag{2.26}$$

where u and f are generic functions.

We will extend the problem found in equation (2.26) to the interval $(-1,1)$ using a coordinate transformation. Using the coordinate transformations described in Section 2.1.1, we obtain

$$\begin{aligned} v_{ss} - \frac{1}{s+1}v_s - \left(\frac{m^2}{(s+1)^2} + \frac{\alpha}{4}\right)v &= \frac{1}{4}\frac{\partial v}{\partial t}, \quad s \in I = (-1,1), \\ v(s,0) &= g(s), \\ v(-1,t) &= 0, \text{ if } m \neq 0, \\ v(1,t) &= 0, \end{aligned} \tag{2.27}$$

where $g(s) = f\left(\frac{s+1}{2}\right)$. To obtain a weighted variational formulation for (2.27), we must find $v \in X(m)$ such that

$$((s+1)v_s, (w\omega)_s) + \left(\frac{m^2}{s+1}v, w\right)_\omega + \frac{\alpha}{4}((s+1)v, w)_\omega = \frac{1}{4}\left((s+1)\frac{\partial v}{\partial t}, w\right)_\omega \tag{2.28}$$

where $X(m) = H^1_{0,\omega}(I)$ if $m \neq 0$, $X(0) = \left\{v \in H^1_\omega(I) : u(1,t) = 0\right\}$, ω is a weight function.

Using the Legendre-Galerkin method to approximate (2.28) where $\omega = 1$, we have to find $v_N \in X_N(m)$ such that $\forall w \in X_N(m)$,

$$\begin{aligned} ((s+1)(v_N)_s, w_s) + \left(\frac{m^2}{s+1}v_N, w\right) + \frac{\alpha}{4}\left((s+1)\frac{\partial v_N}{\partial t}, w\right)_\omega &= \frac{1}{4}\left(\frac{\partial v_N}{\partial t}, w\right), \\ v_N(s,0) &= I_N g(s), \end{aligned} \tag{2.29}$$

where I_N is the interpolation operator based on the Legendre-Gauss-Lobatto points.

In the time-independent case, we get the system $Au = f$ where A is $M + \beta C$ and $M + m^2 B + \beta C$ for $m = 0$ and $m \neq 0$, respectively. We find that the solution to the system is

$\mathbf{u} = A^{-1}f$. But with the time-dependent case, the system becomes $C\mathbf{u}' = -A\mathbf{u}$. This means that $\mathbf{u}^{n+1} = \mathbf{e}^{C^{-1}(-A)\Delta t}\mathbf{u}^n$ which we will discuss in 2.2.1.

This problem arises in the time-dependent problem because of the difference in the weight functions. In [25], Shen was able to include a weight function of $s+1$ in the right-hand side, but this can not be done in the time-dependent case. Therefore, we use the weight function of 1.

2.2.1 Implicit Time-Stepping

We will begin by examining the equation

$$\mathbf{u}_t = L\mathbf{u} \tag{2.30}$$

where L is the linear operator $\Delta - \alpha I$. To discretize this problem, we will an approximate $\frac{du}{dt}$ by the following :

$$\frac{du}{dt} \approx \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} \tag{2.31}$$

where Δt is the change in time. Now, we can adapt this method to equation (2.28) by letting

$$\begin{aligned} \mathbf{u}^n &= ((s+1)\sum_j \phi_j \mathbf{x}_j \phi_i) \\ &= \sum_j ((s+1)\phi_i, \phi_j)\mathbf{x}_j. \end{aligned} \tag{2.32}$$

Notice that the inner products in (2.32) are entries of the matrix C found in(2.14); therefore, we have the following equation

$$\frac{C\mathbf{x}^{n+1} - C\mathbf{x}^n}{\Delta t} = -A\mathbf{x}^{n+1}. \tag{2.33}$$

We can solve (2.33) by multiplying both sides by Δt and then subtracting Cx^n from both sides as follows:

$$\begin{aligned} C\mathbf{x}^{n+1} - C\mathbf{x}^n &= -\Delta t A\mathbf{x}^{n+1} \\ C\mathbf{x}^{n+1} &= C\mathbf{x}^n - \Delta t A\mathbf{x}^{n+1}. \end{aligned}$$

Now, we can solve the equation for $\mathbf{x}^n$:

$$\begin{aligned} C\mathbf{x}^{n+1} + \Delta t A\mathbf{x}^{n+1} &= C\mathbf{x}^n \\ C^{-1}C\mathbf{x}^{n+1} + \Delta t C^{-1}A\mathbf{x}^{n+1} &= \mathbf{x}^n \\ \mathbf{x}^{n+1} + \Delta t C^{-1}A\mathbf{x}^{n+1} &= \mathbf{x}^n \\ (I + \Delta t C^{-1}A)\mathbf{x}^{n+1} &= \mathbf{x}^n. \end{aligned}$$

The Implicit Euler method has first-order accuracy. If we want to compute the solution implicitly and improve the order of accuracy to second degree, we will use the Crank-Nicolson method [3]. Applying this method to equations (2.30) – (2.32), we have

$$\frac{C\mathbf{x}^{n+1} - C\mathbf{x}^{n}}{\Delta t} = -A\left(\frac{\mathbf{x}^{n+1} + \mathbf{x}^{n}}{2}\right). \tag{2.34}$$

Multiplying both sides of equation (2.34) by Δt, we have

$$C\mathbf{x}^{n+1} - C\mathbf{x}^{n} = -\Delta t A\left(\frac{\mathbf{x}^{n+1} + \mathbf{x}^{n}}{2}\right). \tag{2.35}$$

Then we will solve the equation in terms of $\mathbf{x}^{n+1}$ on the left side and $\mathbf{x}^{n}$ on the right side as follows:

$$\begin{aligned} C\mathbf{x}^{n+1} - C\mathbf{x}^{n} &= -\frac{\Delta t}{2}A\mathbf{x}^{n+1} - \frac{\Delta t}{2}A\mathbf{x}^{n} \\ C\mathbf{x}^{n+1} + \frac{\Delta t}{2}A\mathbf{x}^{n+1} &= C\mathbf{x}^{n} - \frac{\Delta t}{2}A\mathbf{x}^{n} \\ (C + \frac{\Delta t}{2}A)\mathbf{x}^{n+1} &= (C - \frac{\Delta t}{2}A)\mathbf{x}^{n} \end{aligned}$$

We will solve the systems that arise from backward Euler and Crank-Nicolson by using KSS methods, as described in the Chapter 3, and then comparing that with using KSS for the temporal discretization instead.

Chapter 3

KRYLOV SUBSPACE SPECTRAL METHODS

3.1 KSS Method

In this section, we will give an overview of KSS methods. We will consider the PDE

$$u_t + Lu = 0$$

on $[0, 2\pi)$ with initial condition $u(x,0) = f(x)$ and periodic boundary conditions. We will let $S = e^{-L\Delta t}$ be the solution operator and let $\langle \cdot, \cdot \rangle$ denote the standard inner product. By applying the solution operator, S, to the computed solution $\tilde{u}(x,t_n)$, we can approximate the Fourier coefficients of the solution $\tilde{u}(x,t_{n+1})$ using the following formula

$$\hat{u}(\omega, t_{n+1}) = \left\langle \frac{1}{\sqrt{2\pi}} e^{i\omega x}, e^{-L\Delta t} \tilde{u}(x,t_n) \right\rangle. \tag{3.1}$$

Using an N-point grid with uniform spacing $h = \frac{2\pi}{N}$, we can approximate the operator L and solution operator S using $N \times N$ matrices, and the quantity in (3.1) can be approximated as a bilinear form,

$$\hat{u}(\omega, t_{n+1}) \approx \hat{\mathbf{e}}_\omega^H e^{-L_N \Delta t} \mathbf{u}(t_n),$$

where

$$[\hat{\mathbf{e}}_\omega]_j = \frac{1}{\sqrt{2\pi}} e^{i\omega jh}, \quad [\mathbf{u}(t_n)]_j = u(jh, t_n). \tag{3.2}$$

Due to the spatial discretization of (3.1), the following bilinear form is obtained

$$\mathbf{u}^T f(A) \mathbf{v}, \tag{3.3}$$

where $\mathbf{u} = \frac{1}{\sqrt{2\pi}} e^{i\omega x}$ and $\mathbf{v} = \tilde{u}(x,t_n)$ are N-vectors (these vectors come from the equations in 3.2), $A = L_N$ is an $N \times N$ symmetric positive definite matrix, where L_N is the discretization of L, and $f(\lambda) = e^{-\lambda \Delta t}$ [12].

Because A is a symmetric positive definite matrix, it has real eigenvalues

$$0 < a = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N = b, \tag{3.4}$$

and corresponding orthonormal eigenvectors $\mathbf{q}_j$, $j = 1, \ldots, N$. As a result, we can write (3.3) as

$$\mathbf{u}^T f(A) \mathbf{v} = \sum_{j=1}^{N} f(\lambda_j) \mathbf{u}^T \mathbf{q}_j \mathbf{q}_j^T \mathbf{v}.$$

Letting $a=\lambda_1$ and $b=\lambda_N$ be the smallest and largest eigenvalues of A, respectively, the measure $\alpha(\lambda)$ can be defined as

$$\alpha(\lambda)=\begin{cases}0, & if\ \lambda<a,\\ \sum_{j=i}^{N}\alpha_j\beta_j, & if\ \lambda_i\leq\lambda<\lambda_{i-1},\\ \sum_{j=1}^{N}\alpha_j\beta_j, & if\ b\leq\lambda,\end{cases} \tag{3.5}$$

where $\alpha_j=\mathbf{u}^T\mathbf{q}_j$ and $\beta_j=\mathbf{q}_j^T\mathbf{v}$. If (3.5) is positive and increasing, then (3.3) can be viewed as a Riemann-Stieltjes integral

$$u^T f(A)v=I[f]=\int_a^b f(\lambda)\,d\alpha(\lambda).$$

Using Gaussian quadrature to approximate $I[f]$, we obtain an approximation of the form

$$I[f]=\sum_{j=1}^{K}w_j f\left(t_j\right)+R[f],$$

where the nodes $t_j,\ j=1,\dots,K$ and the weights $w_j,\ j=1,\dots,K$ are found using the following Lanczos algorithm.
Choose $\mathbf{r}_0$, $\beta_0=||\mathbf{r}_0||_2$, and $\mathbf{x}_0=0$.

for $n=1,2,\dots K$
$\quad\mathbf{x}_n=\mathbf{r}_{n-1}/\beta_{n-1}$
$\quad\alpha_n=\mathbf{x}_n^T A\mathbf{x}_n$
$\quad\mathbf{r}_n=(A-\alpha_n I)\mathbf{x}_n-\beta_{n-1}\mathbf{x}_{n-1}$
$\quad\beta_n=||\mathbf{r}_n||_2$
end

In the case where $\mathbf{u}\neq\mathbf{v}$, the weights could possibly be negative, which would destabilize the quadrature rule [2]. To avoid this problem, we consider the block approach

$$[\mathbf{u}\ \ \mathbf{v}]^T f(A)[\mathbf{u}\ \ \mathbf{v}]. \tag{3.6}$$

As a result of (3.6), we have the following matrix-valued integral

$$\int_a^b f(\lambda)\,d\mu(\lambda)=\begin{bmatrix}\mathbf{u}^T f(A)\mathbf{u} & \mathbf{u}^T f(A)\mathbf{v}\\ \mathbf{v}^T f(A)\mathbf{u} & \mathbf{v}^T f(A)\mathbf{v}\end{bmatrix},$$

where $\mu(\lambda)$ is a 2×2 matrix, each entry of which is a measure of the form $\alpha(\lambda)$ found in (3.5). We seek a quadrature formula

$$\int_a^b f(\lambda)\,d\mu(\lambda)=\sum_{j=1}^{2K}f\left(\lambda_j\right)\mathbf{v}_j\mathbf{v}_j^T+error,$$

where λ_j is a scalar value and $\mathbf{v}_j$ is a two component vector. To find the nodes and weights, we apply the following block Lanczos algorithm [13]:
$X_0 = 0$, $R_0 = [\mathbf{u}\ \ \mathbf{v}]$, $R_0 = X_1 B_0$ (QR factorization).

for $n = 1, 2, \ldots K$
    $V = AX_n$
    $M_n = X_n^T V$
    if $n < K$
        $R_n = V - X_{n-1} B_{n-1}^T - X_n M_n$
        $R_n = X_{n+1} B_n$ (QR factorization)
end

As a result of using the block Lanczos method, we obtain the 2×2 matrices M_j and B_j. These matrices form a block tridiagonal matrix, $\mathcal{T}_K$, where the nodes λ_j are the eigenvalues of

$$\mathcal{T}_K = \begin{bmatrix} M_1 & B_1^T & & & \\ B_1 & M_2 & B_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & B_{K-2} & M_{K-1} & B_{K-1}^T \\ & & & B_{K-1} & M_K \end{bmatrix}. \tag{3.7}$$

We can apply the KSS method to the time-independent problem in equations (2.1) and (2.2) by letting the vectors $\mathbf{u} = \mathbf{e}_j$ and $\mathbf{v} = \mathbf{g}$. The function used for the time-independent problem is $f(x) = \frac{1}{x}$ as opposed to the exponential for the time-dependent problem. In the same manner, the KSS method can be applied to the time-dependent problem in equations (2.19) and (2.20) by letting vectors $\mathbf{u} = \mathbf{e}_j$ and $\mathbf{v} = \mathbf{u}^n$. Instead of using $e^{i\omega x}$, we use columns of identity because the matrix A is already in frequency space. Therefore, using the columns of identity makes this approach more simple. We will use the same idea as the KSS method, except we are computing the coefficients using the polynomials ϕ_i that are short linear combinations of Legendre polynomials instead of the Fourier coefficients in (3.1).

To start the block KSS method for the time-dependent PDE, we define

$$R_0 = \begin{bmatrix} \mathbf{e}_j & \mathbf{u}^n \end{bmatrix},$$

where j is the degree of the Legendre polynomial. Now we can compute QR factorization of R_0 which yields

$$R_0 = X_1 B_0,$$

where

$$X_1 = \begin{bmatrix} \mathbf{e}_j & \frac{\mathbf{u}^n}{\|\mathbf{u}^n\|_2} \end{bmatrix}$$

and

$$B_0 = \begin{bmatrix} 1 & [\mathbf{u}^n]_j \\ 0 & \left\| \mathbf{u}^n - \mathbf{e}_j \mathbf{e}_j^T \mathbf{u}^n \right\| \end{bmatrix}.$$

Applying the block Lanczos iteration to the discretized operator L_N with initial block X_1 yields a block tridiagonal matrix, $\mathcal{T}_K$, of the form (3.7)[5]. By diagonalizing $\mathcal{T}_K$ we obtain the nodes and (matrix-valued) weights for the Gaussian quadrature rule needed to approximate each component of the solution.

3.2 Optimization

The main idea behind KSS methods is to compute each component of the solution, in some orthonormal basis, using an approximation that is optimal for that component. Specifically, each component uses a different polynomial approximation of $S(L_N;\Delta t)$, where the function S is based on the solution operator of the PDE and L_N is the discretization of the spatial differential operator. Combining all of the components together, we have a solution of the form [20]

$$\mathbf{u}^{n+1} = f(L_N;\Delta t)\mathbf{u}^n = \sum_{j=0}^{2K} D_j(\Delta t) A^j \mathbf{u}^n,$$

where $D_j(\Delta t)$ is a matrix that is diagonal in the chosen basis and K is the number of block Lanczos iterations. The diagonal entries are the coefficients of these interpolating polynomials in the monomial basis, with each row corresponding to a particular component. In the original block KSS method [20], the interpolation points are obtained by using block Lanczos iteration and then diagonalizing a $2K \times 2K$ matrix, $\mathcal{T}_K$, for each component.

We will start by letting $\mathbf{u}$ be a discretization of the solution on a uniform N-point grid. By applying KSS methods using the initial block $R_0 = \begin{bmatrix} \mathbf{e}_j & \mathbf{u}^n \end{bmatrix}$ where each $j = 1, 2, \ldots, N$, we can start the first iteration of the block Lanczos algorithm by finding the QR-factorization of R_0. It follows that

$$R_0 = X_1 B_{0,}$$

where

$$X_1 = \begin{bmatrix} \mathbf{e}_j & \frac{\mathbf{u}^n}{\|\mathbf{u}^n\|_2} \end{bmatrix}$$

and

$$B_0 = \begin{bmatrix} 1 & [\mathbf{u}^n]_j \\ 0 & \left\| \mathbf{u}^n - e_j e_j^T \mathbf{u}^n \right\| \end{bmatrix}.$$

The vectors $\mathbf{e}_k$ and $\mathbf{u}^n$ become orthogonal as $k \to \infty$. If $i + j$ is odd, then the (i, j) entry of $\mathcal{T}_K$ and $\lim_{k\to\infty} \mathbf{e}_k^T p(A)\mathbf{u}^n \to 0$ as long as $p(A)\mathbf{u}^n$ is smooth where $p(A)$ is a polynomial of A. For example, we consider the 6×6 case where

$$\mathcal{T}_K = \begin{bmatrix} X & 0 & X & 0 & 0 & 0 \\ 0 & Y & 0 & Y & 0 & 0 \\ X & 0 & X & 0 & X & 0 \\ 0 & Y & 0 & Y & 0 & Y \\ 0 & 0 & X & 0 & X & 0 \\ 0 & 0 & 0 & Y & 0 & Y \end{bmatrix},$$

as $k \to \infty$. Now we can find the permutation matrix, P, by grouping the odd-numbered rows and columns together as follows:

$$P^T \mathcal{T}_K P = \begin{bmatrix} X & X & 0 & 0 & 0 & 0 \\ X & X & X & 0 & 0 & 0 \\ 0 & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & Y & Y & 0 \\ 0 & 0 & 0 & Y & Y & Y \\ 0 & 0 & 0 & 0 & Y & Y \end{bmatrix}, \quad (3.8)$$

where X use non-block Lanczos with initial vector $\mathbf{e}_j$ and Y use initial vector $\mathbf{u}^n$. As a result, the eigenvalue problem for the matrix in (3.8) decouples and the block Gaussian quadrature nodes can be obtained by computing the eigenvalues of these smaller, tridiagonal matrices [21].

We can obtain the frequency-dependent nodes by applying the Lanczos algorithm described in Section 3.1 to matrix A with initial vector $\mathbf{e}_j$, and the frequency-dependent nodes by applying the Lanczos algorithm to with initial vector $\mathbf{u}^n$. We will begin using the matrix from the $m = 0$ case as follows:

$$A = \begin{bmatrix} M_1 & B_1 & C_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & \ddots & 0 \\ 0 & C_2 & B_3 & M_4 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & C_{N-2} & B_{N-1} & M_N \end{bmatrix}$$

where the main diagonal entries M_N are (2.12) plus (2.14) and $B_{N-1}$and C_{N-2} are defined using equation (2.14). Then we let

$$\mathbf{x}_1 = \mathbf{e}_j$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where 1 is the jth component of the vector and $\mathbf{x}_0$ is the zero vector

$$\begin{aligned} \mathbf{x}_0 &= \mathbf{0} \\ &= \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}. \end{aligned}$$

For $k = 1$, we will compute the value of α_1 where

$$\begin{aligned} \alpha_1 &= \mathbf{x}_1^T A \mathbf{x}_1 \\ &= \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} M_1 & B_1 & C_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & \ddots & 0 \\ 0 & C_2 & B_3 & M_4 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & C_{N-2} & B_{N-1} & M_N \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \cdots & 0 & C_{j-2} & B_{j-1} & M_j & B_j & C_j & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &= M_j \\ &= 2j + 2. \end{aligned}$$

Then $\mathbf{r}_1$ can be computed as follows:

$$\mathbf{r}_1 = (A - \alpha_1 I)\,\mathbf{x}_1 - \beta_0 \mathbf{x}_0$$

$$= \left(\begin{bmatrix} M_1 & B_1 & C_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & \ddots & 0 \\ 0 & C_2 & B_3 & M_4 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & C_{N-2} & B_{N-1} & M_N \end{bmatrix} - M_j \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & \vdots \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \left(\begin{bmatrix} M_1 & B_1 & C_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & \ddots & 0 \\ 0 & C_2 & B_3 & M_4 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & C_{N-2} & B_{N-1} & M_N \end{bmatrix} - \begin{bmatrix} M_j & 0 & 0 & \cdots & 0 \\ 0 & M_j & 0 & & \vdots \\ 0 & 0 & M_j & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & M_j \end{bmatrix} \right) \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} (M_1 - M_j) & B_1 & C_1 & 0 & \cdots & 0 \\ B_1 & (M_2 - M_j) & B_2 & C_2 & \ddots & \vdots \\ C_1 & B_2 & (M_3 - M_j) & B_3 & \ddots & 0 \\ 0 & C_2 & B_3 & (M_4 - M_j) & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & C_{N-2} & B_{N-1} & (M_N - M_j) \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ C_{j-2} \\ 0 \\ 0 \\ 0 \\ C_j \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Now, we will compute the value of β_1

$$\begin{aligned} \beta_1 &= \|\mathbf{r}_1\|_2 \\ &= \sqrt{|C_{j-2}|^2 + |C_j|^2} \end{aligned}$$

and $\mathbf{x}_2$

$$\mathbf{x}_2 = \frac{\mathbf{r}_1}{\beta_1}$$

$$= \frac{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ C_{j-2} \\ 0 \\ 0 \\ 0 \\ C_j \\ 0 \\ \vdots \\ 0 \end{bmatrix}}{\sqrt{\left|C_{j-2}\right|^2 + \left|C_j\right|^2}}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{C_{j-2}}{\sqrt{\left|C_{j-2}\right|^2+\left|C_j\right|^2}} \\ 0 \\ 0 \\ 0 \\ \frac{C_j}{\sqrt{\left|C_{j-2}\right|^2+\left|C_j\right|^2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Because $\lim_{j\to\infty} C_{j-2} = 1$ and $\lim_{j\to\infty} C_j = 1$, $\frac{C_j}{\sqrt{|C_{j-2}|^2+|C_j|^2}} \to \frac{1}{\sqrt{2}}$. Now, we will repeat the process for $k = 2$ using the values from the previous step as follows:

$$\alpha_2 = \mathbf{x}_2^T A \mathbf{x}_2$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \begin{bmatrix} M_1 & B_1 & C_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & \ddots & 0 \\ 0 & C_2 & B_3 & M_4 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & C_{N-2} & B_{N-1} & M_N \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{\sqrt{2}}{8} C_{j-4} \cdot B_{j-3} \cdot M_{j-2} \cdot B_{j-2} \cdot C_{j-2} + \frac{\sqrt{2}}{8} C_j \cdot B_{j+1} \cdot M_{j+2} \cdot B_{j+2} \cdot C_{j+2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \frac{1}{2} M_{j-2} + \frac{1}{2} M_{j+2}$$

As $j \to \infty$,

$$\begin{aligned} \frac{M_{j-2} + M_{j+2}}{2} &\approx \frac{2(j-2) + 2 + 2(j+2) + 2}{2} \\ &\approx \frac{2j - 4 + 2 + 2j + 4 + 2}{2} \\ &\approx \frac{4j + 4}{2} \\ &\approx \frac{2(2j+2)}{2} \\ &\approx 2j + 2 \\ &\approx M_j. \end{aligned}$$

Therefore, $\alpha_2 \approx M_j$. Now,

$$\mathbf{r}_2 = (A - \alpha_2 I)\,\mathbf{x}_2 - \beta_1 \mathbf{x}_1$$

$$= \left(\begin{bmatrix} M_1 & B_1 & C_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & \ddots & 0 \\ 0 & C_2 & B_3 & M_4 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & C_{N-2} & B_{N-1} & M_N \end{bmatrix} - \begin{bmatrix} M_j & 0 & 0 & \cdots & 0 \\ 0 & M_j & 0 & & \vdots \\ 0 & 0 & M_j & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & M_j \end{bmatrix} \right)$$

$$\cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sqrt{\left|C_{j-2}\right|^2 + \left|C_j\right|^2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} (M_1 - M_j) & B_1 & C_1 & 0 & \cdots & 0 \\ B_1 & (M_2 - M_j) & B_2 & C_2 & \ddots & \vdots \\ C_1 & B_2 & (M_3 - M_j) & B_3 & \ddots & 0 \\ 0 & C_2 & B_3 & (M_4 - M_j) & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & C_{N-2} & B_{N-1} & (M_N - M_j) \end{bmatrix}$$

$$\cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sqrt{\left|C_{j-2}\right|^2 + \left|C_j\right|^2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}}C_{j-4} \\ \frac{1}{\sqrt{2}}B_{j-3} \\ \frac{1}{\sqrt{2}}\left(M_{j-2}-M_j\right) \\ \frac{1}{\sqrt{2}}B_{j-2} \\ \frac{1}{\sqrt{2}}C_{j-2}+\frac{1}{\sqrt{2}}C_j \\ \frac{1}{\sqrt{2}}B_{j+1} \\ \frac{1}{\sqrt{2}}\left(M_{j+2}-M_j\right) \\ \frac{1}{\sqrt{2}}B_{j+2} \\ \frac{1}{\sqrt{2}}C_{j+2} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sqrt{\left|C_{j-2}\right|^2+\left|C_j\right|^2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}}C_{j-4} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}}C_{j+2} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sqrt{\left|C_{j-2}\right|^2+\left|C_j\right|^2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} C_{j-4} \\ 0 \\ 0 \\ 0 \\ -\sqrt{\left|C_{j-2}\right|^2 + \left|C_j\right|^2} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} C_{j+2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

In the computation of $\mathbf{r}_2$, we neglect the lower order terms. Now, we will compute the values of β_2

$$\begin{aligned} \beta_2 &= \|\mathbf{r}_2\|_2 \\ &= \frac{1}{2}\sqrt{2\left|C_{j-4}\right|^2 + 4\left|C_{j-2}\right|^2 + 4\left|C_j\right|^2 + 2\left|C_{j+2}\right|^2} \end{aligned}$$

and $\mathbf{x}_3$

$$\begin{aligned} \mathbf{x}_3 &= \frac{\mathbf{r}_2}{\beta_2} \\ &= \frac{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} C_{j-4} \\ 0 \\ 0 \\ 0 \\ -\sqrt{\left|C_{j-2}\right|^2 + \left|C_j\right|^2} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} C_{j+2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}}{\frac{1}{2}\sqrt{2\left|C_{j-4}\right|^2 + 4\left|C_{j-2}\right|^2 + 4\left|C_j\right|^2 + 2\left|C_{j+2}\right|^2}} \end{aligned}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{\frac{1}{\sqrt{2}}C_{j-4}}{\frac{1}{2}\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ 0 \\ 0 \\ 0 \\ -\frac{\sqrt{|C_{j-2}|^2+|C_j|^2}}{\frac{1}{2}\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ 0 \\ 0 \\ 0 \\ \frac{\frac{1}{\sqrt{2}}C_{j+2}}{\frac{1}{2}\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{\sqrt{2}C_{j-4}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ 0 \\ 0 \\ 0 \\ -\frac{2\sqrt{|C_{j-2}|^2+|C_j|^2}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ 0 \\ 0 \\ 0 \\ \frac{\sqrt{2}C_{j+2}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

For $k = 3$, we will repeat the process.

$$\alpha_3 \quad = \quad \mathbf{x}_3^T A \mathbf{x}_3$$

$$
= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{\sqrt{2}C_{j-4}}{2\beta_2} \\ 0 \\ 0 \\ 0 \\ -\frac{2\sqrt{|C_{j-2}|^2+|C_j|^2}}{2\beta_2} \\ 0 \\ 0 \\ 0 \\ \frac{\sqrt{2}C_{j+2}}{2\beta_2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \begin{bmatrix} M_1 & B_1 & C_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & \ddots & 0 \\ 0 & C_2 & B_3 & M_4 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & C_{N-2} & B_{N-1} & M_N \end{bmatrix}
\cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{\sqrt{2}C_{j-4}}{2\beta_2} \\ 0 \\ 0 \\ 0 \\ -\frac{2\sqrt{|C_{j-2}|^2+|C_j|^2}}{2\beta_2} \\ 0 \\ 0 \\ 0 \\ \frac{\sqrt{2}C_{j+2}}{2\beta_2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}
$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{\sqrt{2}C_{j-4}M_{j-4}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ \frac{\sqrt{2}C_{j-4}B_{j-4}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ \frac{\sqrt{2}(C_{j-4})^2-2C_{j-2}\sqrt{|C_{j-2}|^2+|C_j|^2}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ -\frac{2B_{j-1}\sqrt{|C_{j-2}|^2+|C_j|^2}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ -\frac{2M_j\sqrt{|C_{j-2}|^2+|C_j|^2}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ -\frac{2B_j\sqrt{|C_{j-2}|^2+|C_j|^2}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ -\frac{2C_j\sqrt{|C_{j-2}|^2+|C_j|^2}+\sqrt{2}(C_{j+2})^2}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ \frac{\sqrt{2}C_{j+2}B_{j+3}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ \frac{\sqrt{2}C_{j+2}M_{j+4}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ \frac{\sqrt{2}C_{j+2}B_{j+4}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{\sqrt{2}C_{j-4}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ 0 \\ 0 \\ 0 \\ -\frac{2\sqrt{|C_{j-2}|^2+|C_j|^2}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ 0 \\ 0 \\ 0 \\ \frac{\sqrt{2}C_{j+2}}{\sqrt{2|C_{j-4}|^2+4|C_{j-2}|^2+4|C_j|^2+2|C_{j+2}|^2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{aligned}
&\approx \frac{2\left(C_{j-4}\right)^2 M_{j-4} + 4M_j\left(\left|C_{j-2}\right|^2 + \left|C_j\right|^2\right) + 2M_{j+4}\left(C_{j+4}\right)^2}{2\left|C_{j-4}\right|^2 + 4\left|C_{j-2}\right|^2 + 4\left|C_j\right|^2 + 2\left|C_{j+2}\right|^2} \\
&\approx \frac{2(1)^2\left[2(j-4)+2\right] + 4\left[2j+2\right]\left(1^2+1^2\right) + 2\left[2(j+4)+2\right](1)^2}{2(1)^2 + 4(1)^2 + 4(1)^2 + 2(1)^2} \\
&\approx \frac{2\left[2j-8+2\right] + 4\left[2j+2\right](2) + 2\left[2j+8+2\right]}{2+4+4+2} \\
&\approx \frac{2(2j-6) + 8(2j+2) + 2(2j+10)}{12} \\
&\approx \frac{4j-12+16j+16+4j+20}{12} \\
&\approx \frac{24j+24}{12} \\
&\approx 2j+2
\end{aligned}$$

$$\approx \quad M_j.$$

Usually, we carry out $K = 2$ or $K = 3$ iterations which corresponds to third and fifth-order accuracy in time for KSS. To determine the frequency-independent nodes, we calculate the eigenvalues of the $K \times K$ matrix that results from the Lanczos method with initial vector $\mathbf{u}^n$. Since the frequency-independent nodes do not depend on the frequency index, j, we only compute them once for each vector $\mathbf{u}$.

Comparing the calculated values of α_1, α_2, and α_3, we can see that they are approximately equal. Using the calculated values of α_k and β_k, we can construct the following Jacobi matrix:

$$J = \begin{bmatrix} \alpha_1 & \beta_1 & 0 \\ \beta_1 & \alpha_2 & \beta_2 \\ 0 & \beta_2 & \alpha_3 \end{bmatrix} \tag{3.9}$$

for three iterations. We then rewrite the matrix found in (3.9) as

$$J \approx \alpha_1 I + \tilde{J} \tag{3.10}$$

where

$$\tilde{J} \approx \alpha_1 I + \begin{bmatrix} 0 & \beta_1 & 0 \\ \beta_1 & 0 & \beta_2 \\ 0 & \beta_2 & 0 \end{bmatrix}. \tag{3.11}$$

Instead of directly finding the eigenvalues of (3.9), we can approximate them by finding the eigenvalues of $\tilde{J}$ found in (3.11) and adding a shift α_1 as follows:

$$\lambda(J) \approx \lambda(\tilde{J}) + \alpha_1.$$

We will repeat this process on the seven-diagonal matrix for the $m \neq 0$ case as follows:

$$A = \begin{bmatrix} M_1 & B_1 & C_1 & D_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & D_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & C_3 & \ddots & 0 \\ D_1 & C_2 & B_3 & M_4 & B_4 & \ddots & D_{N-3} \\ 0 & D_2 & C_3 & B_4 & M_5 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & D_{N-3} & C_{N-2} & B_{N-1} & M_N \end{bmatrix},$$

where main and off diagonal entries M_N and B_{N-1} respectively are composed of (2.12), (2.13), and (2.14), C_{N-2} and D_{N-2} are defined using equation (2.14) and the identity vector.

$$\mathbf{x}_1 \quad = \quad \mathbf{e}_j$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where 1 is the jth component and $\mathbf{x}_0$ is the zero column vector

$$\begin{aligned} \mathbf{x}_0 &= \mathbf{0} \\ &= \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}. \end{aligned}$$

For $k = 1$,

$$\begin{aligned} \alpha_1 &= \mathbf{x}_1^T A \mathbf{x}_1 \\ &= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \begin{bmatrix} M_1 & B_1 & C_1 & D_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & D_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & C_3 & \ddots & 0 \\ D_1 & C_2 & B_3 & M_4 & B_4 & \ddots & D_{N-3} \\ 0 & B_2 & C_3 & B_4 & M_5 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & D_{N-3} & C_{N-2} & B_{N-1} & M_N \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \cdots & 0 & D_{j-3} & C_{j-4} & M_j & B_j & C_j & D_j & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &\approx M_j \\ &\approx 4j + 6 \end{aligned}$$

$$\mathbf{r}_1 = (A - \alpha_1 I)\,\mathbf{x}_1 - \beta_0 \mathbf{x}_0$$

$$= \left(\begin{bmatrix} M_1 & B_1 & C_1 & D_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & D_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & C_3 & \ddots & 0 \\ D_1 & C_2 & B_3 & M_4 & B_4 & \ddots & D_{N-3} \\ 0 & B_2 & C_3 & B_4 & M_5 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & D_{N-3} & C_{N-2} & B_{N-1} & M_N \end{bmatrix} - M_j \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \left(\begin{bmatrix} M_1 & B_1 & C_1 & D_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & D_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & C_3 & \ddots & 0 \\ D_1 & C_2 & B_3 & M_4 & B_4 & \ddots & D_{N-3} \\ 0 & B_2 & C_3 & B_4 & M_5 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & D_{N-3} & C_{N-2} & B_{N-1} & M_N \end{bmatrix} - \begin{bmatrix} M_j & 0 & \cdots & 0 \\ 0 & M_j & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & M_j \end{bmatrix} \right) \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} (M_1 - M_j) & B_1 & C_1 & D_1 & 0 & \cdots & 0 \\ B_1 & (M_2 - M_j) & B_2 & C_2 & D_2 & \ddots & \vdots \\ C_1 & B_2 & (M_3 - M_j) & B_3 & C_3 & \ddots & 0 \\ D_1 & C_2 & B_3 & (M_4 - M_j) & B_4 & \ddots & D_{N-3} \\ 0 & B_2 & C_3 & B_4 & (M_5 - M_j) & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & D_{N-3} & C_{N-2} & B_{N-1} & (M_N - M_j) \end{bmatrix}$$

$$\cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ B_{j-1} \\ 0 \\ B_j \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{aligned}\beta_1 &= \|\mathbf{r}_1\|_2 \\ &= \sqrt{\left|B_{j-1}\right|^2 + \left|B_j\right|^2}\end{aligned}$$

$$\begin{aligned}\mathbf{x}_2 &= \frac{\mathbf{r}_1}{\beta_1} \\ &= \frac{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ B_{j-1} \\ 0 \\ B_j \\ 0 \\ \vdots \\ 0 \end{bmatrix}}{\sqrt{\left|B_{j-1}\right|^2 + \left|B_j\right|^2}} \\ &= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{B_{j-1}}{\sqrt{\left|B_{j-1}\right|^2 + \left|B_j\right|^2}} \\ 0 \\ \frac{B_j}{\sqrt{\left|B_{j-1}\right|^2 + \left|B_j\right|^2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}\end{aligned}$$

For $k = 2$

$$\alpha_2 = \mathbf{x}_2^T A \mathbf{x}_2$$

$$= \begin{bmatrix} 0 & \cdots & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & \cdots & 0 \end{bmatrix}$$

$$\begin{bmatrix} M_1 & B_1 & C_1 & D_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & D_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & C_3 & \ddots & 0 \\ D_1 & C_2 & B_3 & M_4 & B_4 & \ddots & D_{N-3} \\ 0 & B_2 & C_3 & B_4 & M_5 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & D_{N-3} & C_{N-2} & B_{N-1} & M_N \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{aligned} &= \frac{1}{2}M_{j-1} + \frac{1}{2}C_{j-1} + \frac{1}{2}C_{j-1} + \frac{1}{2}M_{j+1} \\ &= \frac{1}{2}M_{j-1} + C_{j-1} + \frac{1}{2}M_{j+1} \end{aligned}$$

As $j \to \infty$

$$\begin{aligned} \frac{M_{j-1} + M_{j+1}}{2} &\approx \frac{4(j-1)+6+4(j+1)+6}{2} \\ &\approx \frac{4j-4+6+4j+4+6}{2} \\ &\approx \frac{8j+12}{2} \\ &\approx \frac{2(4j+6)}{2} \\ &\approx 4j+6 \\ &\approx M_j. \end{aligned}$$

Therefore, $\alpha_2 \approx M_j$. So,

$$\begin{aligned} \mathbf{r}_2 &= (A - \alpha_2 I)\,\mathbf{x}_2 - \beta_1 \mathbf{x}_1 \\ &= \left(\begin{bmatrix} M_1 & B_1 & C_1 & D_1 & 0 & \cdots & 0 \\ B_1 & M_2 & B_2 & C_2 & D_2 & \ddots & \vdots \\ C_1 & B_2 & M_3 & B_3 & C_3 & \ddots & 0 \\ D_1 & C_2 & B_3 & M_4 & B_4 & \ddots & D_{N-3} \\ 0 & B_2 & C_3 & B_4 & M_5 & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & D_{N-3} & C_{N-2} & B_{N-1} & M_N \end{bmatrix} - \begin{bmatrix} M_j & 0 & 0 & \cdots & 0 \\ 0 & M_j & 0 & & \vdots \\ 0 & 0 & M_j & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & M_j \end{bmatrix} \right) \end{aligned}$$

$$\cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sqrt{\left|B_{j-1}\right|^2 + \left|B_j\right|^2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} (M_1 - M_j) & B_1 & C_1 & D_1 & 0 & \cdots & 0 \\ B_1 & (M_2 - M_j) & B_2 & C_2 & D_2 & \ddots & \vdots \\ C_1 & B_2 & (M_3 - M_j) & B_3 & C_3 & \ddots & 0 \\ D_1 & C_2 & B_3 & (M_4 - M_j) & B_4 & \ddots & D_{N-3} \\ 0 & B_2 & C_3 & B_4 & (M_5 - M_j) & \ddots & C_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & B_{N-1} \\ 0 & \cdots & 0 & D_{N-3} & C_{N-2} & B_{N-1} & (M_N - M_j) \end{bmatrix}$$

$$\cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sqrt{\left|B_{j-1}\right|^2 + \left|B_j\right|^2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} D_{j-4} \\ \frac{1}{\sqrt{2}} C_{j-3} \\ \frac{1}{\sqrt{2}} \left(B_{j-2} + D_{j-2} \right) \\ \frac{1}{\sqrt{2}} \left(M_{j-1} - M_j \right) + \frac{1}{\sqrt{2}} C_{j-1} \\ \frac{1}{\sqrt{2}} (B_{j-1} + B_j) \\ \frac{1}{\sqrt{2}} C_{j-1} + \frac{1}{\sqrt{2}} \left(M_{j+1} - M_j \right) \\ \frac{1}{\sqrt{2}} (B_{j+1} + D_{j-1}) \\ \frac{1}{\sqrt{2}} C_{j+1} \\ \frac{1}{\sqrt{2}} D_{j+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sqrt{\left|B_{j-1}\right|^2 + \left|B_j\right|^2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} B_{j-2} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} B_{j+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sqrt{\left|B_{j-1}\right|^2 + \left|B_j\right|^2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}} B_{j-2} \\ 0 \\ -\sqrt{\left|B_{j-1}\right|^2 + \left|B_j\right|^2} \\ 0 \\ \frac{1}{\sqrt{2}} B_{j+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Again, we neglect the lower order terms.

$$\begin{aligned}
\beta_2 &= \|\mathbf{r}_2\|_2 \\
&= \frac{1}{2}\sqrt{2\left|B_{j-2}\right|^2 + 4\left|B_{j-1}\right|^2 + 4\left|B_j\right|^2 + 2\left|B_{j+1}\right|^2}
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}_3 &= \frac{\mathbf{r}_2}{\beta_2} \\
&= \frac{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{2}}B_{j-2} \\ 0 \\ -\sqrt{\left|B_{j-1}\right|^2 + \left|B_j\right|^2} \\ 0 \\ \frac{1}{\sqrt{2}}B_{j+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}}{\frac{1}{2}\sqrt{2\left|B_{j-2}\right|^2 + 4\left|B_{j-1}\right|^2 + 4\left|B_j\right|^2 + 2\left|B_{j+1}\right|^2}} \\
&= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{\frac{1}{\sqrt{2}}B_{j-2}}{\frac{1}{2}\sqrt{2\left|B_{j-2}\right|^2+4\left|B_{j-1}\right|^2+4\left|B_j\right|^2+2\left|B_{j+1}\right|^2}} \\ 0 \\ -\frac{\sqrt{\left|B_{j-1}\right|^2+\left|B_j\right|^2}}{\frac{1}{2}\sqrt{2\left|B_{j-2}\right|^2+4\left|B_{j-1}\right|^2+4\left|B_j\right|^2+2\left|B_{j+1}\right|^2}} \\ 0 \\ \frac{\frac{1}{\sqrt{2}}B_{j+1}}{\frac{1}{2}\sqrt{2\left|B_{j-2}\right|^2+4\left|B_{j-1}\right|^2+4\left|B_j\right|^2+2\left|B_{j+1}\right|^2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}
\end{aligned}$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{\sqrt{2}B_{j-2}}{\sqrt{2|B_{j-2}|^2+4|B_{j-1}|^2+4|B_j|^2+2|B_{j+1}|^2}} \\ 0 \\ -\frac{2\sqrt{|B_{j-1}|^2+|B_j|^2}}{\sqrt{2|B_{j-2}|^2+4|B_{j-1}|^2+4|B_j|^2+2|B_{j+1}|^2}} \\ 0 \\ \frac{\sqrt{2}B_{j+1}}{\sqrt{2|B_{j-2}|^2+4|B_{j-1}|^2+4|B_j|^2+2|B_{j+1}|^2}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The calculated values of α_1 and α_2 are also approximately equal. We can determine the nodes of this case in the same manner, except the α and β values will be different. Now, we have $\alpha_j = M_j$ for $m = 0$ and $m \neq 0$. For $m = 0$ the β_j values are

$$\beta_1 = \sqrt{|C_{j-2}|^2 + |C_j|^2}$$

$$\beta_2 = \frac{1}{2}\sqrt{2|C_{j-4}|^2 + 4|C_{j-2}|^2 + 4|C_j|^2 + 2|C_{j+2}|^2}.$$

For $m \neq 0$ the β_j values are

$$\beta_1 = \sqrt{|B_{j-1}|^2 + |B_j|^2}$$

$$\beta_2 = \frac{1}{2}\sqrt{2|B_{j-2}|^2 + 4|B_{j-1}|^2 + 4|B_j|^2 + 2|B_{j+1}|^2}.$$

When $K = 2$, we have the following:

$$\mathfrak{T}_2 = \begin{bmatrix} M_j & \beta_1 \\ \beta_1 & M_j \end{bmatrix}$$

where the eigenvalues are $\lambda = M_j \pm \beta_1$. Whens $K = 3$,

$$\mathfrak{T}_3 = \begin{bmatrix} M_j & \beta_1 & 0 \\ \beta_1 & M_j & \beta 2 \\ 0 & \beta_2 & M_j \end{bmatrix}$$

where the eigenvalues are $\lambda = M_j$ and $\lambda = M_j \pm \sqrt{\beta_1^2 + \beta_2^2}$.

Chapter 4

ORTHOGONAL POLYNOMIALS

4.1 The Case $m = 0$

In the case where $m = 0$, (2.11) reduces to

$$\left((t+1)\,v_N', w'\right) + \frac{\alpha}{4}\left((t+1)\,v_N, w\right) = \left(I_N g, w\right), \ \forall w \in X_N\,(0)\,.$$

As before, we let $L_k\,(t)$ be the kth degree Legendre polynomial, and define $X_N(0)$ to be the space of all polynomials of degree less than or equal to N that vanish at 1. This space can be described as [25]

$$X_N\,(0) = \text{span}\left\{\phi_i\,(t) = L_i\,(t) - L_{i+1}\,(t) : \, i = 0, 1, \ldots, N-1\right\},$$

where $\phi_i\,(t)$ is the ith basis function. By applying the Gram-Schmidt process [3] to these basis functions, $\phi_i\,(t)$, we can obtain a new set of orthogonal polynomials that will be denoted by $\tilde{\phi}_i$, $i = 0, 1, 2, \ldots$, where the degree of ϕ_i and $\tilde{\phi}_i$ is $i+1$. The new basis functions, $\tilde{\phi}_i$, can be found by computing

$$\tilde{\phi}_i = \phi_i - \sum_{k=0}^{i-1} \frac{\left\langle \tilde{\phi}_k, \phi_i \right\rangle}{\left\langle \tilde{\phi}_k, \tilde{\phi}_k \right\rangle} \tilde{\phi}_k. \tag{4.1}$$

Fortunately, for $0 \le k \le i-2$,

$$\begin{aligned} \left\langle \tilde{\phi}_k, \phi_i \right\rangle &= \left\langle \tilde{\phi}_k, L_i - L_{i+1} \right\rangle \ \text{ if } \ 0 \le k \le i-2, \\ &= \left\langle \tilde{\phi}_k, L_i \right\rangle - \left\langle \tilde{\phi}_k, L_{i+1} \right\rangle \\ &= 0, \end{aligned}$$

due to the orthogonality of the Legendre polynomials, thus greatly simplifying the computation of $\tilde{\phi}_i$. To start the sequence $\tilde{\phi}_i$, we let

$$\begin{aligned} \tilde{\phi}_0 &= \phi_0, \\ &= L_0 - L_1, \\ &= 1 - x, \end{aligned}$$

so then

$$\begin{aligned}
\tilde{\phi}_1 &= \phi_1 - \frac{\langle \tilde{\phi}_0, \phi_1 \rangle}{\langle \tilde{\phi}_0, \tilde{\phi}_0 \rangle}\tilde{\phi}_0 \\
&= \phi_1 - \frac{\int_1^{-1} \left(\tilde{\phi}_0 \cdot \phi_1\right) dx}{\int_1^{-1} \left(\tilde{\phi}_0 \cdot \tilde{\phi}_0\right) dx}\tilde{\phi}_0 \\
&= \left(-\frac{3}{2}x^2 + x + \frac{1}{2}\right) - \frac{\frac{-2}{3}}{\frac{8}{3}}(1-x) \\
&= -\frac{3}{2}x^2 + x + \frac{1}{2} + \frac{1}{4}(1-x) \\
&= -\frac{3}{2}x^2 + x + \frac{1}{2} + \frac{1}{4} - \frac{1}{4}x \\
&= -\frac{3}{2}x^2 + \frac{3}{4}x + \frac{3}{4}
\end{aligned}$$

and

$$\begin{aligned}
\tilde{\phi}_2 &= \phi_2 - \frac{\langle \tilde{\phi}_0, \phi_2 \rangle}{\langle \tilde{\phi}_0, \tilde{\phi}_0 \rangle}\tilde{\phi}_0 - \frac{\langle \tilde{\phi}_1, \phi_2 \rangle}{\langle \tilde{\phi}_1, \tilde{\phi}_1 \rangle}\tilde{\phi}_1 \\
&= \phi_2 - \frac{\int_1^{-1} \left(\tilde{\phi}_0 \cdot \phi_2\right) dx}{\int_1^{-1} \left(\tilde{\phi}_0 \cdot \tilde{\phi}_0\right) dx}\tilde{\phi}_0 - \frac{\int_1^{-1} \left(\tilde{\phi}_1 \cdot \phi_2\right) dx}{\int_1^{-1} \left(\tilde{\phi}_1 \cdot \tilde{\phi}_1\right) dx}\tilde{\phi}_1 \\
&= \frac{1}{2}\left(-5x^3 + 3x^2 + 3x - 1\right) - \frac{0}{\frac{8}{3}}(1-x) - \frac{-\frac{2}{5}}{\frac{9}{10}}\left(-\frac{3}{2}x^2 + \frac{3}{4}x + \frac{3}{4}\right) \\
&= -\frac{5}{2}x^3 + \frac{3}{2}x^2 + \frac{3}{2}x - \frac{1}{2} + \frac{4}{9}\left(-\frac{3}{2}x^2 + \frac{3}{4}x + \frac{3}{4}\right) \\
&= -\frac{5}{2}x^3 + \frac{3}{2}x^2 + \frac{3}{2}x - \frac{1}{2} - \frac{3}{2}x^2 + \frac{1}{3}x + \frac{1}{3} \\
&= -\frac{5}{2}x^3 + \frac{5}{6}x^2 + \frac{11}{6}x - \frac{1}{6}.
\end{aligned}$$

The first several polynomials $\tilde{\phi}_0, \tilde{\phi}_1, \ldots, \tilde{\phi}_i$ are shown in Figure 4.1. Now, comparing ϕ_1 with $\tilde{\phi}_1$ and ϕ_2 with $\tilde{\phi}_2$ we can find a general formula for the values of $\tilde{\phi}_i$ in terms of ϕ_i. By subtracting ϕ_i from $\tilde{\phi}_i$, we obtain

$$\begin{aligned}
\tilde{\phi}_1 - \phi_1 &= -\frac{3}{2}x^2 + \frac{3}{4}x + \frac{3}{4} - \left(-\frac{3}{2}x^2 + x + \frac{1}{2}\right) \\
&= -\frac{3}{2}x^2 + \frac{3}{4}x + \frac{3}{4} + \frac{3}{2}x^2 - x - \frac{1}{2} \\
&= -\frac{1}{4}x + \frac{1}{4} \\
&= \frac{1}{4}(1-x) \\
&= \frac{1}{4}\tilde{\phi}_0, \qquad (4.2)
\end{aligned}$$

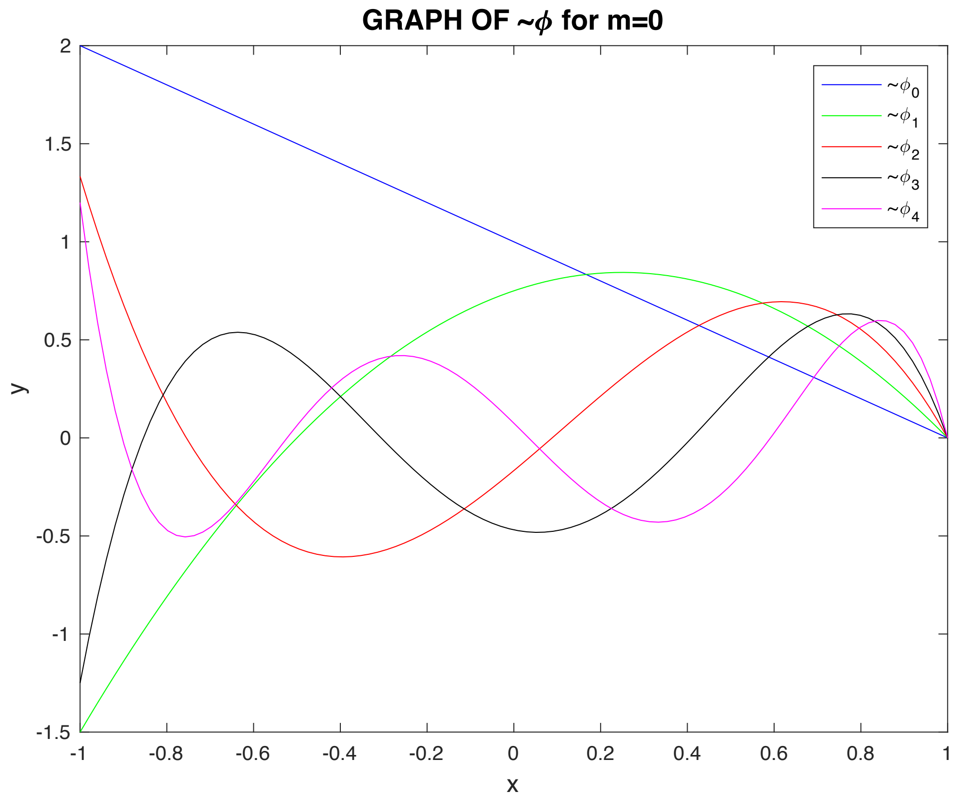


Figure 4.1: Graphs of $\tilde{\phi}_i$ for $i = 0, 1, 2, 3, 4$.

and

$$\begin{aligned}
\tilde{\phi}_2 - \phi_2 &= -\frac{5}{2}x^3 + \frac{5}{6}x^2 + \frac{11}{6}x - \frac{1}{6} - \left(-\frac{5}{2}x^3 + \frac{3}{2}x^2 + \frac{3}{2}x - \frac{1}{2}\right) \\
&= -\frac{5}{2}x^3 + \frac{5}{6}x^2 + \frac{11}{6}x - \frac{1}{6} + \frac{5}{2}x^3 - \frac{3}{2}x^2 - \frac{3}{2}x + \frac{1}{2} \\
&= -\frac{2}{3}x^2 + \frac{1}{3}x + \frac{1}{3} \\
&= \frac{4}{9}\left(-\frac{3}{2}x^2 + \frac{3}{4}x + \frac{3}{4}\right) \\
&= \frac{4}{9}\tilde{\phi}_1. \qquad (4.3)
\end{aligned}$$

This suggests a simple recurrence relation for $\tilde{\phi}_i$ in terms of ϕ_i. Before we prove that this relation holds in general, we need the following result.

Lemma 1. Let $N_k = \langle \tilde{\phi}_k, \tilde{\phi}_k \rangle$.Then

$$N_k = \frac{2(k+2)^2}{(k+1)^2(2k+3)}, \qquad (4.4)$$

$\forall k \geq 0$.

Proof: We proceed by induction. For the base case, we have

$$\begin{aligned}
N_0 &= \left\langle \tilde{\phi}_0, \tilde{\phi}_0, \right\rangle \\
&= \int_{-1}^{1} \tilde{\phi}_0 \cdot \tilde{\phi}_0 \, dx \\
&= \int_{-1}^{1} (1-x)(1-x)\, dx \\
&= \int_{-1}^{1} \left(x^2 - 2x + 1\right) dx \\
&= \frac{x^3}{3} - x^2 + x \Big|_{-1}^{1} \\
&= \left[\frac{(1)^3}{3} - (1)^2 + 1\right] - \left[\frac{(-1)^3}{3} - (-1)^2 + (-1)\right] \\
&= \frac{1}{3} - 1 + 1 + \frac{1}{3} + 1 + 1 \\
&= \frac{8}{3}.
\end{aligned}$$

For the induction step, we assume that there is a $k > 0$, such that $N_{k-1} = \frac{2(k+1)^2}{k^2(2k+1)}$. We must show that the formula found in equation (4.4) is true for k. Given that $\tilde{\phi}_k = \phi_k + \left(\frac{k}{k+1}\right)^2 \tilde{\phi}_{k-1}$, and using

$$\langle L_k, L_k \rangle = \frac{2}{2k+1}, \tag{4.5}$$

we have

$$\begin{aligned}
N_k &= \left\langle \tilde{\phi}_k, \tilde{\phi}_k \right\rangle \\
&= \left\langle \phi_k + \left(\frac{k}{k+1}\right)^2 \tilde{\phi}_{k-1}, \phi_k + \left(\frac{k}{k+1}\right)^2 \tilde{\phi}_{k-1} \right\rangle \\
&= \langle \phi_k, \phi_k \rangle + \left(\frac{k}{k+1}\right)^2 \left\langle \phi_k, \tilde{\phi}_{k-1} \right\rangle + \left(\frac{k}{k+1}\right)^2 \left\langle \tilde{\phi}_{k-1}, \phi_k \right\rangle + \left(\frac{k}{k+1}\right)^4 \left\langle \tilde{\phi}_{k-1}, \tilde{\phi}_{k-1} \right\rangle \\
&= \langle \phi_k, \phi_k \rangle + 2\left(\frac{k}{k+1}\right)^2 \left\langle \phi_k, \tilde{\phi}_{k-1} \right\rangle + \left(\frac{k}{k+1}\right)^4 \left\langle \tilde{\phi}_{k-1}, \tilde{\phi}_{k-1} \right\rangle \\
&= \langle L_k + L_{k+1}, L_k + L_{k+1} \rangle - 2\left(\frac{k}{k+1}\right)^2 \langle L_k, L_k \rangle + \left(\frac{k}{k+1}\right)^4 N_{k-1} \\
&= \frac{8(k+1)}{(2k+1)(2k+3)} - \frac{4k^2}{(k+1)^2(2k+1)} + \left(\frac{k}{k+1}\right)^4 N_{k-1} \\
&= \frac{4\left(3k^2 + 6k + 2\right)}{(k+1)^2(2k+1))(2k+3)} + \left(\frac{k}{k+1}\right)^4 N_{k-1} \\
&= \frac{4\left(3k^2 + 6k + 2\right)}{(k+1)^2(2k+1))(2k+3)} + \left(\frac{k}{k+1}\right)^4 \left(\frac{2}{2k+1} \cdot \frac{(k+1)^2}{k^2}\right)
\end{aligned}$$

$$\begin{aligned}
&= \frac{4\left(3k^2+6k+2\right)}{(k+1)^2(2k+1))(2k+3)} + \left(\frac{k}{k+1}\right)^4 \left(\frac{2(k+1)^2}{k^2(2k+1)}\right) \\
&= \frac{4\left(3k^2+6k+2\right)}{(k+1)^2(2k+1))(2k+3)} + \frac{2k^2}{(k+1)^2(2k+1)} \\
&= \frac{2\left(k^2+4k+4\right)}{(k+1)^2(2k+3)} \\
&= \frac{2(k+2)^2}{(k+1)^2(2k+3)}. \quad \square
\end{aligned}$$

We can now establish the pattern seen in (4.2), (4.3).

Theorem 1. If $\tilde{\phi}_0(x) = 1 - x$ and $\tilde{\phi}_i$ is obtained by orthogonalizing $\phi_i = L_{i+1} - L_i$ against $\tilde{\phi}_0, \tilde{\phi}_1, \ldots, \tilde{\phi}_{i-1}$, then

$$\tilde{\phi}_i = \phi_i + c_i \tilde{\phi}_{i-1} \tag{4.6}$$

for $i = 1, 2, \ldots$, where $c_i = \left(\frac{i}{i+1}\right)^2$.

Proof: Again we proceed by induction. For the base case, we will show that the theorem holds when $i = 1$:

$$\begin{aligned}
\tilde{\phi}_1 &= \phi_1 - \frac{\langle \tilde{\phi}_0, \phi_1 \rangle}{\langle \tilde{\phi}_0, \tilde{\phi}_0 \rangle} \tilde{\phi}_0 \\
&= x - \frac{3}{2}x^2 + \frac{1}{2} - \frac{\frac{-2}{3}}{\frac{8}{3}}(1-x) \\
&= x - \frac{3}{2}x^2 + \frac{1}{2} + \frac{1}{4}(1-x) \\
&= \phi_1 + \frac{1}{4}\tilde{\phi}_0.
\end{aligned} \tag{4.7}$$

Note that equation (4.7) is equivalent to equation (4.2). For the induction step, we assume that there is a $j \geq 0$, such that

$$\tilde{\phi}_j = \phi_j + \left(\frac{j}{j+1}\right)^2 \tilde{\phi}_{j-1}. \tag{4.8}$$

We show that (4.8) holds when $i = j + 1$. We have

$$\begin{aligned}
\tilde{\phi}_{j+1} &= \phi_{j+1} + \sum_{k=0}^{j} \frac{\langle \phi_{j+1}, \tilde{\phi}_j \rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j \rangle} \tilde{\phi}_j \\
&= \phi_{j+1} - \frac{\langle \phi_{j+1}, \tilde{\phi}_j \rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j \rangle} \tilde{\phi}_j \\
&= \phi_{j+1} - \frac{\langle L_{j+1} - L_{j+2}, \tilde{\phi}_j \rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j \rangle} \tilde{\phi}_j
\end{aligned}$$

$$
\begin{aligned}
&= \phi_{j+1} - \frac{\langle L_{j+1}, \tilde{\phi}_j\rangle - \langle L_{j+2}, \tilde{\phi}_j\rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j \\
&= \phi_{j+1} - \frac{\langle L_{j+1}, \tilde{\phi}_j\rangle - \left\langle L_{j+2}, \phi_j + (\frac{j}{j+1})^2\tilde{\phi}_{j-1}\right\rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j \\
&= \phi_{j+1} - \frac{\langle L_{j+1}, \tilde{\phi}_j\rangle - \left[\langle L_{j+2}, \phi_j\rangle + (\frac{j}{j+1})^2\langle L_{j+2}, \tilde{\phi}_{j-1}\rangle\right]}{\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j \\
&= \phi_{j+1} - \frac{\langle L_{j+1}, \tilde{\phi}_j\rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j \\
&= \phi_{j+1} - \frac{\langle L_{j+1}, \phi_j\rangle + (\frac{j}{j+1})^2\langle L_{j+1}, \tilde{\phi}_{j-1}\rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j \\
&= \phi_{j+1} - \frac{\langle L_{j+1}, \phi_j\rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j \\
&= \phi_{j+1} - \frac{\langle L_{j+1}, L_j - L_{j+1}\rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j \\
&= \phi_{j+1} - \frac{\langle L_{j+1}, L_j\rangle - \langle L_{j+1}, L_{j+1}\rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j \\
&= \phi_{j+1} - \left(\frac{-\langle L_{j+1}, L_{j+1}\rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j\right) \\
&= \phi_{j+1} + \frac{\langle L_{j+1}, L_{j+1}\rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j.
\end{aligned}
$$

Legendre polynomials are a set of orthogonal functions on $(-1,1)$, that is,

$$
\int_{-1}^{1} L_l(x)\cdot L_m(x)dx = \begin{cases} 0 & \text{if } l \neq m, \\ \frac{2}{2l+1} & \text{if } l = m. \end{cases}
$$

Therefore, using Lemma 1, and (4.5), we obtain

$$
\begin{aligned}
\tilde{\phi}_{j+1} &= \phi_{j+1} + \frac{\left(\frac{2}{2(j+1)+1}\right)}{\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j \\
&= \phi_{j+1} + \frac{2}{(2j+3)\langle \tilde{\phi}_j, \tilde{\phi}_j\rangle}\tilde{\phi}_j \\
&= \phi_{j+1} + \frac{2}{(2j+3)}\cdot\frac{(j+1)^2(2j+3)}{2(j+2)^2}\tilde{\phi}_k \\
&= \phi_{j+1} + \left(\frac{j+1}{j+2}\right)^2\tilde{\phi}_j \\
&= \phi_{j+1} + \left(\frac{j+1}{(j+1)+1}\right)^2\tilde{\phi}_j. \quad \square
\end{aligned}
$$

We now prove a converse of Theorem 1.

Theorem 2. If $\tilde{\phi}_0(x) = 1 - x$ and $\tilde{\phi}_i$ is defined as in (4.6) for $i = 1, 2, \ldots$, then $\langle \tilde{\phi}_k, \tilde{\phi}_j \rangle = 0$ when $j < k$.

Proof: Case 1: $j < k - 1$

$$\begin{aligned}
\langle \tilde{\phi}_k, \tilde{\phi}_j \rangle &= \left\langle \phi_k + \left(\frac{k}{k+1}\right)^2 \tilde{\phi}_{k-1}, \phi_j + \left(\frac{j}{j+1}\right)^2 \tilde{\phi}_{j-1} \right\rangle \\
&= \langle \phi_k, \phi_j \rangle + \left(\frac{j}{j+1}\right)^2 \langle \tilde{\phi}_{k-1}, \tilde{\phi}_{j-1} \rangle + \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \phi_j \rangle \\
&\quad + \left(\frac{k}{k+1}\right)^2 \left(\frac{j}{j+1}\right)^2 \langle \tilde{\phi}_{k-1}, \tilde{\phi}_{j-1} \rangle \\
&= \langle \phi_k, \phi_j \rangle + \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \phi_j \rangle + \left[\left(\frac{j}{j+1}\right)^2 \langle \tilde{\phi}_{k-1}, \tilde{\phi}_{j-1} \rangle \right] \cdot \left[1 + \left(\frac{k}{k+1}\right)^2 \right] \\
&= \langle \phi_k, \phi_j \rangle + \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \phi_j \rangle \\
&= \langle L_k - L_{k+1}, L_j - L_{j+1} \rangle + \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \phi_j \rangle \\
&= \langle L_k, L_j \rangle - \langle L_k, L_{j+1} \rangle - \langle L_{k+1}, L_j \rangle + \langle L_{k+1}, L_{j+1} \rangle + \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \phi_j \rangle \\
&= \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \phi_j \rangle \\
&= 0.
\end{aligned}$$

Case 2: $j = k - 1$

$$\begin{aligned}
\langle \tilde{\phi}_k, \tilde{\phi}_j \rangle &= \langle \phi_k, \phi_j \rangle + \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \phi_j \rangle \\
&= \langle \phi_k, \phi_{k-1} \rangle + \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \phi_{k-1} \rangle \\
&= \langle L_k - L_{k+1}, L_{k-1} - L_k \rangle + \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \phi_{k-1} \rangle \\
&= \langle L_k, L_{k-1} \rangle - \langle L_k, L_k \rangle - \langle L_{k+1}, L_{k-1} \rangle + \langle L_{k+1}, L_k \rangle + \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \phi_{k-1} \rangle \\
&= -\langle L_k, L_k \rangle + \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \phi_{k-1} \rangle \\
&= -\langle L_k, L_k \rangle + \left(\frac{k}{k+1}\right)^2 \langle \tilde{\phi}_{k-1}, \tilde{\phi}_{k-1} \rangle
\end{aligned}$$

$$\begin{aligned}
&= -\left(\frac{2}{2k+1}\right)+\left(\frac{k}{k+1}\right)^2\left(\frac{2\left((k-1)+2\right)^2}{\left((k-1)+1\right)^2\left(2\left(k-1\right)+3\right)}\right)\\
&= -\left(\frac{2}{2k+1}\right)+\left(\frac{k}{k+1}\right)^2\left(\frac{2\left(k+1\right)^2}{k^2\left(2k+1\right)}\right)\\
&= -\frac{2}{2k+1}+\frac{2}{2k+1}\\
&= 0. \quad \square
\end{aligned}$$

All orthogonal polynomials satisfy a general three-term recurrence relation that has the form

$$\beta_j\tilde{\phi}_{j+1}(x) = \left(x-\alpha_j\right)\tilde{\phi}_j(x) - \gamma_{j-1}\tilde{\phi}_{j-1}(x), \tag{4.9}$$

where α_j, β_j, and γ_j are constants. By enforcing orthogonality, we obtain the formulas

$$\alpha_j = \frac{\langle\tilde{\phi}_j, x\tilde{\phi}_j\rangle}{\langle\tilde{\phi}_j, \tilde{\phi}_j\rangle}, \tag{4.10}$$

$$\beta_j = \frac{\langle\tilde{\phi}_{j+1}, x\tilde{\phi}_j\rangle}{\langle\tilde{\phi}_{j+1}, \tilde{\phi}_{j+1}\rangle}, \tag{4.11}$$

$$\gamma_j = \frac{\langle\tilde{\phi}_{j+1}, x\tilde{\phi}_j\rangle}{\langle\tilde{\phi}_j, \tilde{\phi}_j\rangle} \tag{4.12}$$

First, we will find the value of α_j.

Theorem 3. Let α_j be defined as in (4.10). Then $\alpha_j = -\frac{1}{(j+1)(j+2)}, \forall j \geq 0$.

Proof:

Base case: When $j = 0$, we use (4.10) to obtain

$$\begin{aligned}
\alpha_0 &= \frac{\langle\tilde{\phi}_0, x\tilde{\phi}_0\rangle}{\langle\tilde{\phi}_0, \tilde{\phi}_0\rangle}\\
&= \frac{\int_{-1}^{1}\tilde{\phi}_0\cdot x\tilde{\phi}_0\,dx}{\int_{-1}^{1}\tilde{\phi}_0\cdot\tilde{\phi}_0\,dx}\\
&= \frac{\int_{-1}^{1}(1-x)\cdot x(1-x)\,dx}{\int_{-1}^{1}(1-x)\cdot(1-x)\,dx}\\
&= \frac{\int_{-1}^{1}\left(x^3-2x^2+x\right)dx}{\int_{-1}^{1}\left(x^2-2x+1\right)dx}\\
&= \frac{\left.\frac{1}{4}x^4-\frac{2}{3}x^3+\frac{1}{2}x^2\right|_{-1}^{1}}{\left.\frac{1}{3}x^3-x^2+x\right|_{-1}^{1}}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{-\frac{4}{3}}{\frac{8}{3}} \\
&= -\frac{1}{2}.
\end{aligned}
$$

For the induction hypothesis, we assume there is a $j > 0$ such that $\alpha_{j-1} = -\frac{1}{j(j+1)}$. From $\alpha_j = \frac{\langle \tilde{\phi}_j, x\tilde{\phi}_j \rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j \rangle}$ and $\tilde{\phi}_j = \phi_j + c_j \tilde{\phi}_{j-1}$ where $c_j = \left(\frac{j}{j+1}\right)^2$, we obtain

$$
\begin{aligned}
\left\langle \tilde{\phi}_j, x\tilde{\phi}_j \right\rangle &= \left\langle \phi_j + c_j \tilde{\phi}_{j-1}, x\left(\phi_j + c_j \tilde{\phi}_{j-1}\right) \right\rangle \\
&= \left\langle \phi_j + c_j \tilde{\phi}_{j-1}, x\phi_j + x c_j \tilde{\phi}_{j-1} \right\rangle \\
&= \left\langle \phi_j, x\phi_j \right\rangle + 2c_j \left\langle \tilde{\phi}_{j-1}, x\phi_j \right\rangle + c_j^2 \left\langle \tilde{\phi}_{j-1}, x\tilde{\phi}_{j-1} \right\rangle . \qquad (4.13)
\end{aligned}
$$

Now, from the recurrence relation for Legendre polynomials, we obtain

$$
\begin{aligned}
\left\langle \phi_j, x\phi_j \right\rangle &= \left\langle L_j - L_{j+1}, x\left(L_j - L_{j+1}\right) \right\rangle \\
&= \left\langle L_j - L_{j+1}, xL_j - xL_{j+1} \right\rangle \\
&= \left\langle L_j - L_{j+1}, \left(\frac{j+1}{2j+1} L_{j+1} + \frac{j}{2j+1} L_{j-1}\right) - \left(\frac{j+2}{2j+3} L_{j+2} + \frac{j+1}{2j+3} L_j\right) \right\rangle \\
&= -\left(\frac{j+1}{2j+3}\right) \left\langle L_j, L_j \right\rangle - \left(\frac{j+1}{2j+1}\right) \left\langle L_{j+1}, L_{j+1} \right\rangle \\
&= -\left(\frac{j+1}{2j+3}\right)\left(\frac{2}{2j+1}\right) - \left(\frac{j+1}{2j+1}\right)\left(\frac{2}{2j+3}\right) \\
&= -\frac{2(j+1)}{(2j+1)(2j+3)} - \frac{2(j+1)}{(2j+1)(2j+3)} \\
&= \frac{-4(j+1)}{(2j+1)(2j+3)} \qquad (4.14)
\end{aligned}
$$

and

$$
\begin{aligned}
\left\langle \tilde{\phi}_{j-1}, x\phi_j \right\rangle &= \left\langle \tilde{\phi}_{j-1}, \left(\frac{j+1}{2j+1} L_{j+1} + \frac{j}{2j+1} L_{j-1}\right) - \left(\frac{j+2}{2j+3} L_{j+2} + \frac{j+1}{2j+3} L_j\right) \right\rangle \\
&= \frac{j}{2j+1} \left\langle \tilde{\phi}_{j-1}, L_{j-1} \right\rangle - \frac{j+1}{2j+3} \left\langle \tilde{\phi}_{j-1}, L_j \right\rangle \\
&= \frac{j}{2j+1} \left\langle \tilde{\phi}_{j-1}, L_{j-1} \right\rangle - \frac{j+1}{2j+3} \left\langle \phi_{j-1} + \left(\frac{j-1}{j}\right)^2 \tilde{\phi}_{j-2}, L_j \right\rangle \\
&= \frac{j}{2j+1} \left\langle \tilde{\phi}_{j-1}, L_{j-1} \right\rangle - \frac{j+1}{2j+3} \left[\left\langle \phi_{j-1}, L_j \right\rangle + \left(\frac{j-1}{j}\right)^2 \left\langle \tilde{\phi}_{j-2}, L_j \right\rangle \right] \\
&= \frac{j}{2j+1} \left\langle \tilde{\phi}_{j-1}, L_{j-1} \right\rangle - \frac{j+1}{2j+3} \left\langle \phi_{j-1}, L_j \right\rangle \\
&= \frac{j}{2j+1} \left\langle \tilde{\phi}_{j-1}, L_{j-1} \right\rangle - \frac{j+1}{2j+3} \left\langle L_{j-1} - L_j, L_j \right\rangle
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{j}{2j+1}\left\langle\tilde{\phi}_{j-1},L_{j-1}\right\rangle - \frac{j+1}{2j+3}\left[\left\langle L_{j-1},L_j\right\rangle - \left\langle L_j,L_j\right\rangle\right] \\
&= \frac{j}{2j+1}\left\langle\tilde{\phi}_{j-1},L_{j-1}\right\rangle - \frac{j+1}{2j+3}\left(-\left\langle L_j,L_j\right\rangle\right) \\
&= \frac{j}{2j+1}\left\langle\tilde{\phi}_{j-1},L_{j-1}\right\rangle + \frac{j+1}{2j+3}\left\langle L_j,L_j\right\rangle \\
&= \frac{j}{2j+1}\left[\left\langle L_{j-1},L_{j-1}\right\rangle - c_{j-1}\left\langle L_{j-1},L_{j-1}\right\rangle\right] + \frac{j+1}{2j+3}\left\langle L_j,L_j\right\rangle \\
&= \frac{j}{2j+1}\left[\left(1-c_{j-1}\right)\left\langle L_{j-1},L_{j-1}\right\rangle\right] + \frac{j+1}{2j+3}\left\langle L_j,L_j\right\rangle \\
&= \frac{j}{2j+1}\left[\left(1-\left(\frac{j-1}{j}\right)^2\right)\left(\frac{2}{2j-1}\right)\right] + \frac{j+1}{2j+3}\left(\frac{2}{2j+1}\right) \\
&= \frac{j}{2j+1}\left(\frac{2j-1}{j^2}\right)\left(\frac{2}{2j-1}\right) + \frac{2(j+1)}{(2j+1)(2j+3)} \\
&= \frac{2}{j\,(2j+1)} + \frac{2(j+1)}{(2j+1)\,(2j+3)} \\
&= \frac{2(j^2+3j+3)}{j\,(2j+1)\,(2j+3)}. \qquad (4.15)
\end{aligned}
$$

To calculate the middle term in equation (4.13) we will multiply $2c_j$ by the result from equation (4.15):

$$
\begin{aligned}
2c_j\left\langle\tilde{\phi}_{j-1},x\phi_j\right\rangle &= 2\left(\frac{j}{j+1}\right)^2\left(\frac{2\left(j^2+3j+3\right)}{j\,(2j+1)\,(2j+3)}\right) \\
&= \frac{4j\left(j^2+3j+3\right)}{(j+1)^2\,(2j+1)\,(2j+3)}. \qquad (4.16)
\end{aligned}
$$

We rearrange the formula for α_{j-1} to obtain the following:

$$
\begin{aligned}
\left\langle\tilde{\phi}_{j-1},x\tilde{\phi}_{j-1}\right\rangle &= \alpha_{j-1}\cdot\left\langle\tilde{\phi}_{j-1},\tilde{\phi}_{j-1}\right\rangle \\
&= -\frac{1}{j\,(j+1)}\cdot\frac{2((j+1)^2}{j^2\,(2j+1)} \\
&= \frac{-2\,(j+1)}{j^3\,(2j+1)}.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
c_j^2\left\langle\tilde{\phi}_{j-1},x\tilde{\phi}_{j-1}\right\rangle &= \left(\frac{j}{j+1}\right)^4\cdot\left(\frac{-2\,(j+1)}{j^3\,(2j+1)}\right) \\
&= \frac{-2j}{(j+1)^3\,(2j+1)}. \qquad (4.17)
\end{aligned}
$$

Now we can use the results from Equations (4.14) – (4.17) to determine the numerator of α_j.

$$\begin{aligned}\left\langle \tilde{\phi}_j, x\tilde{\phi}_j \right\rangle &= \frac{-4(j+1)}{(2j+1)(2j+3)} + \frac{4j\left(j^2+3j+3\right)}{(j+1)^2(2j+1)(2j+3)} + \frac{-2j}{(j+1)^3(2j+1)} \\ &= \frac{-2(j+2)}{(j+1)^3(2j+3)}.\end{aligned}$$

Hence,

$$\begin{aligned}\alpha_j &= \frac{-2(j+2)}{(j+1)^3(2j+3)} \cdot \frac{(j+1)^2(2j+3)}{2(j+2)^2} \\ &= -\frac{1}{(j+1)(j+2)}. \quad \square\end{aligned}$$

Now, we will find the value of β_j.

Theorem 4. Let $\beta_j$be defined as in (4.11). Then $\beta_j = \frac{j+2}{2j+3}, \forall j \geq 0$.

Proof: For the base case, we consider $j = 0$:

$$\begin{aligned}\beta_0 &= \frac{\left\langle \tilde{\phi}_1, x\tilde{\phi}_0 \right\rangle}{\left\langle \tilde{\phi}_1, \tilde{\phi}_1 \right\rangle} \\ &= \frac{\int_{-1}^{1} \tilde{\phi}_1 \cdot x\tilde{\phi}_0 \, dx}{\int_{-1}^{1} \tilde{\phi}_1 \cdot \tilde{\phi}_1 \, dx} \\ &= \frac{\int_{-1}^{1} \left(-\frac{3}{2}x^2 + \frac{3}{4}x + \frac{3}{4}\right) \cdot x(1-x)\, dx}{\int_{-1}^{1} \left(-\frac{3}{2}x^2 + \frac{3}{4}x + \frac{3}{4}\right) \cdot \left(-\frac{3}{2}x^2 + \frac{3}{4}x + \frac{3}{4}\right) dx} \\ &= \frac{\int_{-1}^{1} \left(\frac{3}{2}x^4 - \frac{9}{4}x^3 + \frac{3}{4}x\right) dx}{\int_{-1}^{1} \left(\frac{9}{4}x^4 - \frac{9}{4}x^3 - \frac{27}{16}x^2 + \frac{9}{8}x + \frac{9}{16}\right) dx} \\ &= \frac{\left.\frac{3}{10}x^5 - \frac{9}{16}x^4 + \frac{3}{8}x^3\right|_{-1}^{1}}{\left.\frac{9}{20}x^5 - \frac{9}{16}x^4 - \frac{9}{16}x^3 + \frac{9}{16}x^2 + \frac{9}{16}x\right|_{-1}^{1}} \\ &= \frac{\frac{3}{5}}{\frac{9}{10}} \\ &= \frac{2}{3}.\end{aligned}$$

For the induction step, we assume there is a $j \geq 0$, such that $\beta_{j-1} = \frac{j+1}{2j+1}$. From $\beta_j = \frac{\left\langle \tilde{\phi}_{j+1}, x\tilde{\phi}_j \right\rangle}{\left\langle \tilde{\phi}_{j+1}, \tilde{\phi}_{j+1} \right\rangle}$ and $\tilde{\phi}_j = \phi_j + c_j\tilde{\phi}_{j-1}$ where $c_j = \left(\frac{j}{j+1}\right)^2$, we obtain

$$\left\langle \tilde{\phi}_{j+1}, x\tilde{\phi}_j \right\rangle = \left\langle \phi_{j+1} + c_{j+1}\tilde{\phi}_j, x\left(\phi_j + c_j\tilde{\phi}_{j-1}\right) \right\rangle$$

$$
\begin{aligned}
&= \left\langle \phi_{j+1} + c_{j+1}\tilde{\phi}_j, x\phi_j + xc_j\tilde{\phi}_{j-1} \right\rangle \\
&= \left\langle \phi_{j+1}, x\phi_j \right\rangle + c_j \left\langle \tilde{\phi}_{j-1}, x\phi_{j+1} \right\rangle + c_{j+1} \left\langle \tilde{\phi}_j, x\phi_j \right\rangle \\
&\quad + c_j c_{j+1} \left\langle \tilde{\phi}_j, x\tilde{\phi}_{j-1} \right\rangle . \qquad (4.18)
\end{aligned}
$$

Using the recurrence relation for Legendre polynomials, we obtain

$$
\begin{aligned}
\left\langle \phi_{j+1}, x\phi_j \right\rangle &= \left\langle L_{j+1} - L_{j+2}, x\left(L_j - L_{j+1}\right) \right\rangle \\
&= \left\langle L_{j+1} - L_{j+2}, xL_j - xL_{j+1} \right\rangle \\
&= \left\langle L_{j+1} - L_{j+2}, \left(\frac{j+1}{2j+1} L_{j+1} + \frac{j}{2j+1} L_{j-1} \right) - \left(\frac{j+2}{2j+3} L_{j+2} + \frac{j+1}{2j+3} L_j \right) \right\rangle \\
&= \frac{j+1}{2j+1} \left\langle L_{j+1}, L_{j+1} \right\rangle + \frac{j+2}{2j+3} \left\langle L_{j+2}, L_{j+2} \right\rangle \\
&= \frac{j+1}{2j+1} \left(\frac{2}{2j+3} \right) + \frac{j+2}{2j+3} \left(\frac{2}{2j+5} \right) \\
&= \frac{2(j+1)}{(2j+1)(2j+3)} + \frac{2(j+2)}{(2j+3)(2j+5)} \\
&= \frac{2(4j^2+12j+7)}{(2j+1)(2j+3)(2j+5)} \qquad (4.19)
\end{aligned}
$$

and

$$
\begin{aligned}
\left\langle \tilde{\phi}_{j-1}, x\phi_{j+1} \right\rangle &= \left\langle \tilde{\phi}_{j-1}, x\left(L_{j+1} - L_{j+2}\right) \right\rangle \\
&= \left\langle \tilde{\phi}_{j-1}, xL_{j+1} - xL_{j+2} \right\rangle \\
&= \left\langle \tilde{\phi}_{j-1}, \left(\frac{j+2}{2j+3} L_{j+2} + \frac{j+1}{2j+3} L_j \right) - \left(\frac{j+3}{2j+5} L_{j+3} + \frac{j+2}{2j+5} L_{j+1} \right) \right\rangle \\
&= \frac{j+2}{2j+3} \left\langle \tilde{\phi}_{j-1}, L_{j+2} \right\rangle + \frac{j+1}{2j+3} \left\langle \tilde{\phi}_{j-1}, L_j \right\rangle - \frac{j+3}{2j+5} \left\langle \tilde{\phi}_{j-1}, L_{j+3} \right\rangle \\
&\quad - \frac{j+2}{2j+5} \left\langle \tilde{\phi}_{j-1}, L_{j+1} \right\rangle \\
&= \frac{j+1}{2j+3} \left\langle \tilde{\phi}_{j-1}, L_j \right\rangle \\
&= \frac{j+1}{2j+3} \left\langle \phi_{j-1} + \left(\frac{j-1}{j} \right)^2 \tilde{\phi}_{j-2}, L_j \right\rangle \\
&= \frac{j+1}{2j+3} \left[\left\langle \phi_{j-1}, L_j \right\rangle + \left(\frac{j-1}{j} \right)^2 \left\langle \tilde{\phi}_{j-2}, L_j \right\rangle \right] \\
&= \frac{j+1}{2j+3} \left\langle \phi_{j-1}, L_j \right\rangle \\
&= \frac{j+1}{2j+3} \left\langle L_{j-1} - L_j, L_j \right\rangle \\
&= \frac{j+1}{2j+3} \left[\left\langle L_{j-1}, L_j \right\rangle - \left\langle L_j, L_j \right\rangle \right]
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{j+1}{2j+3}\left(-\left\langle L_j, L_j\right\rangle\right) \\
&= -\frac{j+1}{2j+3}\cdot\left\langle L_j, L_j\right\rangle \\
&= -\frac{j+1}{2j+3}\left(\frac{2}{2j+1}\right) \\
&= -\frac{2(j+1)}{(2j+1)(2j+3)}.
\end{aligned}
$$

We then have

$$
\begin{aligned}
c_j\left\langle\tilde{\phi}_{j-1}, x\phi_{j+1}\right\rangle &= \left(\frac{j}{j+1}\right)^2\left(-\frac{2(j+1)}{(2j+1)(2j+3)}\right) \\
&= \frac{-2j^2}{(j+1)(2j+1)(2j+3)}. \qquad (4.20)
\end{aligned}
$$

The last term in (4.18) is obtained as follows:

$$
\begin{aligned}
\left\langle\tilde{\phi}_j, x\phi_j\right\rangle &= \left\langle\tilde{\phi}_j, \left(\frac{j+1}{2j+1}L_{j+1} + \frac{j}{2j+1}L_{j-1}\right) - \left(\frac{j+2}{2j+3}L_{j+2} + \frac{j+1}{2j+3}L_j\right)\right\rangle \\
&= \frac{j+1}{2j+1}\left\langle\tilde{\phi}_j, L_{j+1}\right\rangle + \frac{j}{2j+1}\left\langle\tilde{\phi}_j, L_{j-1}\right\rangle - \frac{j+1}{2j+3}\left\langle\tilde{\phi}_j, L_j\right\rangle \\
&= \frac{j+1}{2j+1}\left(-\left\langle L_{j+1}, L_{j+1}\right\rangle\right) + \frac{j}{2j+1}\left[c_j\left(1-c_{j-1}\right)\left\langle L_{j-1}, L_{j-1}\right\rangle\right] \\
&\quad -\frac{j+1}{2j+3}\left[\left(1-c_j\right)\left\langle L_j, L_j\right\rangle\right] \\
&= -\frac{j+1}{2j+1}\left(\frac{2}{2j+3}\right) + \frac{j}{2j+1}\left(\frac{2}{(j+1)^2}\right) - \frac{j+1}{2j+3}\left(\frac{2}{(j+1)^2}\right) \\
&= -\frac{2(j+1)}{(2j+1)(2j+3)} + \frac{2j}{(j+1)^2(2j+1)} - \frac{2}{(j+1)(2j+3)} \\
&= -\frac{2(j+2)\left(j^2+j+1\right)}{(j+1)^2(2j+1)(2j+3)}.
\end{aligned}
$$

We then have

$$
\begin{aligned}
c_{j+1}\left\langle\tilde{\phi}_j, x\phi_j\right\rangle &= \left(\frac{j+1}{j+2}\right)^2\left(-\frac{2(j+2)\left(j^2+j+1\right)}{(j+1)^2(2j+1)(2j+3)}\right) \\
&= \frac{-2\left(j^2+j+1\right)}{(j+2)(2j+1)(2j+3)}. \qquad (4.21)
\end{aligned}
$$

We rearrange the formula for β_{j-1} to obtain the following:

$$
\left\langle\tilde{\phi}_j, x\tilde{\phi}_{j-1}\right\rangle = \beta_{j-1}\cdot\left\langle\tilde{\phi}_j, \tilde{\phi}_j\right\rangle
$$

$$= \frac{j+1}{2j+1} \cdot \frac{2((j+2)^2}{(j+1)^2(2j+3)}$$
$$= \frac{2(j+2)^2}{(j+1)(2j+1)(2j+3)}.$$

Therefore,

$$c_j c_{j+1} \langle \tilde{\phi}_j, x\tilde{\phi}_{j-1} \rangle = \left(\frac{j}{j+1}\right)^2 \cdot \left(\frac{j+1}{j+2}\right)^2 \cdot \left(\frac{2(j+2)^2}{(j+1)(2j+1)(2j+3)}\right)$$
$$= \frac{2j^2}{(j+1)(2j+1)(2j+3)}. \tag{4.22}$$

Now we can use the results from Equations (4.19) – (4.22) to determine the numerator of β_j.

$$\langle \tilde{\phi}_{j+1}, x\tilde{\phi}_j \rangle = \frac{2(4j^2+12j+7)}{(2j+1)(2j+3)(2j+5)} - \frac{2j^2}{(j+1)(2j+1)(2j+3)}$$
$$- \frac{2(j^2+j+1)}{(j+2)(2j+1)(2j+3)} + \frac{2j^2}{(j+1)(2j+1)(2j+3)}$$
$$= \frac{2(4j^2+12j+7)}{(2j+1)(2j+3)(2j+5)} - \frac{2(j^2+j+1)}{(j+2)(2j+1)(2j+3)}$$
$$= \frac{2(j+3)^2}{(j+2)(2j+3)(2j+5)}$$

Hence,

$$\beta_j = \frac{2(j+3)^2}{(j+2)(2j+3)(2j+5)} \cdot \frac{(j+2)^2(2j+5)}{2(j+3)^2}$$
$$= \frac{j+2}{2j+3}. \quad \square$$

Using the results from Theorem 4, we can find the value of γ_j.

Theorem 5. Let γ_j be defined as in (4.12). Then $\gamma_j = \frac{(j+1)^2(j+3)}{(j+2)^3(2j+5)}, \forall j \geq 0$.

Proof:

Base Case: When $j = 0$,

$$\gamma_0 = \frac{\langle \tilde{\phi}_1, x\tilde{\phi}_0 \rangle}{\langle \tilde{\phi}_0, \tilde{\phi}_0 \rangle}$$
$$= \frac{\int_{-1}^{1} \tilde{\phi}_1 \cdot x\tilde{\phi}_0 \, dx}{\int_{-1}^{1} \tilde{\phi}_0 \cdot \tilde{\phi}_0 \, dx}$$
$$= \frac{\int_{-1}^{1} \left(-\frac{3}{2}x^2 + \frac{3}{4}x + \frac{3}{4}\right) \cdot x(1-x)\, dx}{\int_{-1}^{1} (1-x) \cdot (1-x)\, dx}$$

$$
\begin{aligned}
&= \frac{\int_{-1}^{1}\left(\frac{3}{2}x^4 - \frac{9}{4}x^3 + \frac{3}{4}x\right)dx}{\int_{-1}^{1}\left(x^2 - 2x + 1\right)dx} \\
&= \frac{\frac{3}{10}x^5 - \frac{9}{16}x^4 + \frac{3}{8}x^3\Big|_{-1}^{1}}{\frac{1}{3}x^3 - x^2 + x\Big|_{-1}^{1}} \\
&= \frac{\frac{3}{5}}{\frac{8}{3}} \\
&= \frac{9}{40}.
\end{aligned}
$$

Induction Hypothesis: Assume there is a $j \geq 0$, such that $\gamma_{j-1} = \frac{j^2(j+2)^2}{(j+1)^3(2j+3)}$. Let $\gamma_j = \frac{\langle \tilde{\phi}_{j+1}, x\tilde{\phi}_j \rangle}{\langle \tilde{\phi}_j, \tilde{\phi}_j \rangle}$ and $\tilde{\phi}_j = \phi_j + c_j\tilde{\phi}_{j-1}$ where $c_j = \left(\frac{j}{j+1}\right)^2$. Induction Step: Notice that β_j and γ_j have the same numerator. So, we will use the induction steps found in Theorem 4. Thus,

$$
\begin{aligned}
\gamma_j &= \frac{2\,(j+3)^2}{(j+2)\,(2j+3)\,(2j+5)} \cdot \frac{(j+1)^2\,(2j+3)}{2\,(j+2)^2} \\
&= \frac{(j+1)^2(j+3)^2}{(j+2)^3(2j+5)}. \quad \square
\end{aligned}
$$

In summary, the polynomials $\tilde{\phi}_i$ satisfy the recurrence relation

$$
\frac{j+2}{j+3}\tilde{\phi}_{j+1}(x) = \left(x + \frac{1}{(j+1)(j+2)}\right)\tilde{\phi}_j(x) - \frac{j^2(j+2)^2}{(j+1)^3(2j+3)}\tilde{\phi}_{j-1}(x). \qquad (4.23)
$$

We can rewrite equation (4.8) as $\tilde{\phi}_j - c_j\tilde{\phi}_{j-1} = \phi_j$. In matrix form, we have

$$
\Phi = \tilde{\Phi}C, \quad C = \begin{bmatrix} 1 & -c_1 & & & \\ & 1 & -c_2 & & \\ & & 1 & \ddots & \\ & & & \ddots & -c_n \\ & & & & 1 \end{bmatrix}, \qquad (4.24)
$$

where $\Phi = \begin{bmatrix} \phi_0(\mathbf{x}) & \phi_1(\mathbf{x}) & \cdots & \phi_i(\mathbf{x}) \end{bmatrix}$ and $\tilde{\Phi} = \begin{bmatrix} \tilde{\phi}_0(\mathbf{x}) & \tilde{\phi}_1(\mathbf{x}) & \cdots & \tilde{\phi}_i(\mathbf{x}) \end{bmatrix}$, with $\mathbf{x}$ being a vector of at least $n+2$ Legendre-Gauss-Lobatto points. This ensures that the columns of $\tilde{\Phi}$ are orthogonal. Then, given $f \in X_{n+1}(0)$, we can obtain the coefficients $\tilde{f}_i$ in

$$
f(x) = \sum_{i=0}^{n} \tilde{f}_i\tilde{\phi}_i(x)
$$

by simply computing $\tilde{f}_i = \langle \tilde{\phi}_i, f \rangle / N_i$, where N_i is as defined in (4.5). Then the coefficients f_i in

$$
f(x) = \sum_{i=0}^{n} f_i\phi_i(x)
$$

can be obtained by solving the system $C\mathbf{f} = \tilde{\mathbf{f}}$ using back substitution, where C is as defined in (4.24). These coefficients can be used in conjunction with the discretization used in [25], which makes use of the basis $\{\phi_i\}$.

4.2 The Case $m \neq 0$

In the case where $m \neq 0$, we work with the space

$$X_N(m) = \{p \in P_N | p(-1) = p(1) = 0\}.$$

As discussed in [25], this space can easily be described in terms of Legendre polynomials:

$$X_N(m) = \operatorname{span}\{\phi_i(t) = L_i(t) - L_{i+2}(t),\ i = 0, 1, \ldots, N-2\}.$$

Applying the Gram-Schmidt process to the basis function $\{\phi_i\}$, we obtain a new set of orthogonal polynomials that will be denoted as $\{\hat{\phi}_i\}$. These basis functions are obtained in the same way as in equation (4.1). First, we let

$$\begin{aligned}
\hat{\phi}_0 &= \phi_0 \\
&= L_0 - L_2 \\
&= -\frac{3}{2}x^2 + \frac{3}{2}
\end{aligned}$$

and

$$\begin{aligned}
\hat{\phi}_1 &= \phi_1 \\
&= L_1 - L_3 \\
&= -\frac{5}{2}x^3 + \frac{5}{2}x.
\end{aligned}$$

Then, we have

$$\begin{aligned}
\hat{\phi}_2 &= \phi_2 - \frac{\langle \hat{\phi}_0, \phi_2 \rangle}{\langle \hat{\phi}_0, \hat{\phi}_0 \rangle}\hat{\phi}_0 - \frac{\langle \hat{\phi}_1, \phi_2 \rangle}{\langle \hat{\phi}_1, \hat{\phi}_1 \rangle}\hat{\phi}_1 \\
&= \phi_2 - \frac{\int_1^{-1} (\hat{\phi}_0 \cdot \phi_2)\, dx}{\int_1^{-1} (\hat{\phi}_0 \cdot \hat{\phi}_0)\, dx}\hat{\phi}_0 - \frac{\int_1^{-1} (\hat{\phi}_1 \cdot \phi_2)\, dx}{\int_1^{-1} (\hat{\phi}_1 \cdot \hat{\phi}_1)\, dx}\hat{\phi}_1 \\
&= \left(-\frac{35}{8}x^4 + \frac{21}{4}x^2 - \frac{7}{8}\right) - \frac{\frac{-2}{5}}{\frac{12}{5}}\left(-\frac{3}{2}x^2 + \frac{3}{2}\right) - \frac{0}{\frac{20}{21}}\left(-\frac{5}{2}x^3 + \frac{5}{2}x\right) \\
&= -\frac{35}{8}x^4 + \frac{21}{4}x^2 - \frac{7}{8} + \frac{1}{6}\left(-\frac{3}{2}x^2 + \frac{3}{2}\right) \\
&= -\frac{35}{8}x^4 + \frac{21}{4}x^2 - \frac{7}{8} - \frac{1}{4}x^2 + \frac{1}{4}
\end{aligned}$$

$$= \quad -\frac{35}{8}x^4 + 5x^2 - \frac{5}{8}$$

and

$$\begin{aligned}
\hat{\phi}_3 &= \phi_3 - \frac{\langle \hat{\phi}_0, \phi_3 \rangle}{\langle \hat{\phi}_0, \hat{\phi}_0 \rangle}\hat{\phi}_0 - \frac{\langle \hat{\phi}_1, \phi_3 \rangle}{\langle \hat{\phi}_1, \hat{\phi}_1 \rangle}\hat{\phi}_1 - \frac{\langle \hat{\phi}_2, \phi_3 \rangle}{\langle \hat{\phi}_2, \hat{\phi}_2 \rangle}\hat{\phi}_2 \\
&= \phi_3 - \frac{\int_1^{-1} \left(\hat{\phi}_0 \cdot \phi_3\right) dx}{\int_1^{-1} \left(\hat{\phi}_0 \cdot \hat{\phi}_0\right) dx}\hat{\phi}_0 - \frac{\int_1^{-1} \left(\hat{\phi}_1 \cdot \phi_3\right) dx}{\int_1^{-1} \left(\hat{\phi}_1 \cdot \tilde{\phi}_1\right) dx}\hat{\phi}_1 - \frac{\int_1^{-1} \left(\hat{\phi}_2 \cdot \phi_3\right) dx}{\int_1^{-1} \left(\hat{\phi}_2 \cdot \hat{\phi}_2\right) dx}\hat{\phi}_2 \\
&= \left(-\frac{63}{8}x^5 + \frac{45}{4}x^3 - \frac{27}{8}x\right) - \frac{0}{\frac{12}{5}}\left(-\frac{3}{2}x^2 + \frac{3}{2}\right) - \frac{\frac{-2}{7}}{\frac{20}{21}}\left(-\frac{5}{2}x^3 + \frac{5}{2}x\right) \\
&\quad -\frac{0}{\frac{5}{9}}\left(-\frac{35}{8}x^4 + 5x^2 - \frac{5}{8}\right) \\
&= -\frac{63}{8}x^5 + \frac{45}{4}x^3 - \frac{27}{8}x + \frac{3}{10}\left(-\frac{5}{2}x^3 + \frac{5}{2}x\right) \\
&= -\frac{63}{8}x^5 + \frac{45}{4}x^3 - \frac{27}{8}x - \frac{3}{4}x^3 + \frac{3}{4}x \\
&= -\frac{63}{8}x^5 + \frac{21}{3}x^3 - \frac{21}{8}x.
\end{aligned}$$

The graphs of the first several members of the sequence $\{\hat{\phi}_i\}$ are shown in Figure 4.2.

Again, we will compare ϕ_2 with $\hat{\phi}_2$ and ϕ_3 with $\hat{\phi}_3$ to find a general formula for the values of $\hat{\phi}_i$. We obtain the following formula

$$\begin{aligned}
\hat{\phi}_2 - \phi_2 &= -\frac{35}{8}x^4 + 5x^2 - \frac{5}{8} - \left(-\frac{35}{8}x^4 + \frac{21}{4}x - \frac{7}{8}\right) \\
&= -\frac{35}{8}x^4 + 5x^2 - \frac{5}{8} + \frac{35}{8}x^4 - \frac{21}{4}x^2 + \frac{7}{8} \\
&= -\frac{1}{4}x^2 + \frac{1}{4} \\
&= \frac{1}{6}\left(-\frac{3}{2}x^2 + \frac{3}{2}\right) \\
&= \frac{1}{6}\tilde{\phi}_0. \qquad (4.25)
\end{aligned}$$

and

$$\begin{aligned}
\hat{\phi}_3 - \phi_3 &= -\frac{63}{8}x^5 + \frac{21}{2}x^3 - \frac{21}{8}x - \left(-\frac{63}{8}x^5 + \frac{45}{4}x^3 - \frac{27}{8}x\right) \\
&= -\frac{63}{8}x^5 + \frac{21}{2}x^3 - \frac{21}{8}x + \frac{63}{8}x^5 - \frac{45}{4}x^3 + \frac{27}{8}x \\
&= -\frac{3}{4}x^3 + \frac{3}{4}x \\
&= \frac{3}{10}\left(-\frac{5}{2}x^3 + \frac{5}{2}x\right)
\end{aligned}$$

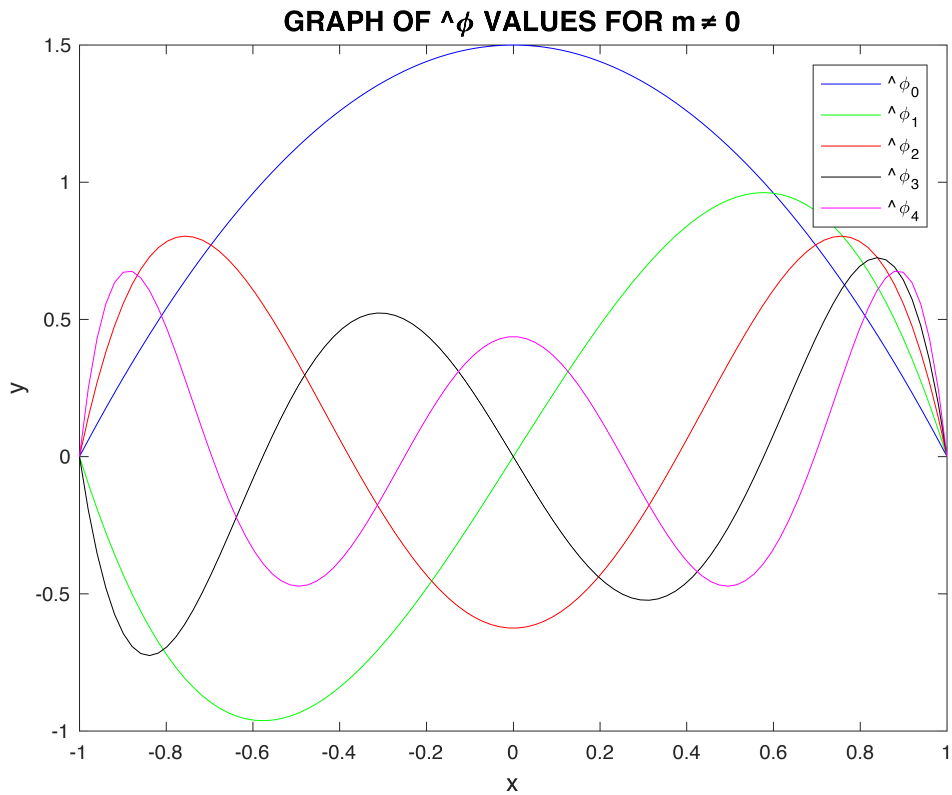


Figure 4.2: Graphs of $\hat{\phi}_i$, $i = 0,1,2,3,4$.

$$= \frac{3}{10}\hat{\phi}_1.$$

These results suggest a simple recurrence relation for $\hat{\phi}_i$ in terms of ϕ_i and $\hat{\phi}_{i-2}$, in which the coefficient of $\hat{\phi}_{i-2}$ is a ratio of triangular numbers $d_i = i(i-1)/[(i+1)(i+2)]$. We therefore define

$$\hat{\phi}_i = \phi_i - \frac{i(i-1)}{(i+1)(i+2)}\hat{\phi}_{i-2}, \quad i = 2,3,\ldots,N-2 \tag{4.26}$$

with initial conditions

$$\hat{\phi}_0 = \phi_0 = 1 - x^2, \quad \hat{\phi}_1 = \phi_1 = \frac{5}{2}(x - x^3) \tag{4.27}$$

To prove that these polynomials are actually orthogonal, we first need this result.

Lemma 2. Let $\hat{\phi}_j(x)$ be defined as in (4.26), (4.27), and $N_j = \langle \hat{\phi}_j, \tilde{\phi}_j \rangle$, $\forall\ j \geq 2$. Then

$$N_j = \frac{2\,(j+3)\,(j+4)}{(2j+5)\,(j+2)\,(j+1)}, \tag{4.28}$$

$\forall\ j \geq 2$ and

$$\hat{\phi}_j = \phi_j + \frac{j\,(j-1)}{(j+1)\,(j+2)}\hat{\phi}_{j-2}, \quad j \geq 2, \quad \hat{\phi}_j = \phi_j, \quad j \leq 1.$$

Proof: For the base case we compute N_0 and N_1 directly. We have

$$\begin{aligned} N_0 &= \left\langle \hat{\phi}_0, \hat{\phi}_0 \right\rangle \\ &= \langle \phi_0, \phi_0 \rangle \\ &= \langle L_0 - L_2,\ L_0 - L_2 \rangle \\ &= \langle L_0, L_0 \rangle + \langle L_2, L2 \rangle \\ &= 2 + \frac{2}{5} \\ &= \frac{12}{5}, \end{aligned}$$

and

$$\begin{aligned} N_1 &= \left\langle \hat{\phi}_1, \hat{\phi}_1 \right\rangle \\ &= \langle \phi_1, \phi_1 \rangle \\ &= \langle L_1 - L_3,\ L_1 - L_3 \rangle \\ &= \langle L_1, L_1 \rangle + \langle L_3, L3 \rangle \\ &= \frac{2}{3} + \frac{2}{7} \\ &= \frac{20}{21}. \end{aligned}$$

For the induction step, we assume there is a $j > 2$ such that $N_{j-2} = \frac{2j(j+3)}{(j)(j+1)(2j+3)}$. Now, we must show that the formula (4.28) is true for j. We have

$$\begin{aligned} N_j = \left\langle \hat{\phi}_j, \hat{\phi}_j \right\rangle &= \left\langle \phi_j + \frac{j(j-1)}{(j+1)(j+2)} \hat{\phi}_{j-2}, \phi_j + \frac{j(j-1)}{(j+1)(j+2)} \hat{\phi}_{j-2} \right\rangle \\ &= \left\langle \phi_j, \phi_j \right\rangle + \frac{2j(j-1)}{(j+1)(j+2)} \left\langle \phi_j, \hat{\phi}_{j-2} \right\rangle + \left(\frac{j(j-1)}{(j+1)(j+2)} \right)^2 \left\langle \hat{\phi}_{j-2}, \hat{\phi}_{j-2} \right\rangle \\ &= \frac{4(2j+3)}{(2j+1)(2j+5)} + \frac{2j(j-1)}{(j+1)(j+2)} \left(-\left\langle L_j, L_j \right\rangle \right) \\ &\quad + \left(\frac{j(j-1)}{(j+1)(j+2)} \right)^2 N_{j-2} \\ &= \frac{4(2j+3)}{(2j+1)(2j+5)} + \frac{2j(j-1)}{(j+1)(j+2)} \left(\frac{2}{2j+1} \right) + \left(\frac{j(j-1)}{(j+1)(j+2)} \right)^2 N_{j-2} \\ &= \frac{4(2j+3)}{(2j+1)(2j+5)} + \frac{4j(j-1)}{(j+1)(j+2)(2j+1)} + \left(\frac{j(j-1)}{(j+1)(j+2)} \right)^2 N_{j-2} \\ &= \frac{24\left(j^2+3j+1\right)}{(j+1)(j+2)(2j+1)(2j+5)} + \left(\frac{j(j-1)}{(j+1)(j+2)} \right)^2 \left[\frac{2(j+1)(j+2)}{j(j-1)(2j+1)} \right] \\ &= \frac{24\left(j^2+3j+1\right)}{(j+1)(j+2)(2j+1)(2j+5)} + \frac{2j(j-1)}{(j+1)(j+2)(2j+1)}. \end{aligned}$$

$$
\begin{aligned}
&= \frac{2\left(j^2+7j+12\right)}{(j+1)(j+2)(2j+5)} \\
&= \frac{2\left(j^2+7j+12\right)}{(j+1)(j+2)(2j+5)} \\
&= \frac{2(j+3)(j+4)}{(j+1)(j+2)(2j+1)}. \quad \square
\end{aligned}
$$

Theorem 6. Let $\hat{\phi}_i$ be obtained by orthogonalizing ϕ_i against $\hat{\phi}_0$, $\hat{\phi}_1, \ldots$. Then $\hat{\phi}_0 = \phi_0$, $\hat{\phi}_1 = \phi_1$, and

$$\hat{\phi}_j = \phi_j + d_j \hat{\phi}_{j-2}, \quad j \geq 2, \tag{4.29}$$

where $d_j = \frac{j(j-1)}{(j+1)(j+2)}$.

Proof: For the base case, we first show that $\hat{\phi}_0 = \phi_0$ and $\hat{\phi}_1 = \phi_1$ are already orthogonal. We have

$$
\begin{aligned}
\left\langle \hat{\phi}_1, \hat{\phi}_0 \right\rangle &= \langle L_1 - L_3, L_0 - L_2 \rangle \\
&= \langle L_0, L_1 \rangle - \langle L_1, L_2 \rangle - \langle L_0, L_3 \rangle + \langle L_2, L_3 \rangle \\
&= 0.
\end{aligned}
$$

Next, we show directly that the theorem holds when $j = 2$:

$$
\begin{aligned}
\hat{\phi}_2 &= \phi_2 - \frac{\left\langle \hat{\phi}_0, \phi_2 \right\rangle}{\left\langle \hat{\phi}_0, \hat{\phi}_0 \right\rangle} \cdot \hat{\phi}_0 - \frac{\left\langle \hat{\phi}_1, \phi_2 \right\rangle}{\left\langle \hat{\phi}_1, \hat{\phi}_1 \right\rangle} \cdot \hat{\phi}_1 \\
&= -\frac{35}{8}x^4 - \frac{21}{4}x^2 - \frac{7}{8} - \frac{-\frac{2}{5}}{\frac{12}{5}} \cdot \left(-\frac{3}{2}x^2 + \frac{3}{2}\right) - \frac{0}{\frac{20}{21}} \cdot \left(-\frac{5}{2}x^3 + \frac{5}{2}x\right) \\
&= -\frac{35}{8}x^4 - \frac{21}{4}x^2 - \frac{7}{8} + \frac{1}{6} \cdot \left(-\frac{3}{2}x^2 + \frac{3}{2}\right) \\
&= \phi_2 + \frac{1}{6}\hat{\phi}_0.
\end{aligned} \tag{4.30}
$$

For the induction step, we assume that $\hat{\phi}_0, \ldots, \hat{\phi}_{j-1}$ are orthogonal, where $j \geq 2$, and that

$$\hat{\phi}_j = \phi_j + d_j \hat{\phi}_{j-2}, \tag{4.31}$$

where $d_j = \frac{j(j-1)}{(j+1)(j+2)}$. Then

$$
\begin{aligned}
\hat{\phi}_{j+1} &= \phi_{j+1} + \sum_{k=0}^{j} \frac{\left\langle \hat{\phi}_k, \phi_{j+1} \right\rangle}{\left\langle \hat{\phi}_{k-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{k-1} \\
&= \phi_{j+1} - \frac{\left\langle \hat{\phi}_{j-1}, \phi_{j+1} \right\rangle}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1}
\end{aligned}
$$

$$
\begin{aligned}
&= \phi_{j+1} - \frac{\left\langle \hat{\phi}_{j-1}, L_{j+1} - L_{j+3} \right\rangle}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1} \\
&= \phi_{j+1} - \frac{\left\langle \hat{\phi}_{j-1}, L_{j+1} \right\rangle - \left\langle \hat{\phi}_{j-1}, L_{j+3} \right\rangle}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1} \\
&= \phi_{j+1} - \frac{\left\langle \hat{\phi}_{j-1}, L_{j+1} \right\rangle}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1} \\
&= \phi_{j+1} - \frac{\left\langle \phi_{j-1} + c_{j-1} \tilde{\phi}_{j-3}, L_{j+1} \right\rangle}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1} \\
&= \phi_{j+1} - \frac{\left\langle \phi_{j-1}, L_{j+1} \right\rangle + c_{j-1} \left\langle \hat{\phi}_{j-3}, L_{j+1} \right\rangle}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1} \\
&= \phi_{j+1} - \frac{\left\langle \phi_{j-1}, L_{j+1} \right\rangle}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1} \\
&= \phi_{j+1} - \frac{\left\langle L_{j-1} - L_{j+1}, L_{j+1} \right\rangle}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1} \\
&= \phi_{j+1} - \frac{\left\langle L_{j-1}, L_{j+1} \right\rangle - \left\langle L_{j+1}, L_{j+1} \right\rangle}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1} \\
&= \phi_{j+1} - \frac{\left(- \left\langle L_{j+1}, L_{j+1} \right\rangle \right)}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1} \\
&= \phi_{j+1} + \frac{\left\langle L_{j+1}, L_{j+1} \right\rangle}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1} \\
&= \phi_{j+1} + \frac{\frac{2}{(2j+3)}}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1} \\
&= \phi_{j+1} + \frac{2}{2j+3} \cdot \frac{1}{\left\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1} \right\rangle} \hat{\phi}_{j-1}
\end{aligned}
$$

Using Lemma 2, we obtain

$$
\begin{aligned}
\hat{\phi}_{j+1} &= \phi_{j+1} + \frac{2}{2j+3} \cdot \frac{j(j+1)(2j+3)}{2(j+2)(j+3)} \hat{\phi}_{j-1} \\
&= \phi_{j+1} + \frac{j(j+1)}{(j+2)(j+3)} \hat{\phi}_{j-1}. \quad \square
\end{aligned}
$$

We can now confirm that the polynomials defined using the recurrence (4.29) are orthogonal.

Theorem 7. Let $\hat{\phi}_k$ be defined as follows:

$$
\hat{\phi}_k = \phi_k + \frac{k(k-1)}{(k+1)(k+2)} \hat{\phi}_{k-2}, \ k \geq 2, \ \hat{\phi}_k = \phi_k, \ k \leq 1.
$$

Then $\left\langle \hat{\phi}_k, \hat{\phi}_j \right\rangle = 0$ for $j \neq k$.

Proof: We will show that for each $k \geq 0$, $\left\langle \hat{\phi}_k, \hat{\phi}_j \right\rangle = 0$ for $0 \leq j < k$. The case $k = 1$ was handled in the proof of Theorem 6. Proceeding by induction, we assume $\hat{\phi}_0, \ldots, \hat{\phi}_{k-1}$ are all orthogonal, and show that $\left\langle \hat{\phi}_k, \hat{\phi}_j \right\rangle = 0$ for $j = 0, 1, \ldots, k-1$.

Case 1: $j < k-2$

$$\begin{aligned}
\left\langle \hat{\phi}_k, \tilde{\phi}_j \right\rangle &= \left\langle \phi_k, \hat{\phi}_j \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle \phi_{k-2}, \hat{\phi}_j \right\rangle \\
&= \left\langle L_k - L_{k+2}, \hat{\phi}_j \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle L_{k-2} - L_k, \hat{\phi}_j \right\rangle \\
&= \left\langle L_k, \hat{\phi}_j \right\rangle - \left\langle L_{k+2}, \hat{\phi}_j \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left[\left\langle L_{k-2}, \hat{\phi}_j \right\rangle - \left\langle L_k, \hat{\phi}_j \right\rangle \right] \\
&= 0.
\end{aligned}$$

Case 2: $j = k-2$

$$\begin{aligned}
\left\langle \hat{\phi}_k, \hat{\phi}_j \right\rangle &= \left\langle \phi_k, \hat{\phi}_j \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle \phi_{k-2}, \hat{\phi}_j \right\rangle \\
&= \left\langle \phi_k, \tilde{\phi}_{k-2} \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle \phi_{k-2}, \hat{\phi}_{k-2} \right\rangle \\
&= \left\langle L_k - L_{k+2}, \hat{\phi}_{k-2} \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle \hat{\phi}_{k-2}, \hat{\phi}_{k-2} \right\rangle \\
&= \left\langle L_k, \hat{\phi}_{k-2} \right\rangle - \left\langle L_{k+2}, \hat{\phi}_{k-2} \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle \hat{\phi}_{k-2}, \hat{\phi}_{k-2} \right\rangle \\
&= \left\langle L_k, \hat{\phi}_{k-2} \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle \hat{\phi}_{k-2}, \hat{\phi}_{k-2} \right\rangle \\
&= -\left\langle L_k, L_k \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle \hat{\phi}_{k-2}, \hat{\phi}_{k-2} \right\rangle \\
&= -\frac{2}{2k+1} + \frac{k(k-1)}{(k+1)(k+2)} \left(\frac{2(k+1)(k+2)}{k(k-1)(2k+1)} \right) \\
&= -\frac{2}{2k+1} + \frac{2}{2k+1} \\
&= 0.
\end{aligned}$$

Case 3: $j = k-1$

$$\begin{aligned}
\left\langle \hat{\phi}_k, \hat{\phi}_j \right\rangle &= \left\langle \phi_k, \hat{\phi}_j \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle \phi_{k-2}, \hat{\phi}_j \right\rangle \\
&= \left\langle \phi_k, \hat{\phi}_{k-1} \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle \phi_{k-2}, \hat{\phi}_{k-1} \right\rangle \\
&= \left\langle L_k - L_{k+2}, \hat{\phi}_{k-1} \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle \phi_{k-2}, \tilde{\phi}_{k-1} \right\rangle \\
&= \left\langle L_k, \tilde{\phi}_{k-1} \right\rangle - \left\langle L_{k+2}, \hat{\phi}_{k-1} \right\rangle + \frac{k(k-1)}{(k+1)(k+2)} \left\langle \phi_{k-2}, \hat{\phi}_{k-1} \right\rangle
\end{aligned}$$

$$= \quad 0.$$

If $k = 2$, then the steps are the same except that the term with $\hat{\phi}_{k-3}$ is not present. $\square$

Like all families of orthogonal polynomials, the $\hat{\phi}_k$ satisfy the recurrence relation

$$\beta_j \hat{\phi}_{j+1}(x) = (x - \alpha_j)\hat{\phi}_j(x) - \gamma_{j-1}\hat{\phi}_{j-1}(x). \tag{4.32}$$

By analogy with (4.10), (4.11), and (4.12), we have

$$\alpha_j = \frac{\langle \hat{\phi}_j, x\hat{\phi}_j \rangle}{\langle \hat{\phi}_j, \hat{\phi}_j \rangle}, \tag{4.33}$$

$$\beta_j = \frac{\langle \hat{\phi}_{j+1}, x\hat{\phi}_j \rangle}{\langle \hat{\phi}_{j+1}, \hat{\phi}_{j+1} \rangle}, \tag{4.34}$$

$$\gamma_j = \frac{\langle \hat{\phi}_{j+1}, x\hat{\phi}_j \rangle}{\langle \hat{\phi}_j, \hat{\phi}_j \rangle}. \tag{4.35}$$

Because $\hat{\phi}_j$ contains only terms of odd degree, if j is odd and of even degree if j is even, just like the Legendre polynomials, it is easily shown that $\alpha_j = 0$ for $j = 1, 2, \ldots$. We will now find the value of β_j and γ_j.

Theorem 8. Let β_j be defined as in (4.34). Then $\beta_j = \frac{j+3}{2j+5}, \quad \forall j \geq 0$.

Proof: We show the base case $j = 0$ directly:

$$\begin{aligned}
\beta_0 &= \frac{\langle \hat{\phi}_1, x\hat{\phi}_0 \rangle}{\langle \hat{\phi}_1, \hat{\phi}_1 \rangle} \\
&= \frac{\int_{-1}^{1} \hat{\phi}_1 \cdot x\hat{\phi}_0 \, dx}{\int_{-1}^{1} \hat{\phi}_1 \cdot \hat{\phi}_1 \, dx} \\
&= \frac{\int_{-1}^{1} \left(-\frac{5}{2}x^3 + \frac{5}{2}x\right) \cdot x\left(-\frac{3}{2}x^2 + \frac{3}{2}\right) dx}{\int_{-1}^{1} \left(-\frac{5}{2}x^3 + \frac{5}{2}x\right) \cdot \left(-\frac{5}{2}x^3 + \frac{5}{2}x\right) dx} \\
&= \frac{\int_{-1}^{1} \left(\frac{15}{4}x^6 - \frac{15}{2}x^4 + \frac{15}{4}x^2\right) dx}{\int_{-1}^{1} \left(\frac{25}{4}x^6 - \frac{25}{2}x^4 + \frac{25}{4}x^2\right) dx} \\
&= \frac{\left.\frac{15}{28}x^7 - \frac{3}{2}x^5 + \frac{5}{4}x^3\right|_{-1}^{1}}{\left.\frac{25}{28}x^7 - \frac{5}{2}x^5 + \frac{25}{12}x^3\right|_{-1}^{1}} \\
&= \frac{\frac{4}{7}}{\frac{20}{21}} \\
&= \frac{3}{5}.
\end{aligned}$$

For the induction step, we assume there is a $j \geq 0$ such that $\beta_{j-1} = \frac{j+2}{2j+3}$. Then, using (4.34), we have $\beta_j = \frac{\langle \hat{\phi}_{j+1}, x\hat{\phi}_j \rangle}{\langle \hat{\phi}_{j+1}, \hat{\phi}_{j+1} \rangle}$ and $\hat{\phi}_j = \phi_j + d_j \hat{\phi}_{j-2}$ where $d_j = \frac{j(j-1)}{(j+1)(j+2)}$. For the numerator, we have

$$
\begin{aligned}
\left\langle \hat{\phi}_{j+1}, x\hat{\phi}_j \right\rangle &= \left\langle \phi_{j+1} + d_{j+1}\hat{\phi}_j, x\left(\phi_j + d_j\hat{\phi}_{j-2}\right) \right\rangle \\
&= \left\langle \phi_{j+1} + d_{j+1}\tilde{\phi}_j, x\phi_j + xd_j\hat{\phi}_{j-2} \right\rangle \\
&= \left\langle \phi_{j+1}, x\phi_j \right\rangle + d_j \left\langle \hat{\phi}_{j-2}, x\phi_{j+1} \right\rangle + d_{j+1} \left\langle \hat{\phi}_{j-1}, x\phi_j \right\rangle \\
&\quad + d_j d_{j+1} \left\langle \hat{\phi}_{j-1}, x\hat{\phi}_{j-2} \right\rangle. \qquad (4.36)
\end{aligned}
$$

We now compute each part of the numerator as follows:

$$
\begin{aligned}
\left\langle \phi_{j+1}, x\phi_j \right\rangle &= \left\langle L_{j+1} - L_{j+3}, x\left(L_j - L_{j+2}\right) \right\rangle \\
&= \left\langle L_{j+1} - L_{j+3}, xL_j - xL_{j+2} \right\rangle \\
&= \left\langle L_{j+1} - L_{j+3}, \left(\frac{j+1}{2j+1}L_{j+1} + \frac{j}{2j+1}L_{j-1}\right) - \left(\frac{j+3}{2j+5}L_{j+3} + \frac{j+2}{2j+5}L_{j+1}\right) \right\rangle \\
&= \frac{j+1}{2j+1}\left\langle L_{j+1}, L_{j+1} \right\rangle - \frac{j+2}{2j+5}\left\langle L_{j+1}, L_{j+1} \right\rangle + \frac{j+3}{2j+5}\left\langle L_{j+3}, L_{j+3} \right\rangle \\
&= \frac{j+1}{2j+1}\left(\frac{2}{2j+3}\right) - \frac{j+2}{2j+5}\left(\frac{2}{2j+3}\right) + \frac{j+3}{2j+5}\left(\frac{2}{2j+7}\right) \\
&= \frac{2(j+1)}{(2j+1)(2j+3)} - \frac{2(j+2)}{(2j+3)(2j+5)} + \frac{2(j+3)}{(2j+5)(2j+7)} \\
&= \frac{2(j+2)}{(2j+1)(2j+7)}. \qquad (4.37)
\end{aligned}
$$

$$
\begin{aligned}
\left\langle \hat{\phi}_{j-2}, x\phi_{j+1} \right\rangle &= \left\langle \hat{\phi}_{j-2}, \left(\frac{j+2}{2j+3}L_{j+2} + \frac{j+1}{2j+3}L_j\right) - \left(\frac{j+4}{2j+7}L_{j+4} + \frac{j+3}{2j+7}L_{j+2}\right) \right\rangle \\
&= \frac{j+1}{2j+3}\left\langle \tilde{\phi}_{j-2}, L_j \right\rangle \\
&= \frac{j+1}{2j+3}\left(-\left\langle L_j, L_j \right\rangle\right) \\
&= -\frac{j+1}{2j+3} \cdot \left\langle L_j, L_j \right\rangle \\
&= -\frac{j+1}{2j+3}\left(\frac{2}{2j+1}\right) \\
&= -\frac{2(j+1)}{(2j+1)(2j+3)}.
\end{aligned}
$$

Then

$$
d_j \left\langle \hat{\phi}_{j-2}, x\phi_{j+1} \right\rangle = \left(\frac{j(j-1)}{(j+1)(j+2)}\right) \cdot \left(-\frac{2(j+1)}{(2j+1)(2j+3)}\right)
$$

$$= \frac{-2j(j-1)}{(j+2)(2j+1)(2j+3)}. \tag{4.38}$$

For the third term in (4.36), we have

$$\begin{aligned}
\langle \hat{\phi}_{j-1}, x\phi_j \rangle &= \left\langle \hat{\phi}_{j-1}, \left(\frac{j+1}{2j+1}L_{j+1} + \frac{j}{2j+1}L_{j-1}\right) - \left(\frac{j+3}{2j+5}L_{j+3} + \frac{j+2}{2j+5}L_{j+1}\right)\right\rangle \\
&= \frac{j+1}{2j+1}\langle \hat{\phi}_{j-1}, L_{j+1}\rangle + \frac{j}{2j+1}\langle \hat{\phi}_{j-1}, L_{j-1}\rangle - \frac{j+2}{2j+5}\langle \hat{\phi}_{j-1}, L_{j+1}\rangle \\
&= \frac{j+1}{2j+1}\left(-\langle L_{j+1}, L_{j+1}\rangle\right) + \frac{j}{2j+1}\left[(1-d_{j-1})\cdot\langle L_{j-1}, L_{j-1}\rangle\right] \\
&\quad -\frac{j+2}{2j+5}\left[-\langle L_{j+1}, L_{j+1}\rangle\right] \\
&= -\frac{j+1}{2j+1}\left(\frac{2}{2j+3}\right) + \frac{j}{2j+1}\left(\frac{4}{j(j+1)}\right) - \frac{j+2}{2j+3}\left(\frac{2}{2j+3}\right) \\
&= -\frac{2(j+1)}{(2j+1)(2j+3)} + \frac{4}{(j+1)(2j+1)} - \frac{2(j+2)}{(2j+3)(2j+5)} \\
&= \frac{6(j+3)}{(j+1)(2j+1)(2j+5)},
\end{aligned}$$

and therefore

$$\begin{aligned}
d_{j+1}\langle \hat{\phi}_{j-1}, x\phi_j \rangle &= \frac{j(j+1)}{(j+2)(j+3)}\cdot\left(\frac{6(j+3)}{(j+1)(2j+1)(2j+5)}\right) \\
&= \frac{6j}{(j+2)(2j+1)(2j+5)}.
\end{aligned} \tag{4.39}$$

We rearrange the formula for β_{j-2} to get the following:

$$\begin{aligned}
\langle \hat{\phi}_{j-1}, x\hat{\phi}_{j-2} \rangle &= \beta_{j-2}\cdot\langle \hat{\phi}_{j-1}, \hat{\phi}_{j-1}\rangle \\
&= \frac{j+1}{2j+1}\cdot\frac{2((j+2)(j+3)}{j(j+1)(2j+3)} \\
&= \frac{2(j+2)(j+3)}{j(2j+1)(2j+3)}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
d_j d_{j+1}\langle \hat{\phi}_{j-1}, x\hat{\phi}_{j-2} \rangle &= \left(\frac{j(j-1)}{(j+1)(j+2)}\right)\cdot\left(\frac{j(j+1)}{(j+2)(j+3)}\right)\cdot\left(\frac{2(j+2)(j+3)}{j(2j+1)(2j+3)}\right) \\
&= \frac{2j(j-1)}{(j+1)(2j+1)(2j+3)}.
\end{aligned} \tag{4.40}$$

Now we can use the results from Equations (4.37) – (4.40) to determine the numerator of β_j.

$$\langle \hat{\phi}_{j+1}, x\hat{\phi}_j \rangle = \frac{2(j+2)}{(2j+1)(2j+7)} - \frac{2j(j-1)}{(j+2)(2j+1)(2j+3)} + \frac{6j}{(j+2)(2j+1)(2j+5)}$$

$$
\begin{aligned}
& +\frac{2j(j-1)}{(j+1)(2j+1)(2j+3)} \\
= \quad & \frac{6(j+4)}{(j+2)(2j+3)(2j+7)}+\frac{6j}{(j+2)(2j+1)(2j+5)} \\
& +\frac{2j(j-1)}{(j+1)(2j+1)(2j+3)} \\
= \quad & \frac{2(j+4)(j+5)}{(j+2)(2j+5)(2j+7)} \qquad (4.41)
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\beta_j \quad = \quad & \frac{2(j+4)(j+5)}{(j+2)(2j+5)(2j+7)} \cdot \frac{(j+2)(j+3)(2j+7)}{2(j+4)(j+5)} \\
= \quad & \frac{j+3}{2j+5}. \quad \square
\end{aligned}
$$

From (4.35), (4.41), and Lemma 2, we obtain

$$
\begin{aligned}
\gamma_j \quad = \quad & \frac{2(j+4)(j+5)}{(j+2)(2j+5)(2j+7)} \cdot \frac{(j+1)(j+2)(2j+5)}{2(j+3)(j+4)} \\
= \quad & \frac{(j+1)(j+5)}{(j+3)(2j+7)}. \qquad (4.42)
\end{aligned}
$$

In summary, we have

$$
\frac{j+3}{2j+5}\hat{\phi}_{j+1}(x) = x\hat{\phi}_j(x) - \frac{j(j+4)}{(j+2)(2j+5)}\hat{\phi}_{j-1}(x). \qquad (4.43)
$$

Equation (4.31) can be rewritten as $\phi_j = \hat{\phi}_j - d_j\hat{\phi}_{j-2}$. Now, we have the system

$$
\Phi = \hat{\Phi}D, \quad D = \begin{bmatrix} 1 & 0 & -d_2 & & & \\ & 1 & 0 & -d_3 & & \\ & & 1 & 0 & \ddots & \\ & & & 1 & \ddots & -d_n \\ & & & & \ddots & 0 \\ & & & & & 1 \end{bmatrix}
$$

where $\Phi = \begin{bmatrix} \phi_0(\mathbf{x}) & \phi_1(\mathbf{x}) & \cdots & \phi_i(\mathbf{x}) \end{bmatrix}$ and $\hat{\Phi} = \begin{bmatrix} \hat{\phi}_0(\mathbf{x}) & \hat{\phi}_1(\mathbf{x}) & \cdots & \hat{\phi}_i(\mathbf{x}) \end{bmatrix}$, with $\mathbf{x}$ being a vector of at least $n+3$ Legendre-Gauss-Lobatto points. This ensures that the columns of $\hat{\Phi}$ are orthogonal.

Then, given $f \in X_{n+2}(m)$, we can obtain the coefficients $\tilde{f}_i$ in

$$
f(x) = \sum_{i=0}^{n} \hat{f}_i\hat{\phi}_i(x)
$$

by simply computing $\hat{f}_i = \langle \hat{\phi}_i, f \rangle / N_i$, where N_i is as defined in (4.28). Then the coefficients f_i in

$$f(x) = \sum_{i=0}^{n} f_i \phi_i(x)$$

can be obtained by solving the system $D\mathbf{f} = \hat{\mathbf{f}}$ using back substitution, where D is as defined in (4.2). These coefficients can be used in conjunction with the discretization used in [25], which makes use of the basis $\{\phi_i\}$.

4.3 Boundary Condition $p(1) = p'(1) = 0$

In the case where we have a derivative on the boundary, we work with the space

$$X_N(m) = \{p \in P_N | p(1) = p'(1) = 0\}. \tag{4.44}$$

This space can easily be described in terms of Legendre polynomials:

$$X_N(m) = \text{span}\{\phi_i(t) = L_i(x) + b_i L_{i+1}(x) + c_i L_{i+2}(x) \quad i = 0, 1, \ldots, N-2\}.$$

To satisfy the boundary conditions in equation (4.44), we need recurrence relations that will satisfy $p'(1) = 0$. We begin by letting

$$\begin{aligned} \phi_0 &= L_0(x) + b_0 L_1(x) + c_0 L_2(x) \\ &= 1 + b_0 x + c_0 \left(\frac{3}{2}x^2 - \frac{1}{2} \right), \end{aligned} \tag{4.45}$$

so then

$$\begin{aligned} \phi_1 &= L_1(x) + b_1 L_2(x) + c_1 L_3(x) \\ &= x + b_1 \left(\frac{3}{2}x^2 - \frac{1}{2} \right) + c_1 \left(\frac{5}{2}x^3 - \frac{3}{2}x \right), \end{aligned} \tag{4.46}$$

and

$$\begin{aligned} \phi_2 &= L_2(x) + b_2 L_3(x) + c_2 L_4(x) \\ &= \frac{3}{2}x^2 - \frac{1}{2} + b_2 \left(\frac{5}{2}x^3 - \frac{3}{2}x \right) + c_2 \left(\frac{35}{8}x^4 - \frac{30}{8}x^2 + \frac{3}{8} \right). \end{aligned} \tag{4.47}$$

For equations (4.45), (4.46), and (4.47) we will impose the first boundary condition $p(1) = 0$. So, we have

$$\phi_0(x) = 1 + b_0 x + c_0 \left(\frac{3}{2}x^2 - \frac{1}{2} \right)$$

$$\begin{aligned}\phi_0(1) &= 1+b_0\cdot 1+c_0\left(\frac{3}{2}(1)^2-\frac{1}{2}\right)\\ 0 &= 1+b_0+c_0,\end{aligned}\tag{4.48}$$

and

$$\begin{aligned}\phi_1(x) &= x+b_1\left(\frac{3}{2}x^2-\frac{1}{2}\right)+c_1\left(\frac{5}{2}x^3-\frac{3}{2}x\right)\\ \phi_1(1) &= 1+b_1\left(\frac{3}{2}(1)^2-\frac{1}{2}\right)+c_1\left(\frac{5}{2}(1)^3-\frac{3}{2}(1)\right)\\ 0 &= 1+b_1+c_1,\end{aligned}\tag{4.49}$$

and

$$\begin{aligned}\phi_2(x) &= \frac{3}{2}x^2-\frac{1}{2}+b_2\left(\frac{5}{2}x^3-\frac{3}{2}x\right)+c_2\left(\frac{35}{8}x^4-\frac{30}{8}x^2+\frac{3}{8}\right)\\ \phi_2(1) &= \frac{3}{2}(1)^2-\frac{1}{2}+b_2\left(\frac{5}{2}(1)^3-\frac{3}{2}(1)\right)+c_2\left(\frac{35}{8}(1)^4-\frac{30}{8}(1)^2+\frac{3}{8}\right)\\ 0 &= 1+b_2+c_2.\end{aligned}\tag{4.50}$$

Now, we will impose the second boundary condition $p'(1)=0$ and use the formula $L_j'(1)=\frac{j(j+1)}{2}$. Then

$$\begin{aligned}\phi_0'(x) &= L_0'(1)+b_0L_1'(1)+c_0L_2'(1)\\ \phi_0'(1) &= 0+b_0\cdot\frac{1(1+1)}{2}+c_0\cdot\frac{2(2+1)}{2}\\ 0 &= b_0+3c_0,\end{aligned}\tag{4.51}$$

and

$$\begin{aligned}\phi_1'(x) &= L_1'(x)+b_1L_2'(x)+c_1L_3'(x)\\ \bar{\phi}_1'(x) &= \frac{1(1+1)}{2}+b_1\cdot\frac{2(2+1)}{2}+c_1\cdot\frac{3(3+1)}{2}\\ 0 &= 1+3b_1+6c_1,\end{aligned}\tag{4.52}$$

and

$$\begin{aligned}\phi_2(x) &= L_2'(x)+b_2L_3'(x)+c_2L_4'(x)\\ \phi_2'(x) &= \frac{2(2+1)}{2}+b_2\cdot\frac{3(3+1)}{2}+c_2\cdot\frac{4(4+1)}{2}\\ 0 &= 3+6b_2+10c_2\end{aligned}\tag{4.53}$$

Combining $\phi_0(x)$ and $\phi_0'(x)$ into the following system of equations, we have

$$\begin{bmatrix}1 & 1\\ 1 & 3\end{bmatrix}\begin{bmatrix}b_0\\ c_0\end{bmatrix}=\begin{bmatrix}-1\\ 0\end{bmatrix}.$$

This system can be solved using Gaussian elimination which yields $b_0 = -\frac{3}{2}$ and $c_0 = \frac{1}{2}$. The system of equations formed by $\phi_1(x)$ and $\phi_1'(x)$ can be solved the same way.

$$\begin{bmatrix} 1 & 1 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} b_1 \\ c_1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix},$$

yields $b_1 = -\frac{5}{3}$, and $c_1 = \frac{2}{3}$. Therefore,

$$\begin{aligned} \phi_0(x) &= 1 - \frac{3}{2}x + \frac{1}{2} \cdot \left(\frac{3}{2}x^2 - \frac{1}{2} \right) \\ &= 1 - \frac{3}{2}x + \frac{3}{4}x^2 - \frac{1}{4} \\ &= \frac{3}{4}x^2 - \frac{3}{2}x + \frac{3}{4} \\ &= \frac{3}{4}\left(x^2 - 2x + 1\right) \\ &= \frac{3}{4}(x-1)^2 \end{aligned}$$

and

$$\begin{aligned} \phi_1 &= x + b_1 \left(\frac{3}{2}x^2 - \frac{1}{2} \right) + c_1 \left(\frac{5}{2}x^3 - \frac{3}{2}x \right) \\ &= x - \frac{5}{3} \cdot \left(\frac{3}{2}x^2 - \frac{1}{2} \right) + \frac{2}{3} \cdot \left(\frac{5}{2}x^3 - \frac{3}{2}x \right) \\ &= x - \frac{5}{2}x^2 + \frac{5}{6} + \frac{5}{3}x^3 - x \\ &= \frac{5}{3}x^3 - \frac{5}{2}x^2 + \frac{5}{6}. \end{aligned}$$

By applying the Gram-Schmidt process to $\{\phi_i\}$, we obtain a new set of orthogonal polynomials that will be denoted by $\bar{\phi}_i$. To start the sequence $\{\bar{\phi}_i\}$, we let

$$\begin{aligned} \bar{\phi}_0 &= \phi_0 \\ &= \frac{3}{4}(x-1)^2, \end{aligned}$$

and

$$\begin{aligned} \bar{\phi}_1 &= \bar{\phi}_0 - \frac{\langle \bar{\phi}_0, \phi_1 \rangle}{\langle \bar{\phi}_0, \bar{\phi}_0 \rangle} \bar{\phi}_0 \\ &= \frac{5}{3}x^3 - \frac{5}{2}x^2 + \frac{5}{6} + \frac{10}{27} \cdot \left(\frac{3}{4}x^2 - \frac{3}{2}x + \frac{3}{4} \right) \\ &= \frac{5}{3}x^3 - \frac{5}{2}x^2 + \frac{5}{6} + \frac{5}{18}x^2 - \frac{5}{9}x + \frac{5}{18} \\ &= \frac{5}{3}x^3 - \frac{20}{9}x^2 - \frac{5}{9}x + \frac{10}{9}. \end{aligned}$$

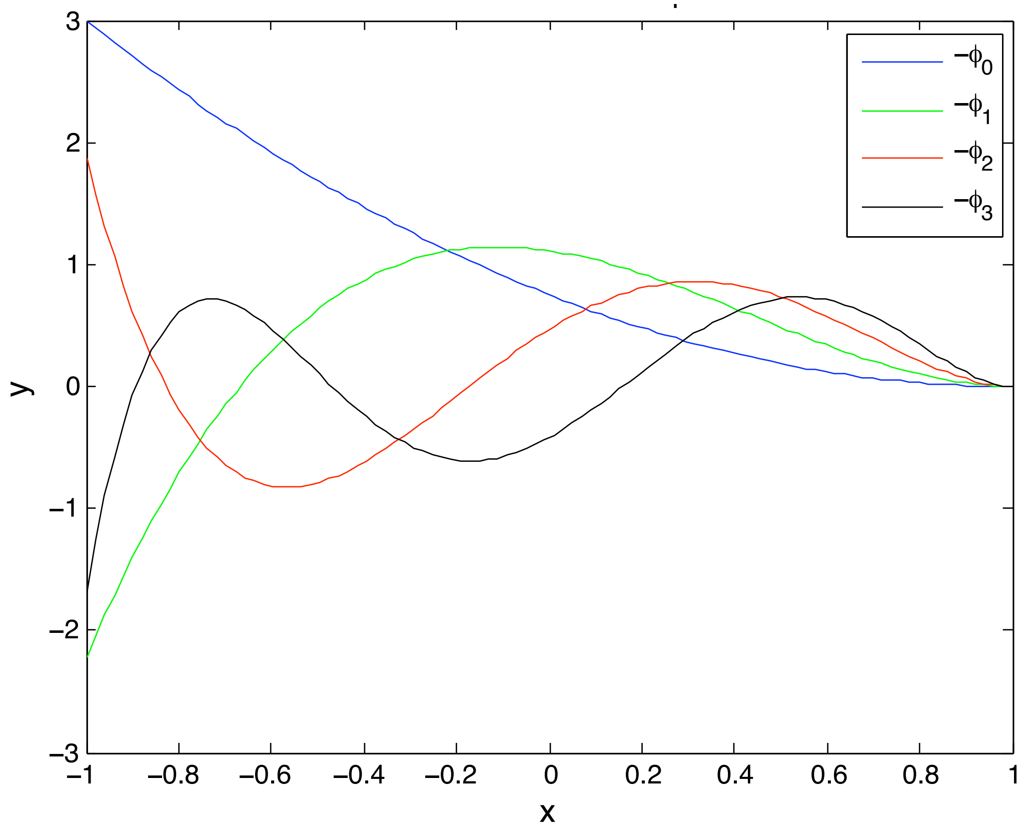


Figure 4.3: Graphs of $\bar{\phi}_j$ for $j = 0, 1, 2, 3$.

The graphs of the first several members of the sequence $\{\bar{\phi}_j\}$ are shown in Figure 4.3. Instead of computing the values of α_j, β_j, and γ_j directly, we use MATLAB to compute them.

4.4 Recurrence Relations for Generalized Jacobi Polynomials

In Sections 4.1 and 4.2, we developed families of polynomials that are orthogonal with respect to the weight function $\omega(x) \equiv 1$. The orthogonal bases developed in [26] are defined in such a way as to satisfy specified boundary conditions, such as the ones employed in this dissertation. These orthogonal bases are known as generalized Jacobi polynomials (GJPs) [15, 26]. These GJPs have parameters $\alpha, \beta \leq -1$ that are orthogonal with respect to the weight function $\omega^{\alpha,\beta}(x) \equiv (1-x)^{\alpha}(1+x)^{\beta}$. Originally, the polynomials developed in [26] were used to solve third or higher odd-order equations, but we can adapt them to our method because they both use short linear combinations of Legendre polynomials. We will now examine the changes that occur to the orthonormal polynomials and their three-term recurrence relations as we change the weight function.

Let J_n be the $n \times n$ Jacobi matrix consisting of the recursion coefficients corresponding to a sequence of polynomials $p_j(t)$, $j = 0, 1, \ldots, n-1$ that is orthonormal with respect to the inner product

$$\langle f, g \rangle_\omega = \int_{-1}^{1} \overline{f(t)} g(t) /, d\lambda t,$$

where $d\lambda(t) = \omega(t)\,dt$, and let $\tilde{J}_n$ be the $n \times n$ Jacobi matrix for a sequence of polynomials $\tilde{p}_j(t)$, $j = 0, 1, \ldots, n-1$ that is orthonormal with respect to the inner product

$$\langle f, g \rangle_{\tilde{\omega}} = \int_{-1}^{1} \overline{f(t)} g(t)\, d\tilde{\lambda} t,$$

where the measure $d\tilde{\lambda}(t) = \tilde{\omega}(t)\,dt$ is a modification of $d\lambda(t)$ by some factor. The following procedures can be used to generate $\tilde{J}_n$ from J_n:

- Multiplying by a linear factor: In the case $d\tilde{\lambda}(t) = (t-c)d\lambda(t)$, we have

  $$\tilde{J}_n = L^T L + cI + \left(\frac{\delta_{n-1}}{l_{nn}} \right)^2 \mathbf{e}_n \mathbf{e}_n^T,$$

  where L is a lower triangular matrix and $J_n - cI = LL^T$ is the Cholesky factorization [9, 10].

- Dividing by a linear factor: In the case $d\tilde{\lambda}(t) = (t-c)^{-1} d\lambda(t)$, where c is near or on the boundary of the interval of integration, the inverse Cholesky (IC) procedure [7] can be used to obtain $\tilde{J}_n$. We have the following equation

  $$\tilde{J}_n = L^{-1} J_n L - cI + \left(\frac{\delta_{n-1}}{l_{nn}} \right) \mathbf{e}_n \mathbf{c}^T,$$

  where $I = (J_n - cI) LL^T + \mathbf{e}_n \mathbf{d}^T$ and $\mathbf{c}$ and $\mathbf{d}$ are vectors that do not have to be computed if one is content with only computing $\tilde{J}_{n-1}$.

In both cases, the modified and original polynomials are related by L:

$$\mathbf{p}(t) = L\tilde{\mathbf{p}}(t),$$

where $\mathbf{p}(t) = [p_0(t) \cdots p_{n-1}(t)]^T$ and $\tilde{\mathbf{p}}(t) = [\tilde{p}_0(t) \cdots \tilde{p}_{n-1}(t)]^T$.

Three-term recurrence relations for the Jacobi polynomials are well-known, but we are not aware of similar recurrence relations for GJPs. We now present efficient algorithms for modifying the family of polynomials $\{\bar{\phi}_j\}$ in Section 4.3 to obtain such recurrences.

4.4.1 GJPs for the Boundary Condition $p(1) = p'(1) = 0$

The polynomials $\{\bar{\phi}_j\}$ can be modified to obtain the three-term recurrence relation for the GJPs

$$\varphi_j(x) = (1-x)^2 J_j^{2,0}(x) = \frac{(-1)^j}{2^j (j)!} \frac{d^j}{dx^j} \left\{ (1-x)^{j+2} (1+x)^j \right\}, \ j = 0, 1, \ldots, \tag{4.54}$$

which are orthogonal on $(-1,1)$ with respect to the weight function $(1-x)^{-2}$ [15]. These polynomials satisfy the boundary condition $\varphi(1) = \varphi'(1) = 0$.

Let

$$J_n = \begin{bmatrix} \alpha_0 & \gamma_0 & & & \\ \beta_0 & \alpha_1 & \gamma_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-3} & \alpha_{n-2} & \gamma_{n-2} \\ & & & \beta_{n-2} & \alpha_{n-1} \end{bmatrix}$$

be the matrix of recursion coefficients for the family of polynomials $\{\bar{\phi}_j\}_{j=0}^{n-1}$ where

$$\alpha_j = \frac{\langle \bar{\phi}_j, x\bar{\phi}_j \rangle}{\langle \bar{\phi}_j, \bar{\phi}_j \rangle},$$

$$\beta_j = \frac{\langle \bar{\phi}_{j+1}, x\bar{\phi}_j \rangle}{\langle \bar{\phi}_{j+1}, \bar{\phi}_{j+1} \rangle},$$

$$\gamma_j = \frac{\langle \bar{\phi}_{j+1}, x\bar{\phi}_j \rangle}{\langle \bar{\phi}_j, \bar{\phi}_j \rangle}.$$

To symmetrize J_n, we must apply a diagonal similarity transformation which yields the following

$$\tilde{J}_n = \begin{bmatrix} \alpha_0 & \delta_0 & & & \\ \delta_0 & \alpha_1 & \delta_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{n-3} & \alpha_{n-2} & \delta_{n-2} \\ & & & \delta_{n-2} & \alpha_{n-1} \end{bmatrix}$$

where $\delta_j = \sqrt{\gamma_j \beta_j}$ for $j = 0, 1, \ldots, n-2$.

Let $\hat{J}_n$ be the Jacobi matrix for the polynomials $\varphi_j(x)$. We can apply the inverse Cholesky algorithm to compute $\hat{J}_{n-1}$ directly from $\tilde{J}_n$ since its measure is a modification of that of J_n and $\tilde{J}_n$ by dividing by a linear factor. Unfortunately, this computationally expensive.

To get around this problem, we let $\bar{J}_n$ be the Jacobi matrix for polynomials $\varphi_j(x)$ that are orthonormal with respect to the weight function $\omega \equiv (1-x)^{-2}$. We would like to obtain $\bar{J}_{n-1}$ from $\tilde{J}_n$ and then obtain $\hat{J}_{n-2}$ from $\bar{J}_{n-1}$.

First, we let $T_n = I - \tilde{J}_n$ with the modification $d\tilde{\lambda}(t) = (1-t)^{-1}d\lambda(t)$. Then, we can find the (n,n) entry of the matrix equation

$$T_n = L^T L + \left(\frac{\bar{\delta}_{n-1}}{l_{nn}}\right)^2 \mathbf{e}_n \mathbf{e}_n^T \tag{4.55}$$

for l_{nn}^2, where $\bar{\delta}_{n-1} = \langle x\bar{\varphi}_{n-2}, \bar{\varphi}_{n-1}\rangle_{\bar{\omega}}$. The entry $\bar{\delta}_{n-1}$ of $\bar{J}_n$ is unknown, so we will leave it that way for now. Next, we will compute the factorization

$$L^T L = T_n - \left(\frac{\bar{\delta}_{n-1}}{l_{nn}}\right)^2 \mathbf{e}_n \mathbf{e}_n^T.$$

Therefore,

$$\hat{J}_n = I - LL^T. \tag{4.56}$$

The correct $\bar{J}_n$ and the matrix obtained in (4.56) differ by the (n,n) entry. So, deleting the last row and column yields the correct $\bar{J}_{n-1}$. We will repeat this process for the modification of the weight function by dividing by another factor $(1-x)$.

Let $\bar{T}_{n-1} = I - \bar{J}_{n-1}$. Now, we can solve the $(n-1,n-1)$ entry of the matrix

$$\bar{T}_{n-1} = \bar{L}^T \bar{L} + \left(\frac{\hat{\bar{\delta}}_{n-2}}{l_{n-1,n-1}}\right)^2 \mathbf{e}_{n-1} \mathbf{e}_{n-1}^T$$

for $l_{n-1,n-1}^2$ where $\hat{\bar{\delta}}_{n-2}$ can be computed using (4.54). Next, we will compute the factorization

$$\bar{L}^T \bar{L} = \bar{T}_{n-1} - \left(\frac{\hat{\bar{\delta}}_{n-2}}{l_{n-1,n-1}}\right)^2 \mathbf{e}_{n-1} \mathbf{e}_{n-1}^T.$$

As a result, we have

$$\hat{J}_{n-1} = I - \bar{L}\bar{L}^T.$$

Lastly, we need to delete the last row and column of $\hat{J}_{n-1}$ to obtain $\hat{J}_{n-2}$.

To find the value of the unknown $\bar{\delta}_{n-1}$, we will note that the correct value of the $(n-2,n-2)$ entry is now known and its value can be found by using (4.54). However, it can be determined using the properties of even and odd functions that its value must be zero. Thus, we solve the equation

$$F(\bar{\delta}_{n-1}) = 0,$$

where $F(\delta)$ is the $(n-2,n-2)$ entry of $\hat{J}_{n-2}$ obtained from $\tilde{J}_n$ using the above procedure, with $\bar{\delta}_{n-1} = \delta$.

To solve this equation, we use the secant method. Applying the quadratic formula in solving (4.55), we have it can be determined that the solution must lie in $(0, \frac{1}{2}]$. Choosing the initial guesses close to the upper bound of $\frac{1}{2}$ results in rapid convergence.

Chapter 5

NUMERICAL RESULTS

5.1 Computing Functions of A

First, we will look at the time-independent case with various values of m. In the time-independent case, we will show that the KSS method outperforms the Lanczos method.

5.1.1 Solving $A\mathbf{u} = \mathbf{f}$

Table 5.1: Time-independent estimates of relative error for $m = 0$

N	KSS	Lanczos	Lanczos Iterations
20	3.4793e-06	8.4745e-06	16
80	8.7820e-06	7.1359e-06	35
320	1.3737e-05	1.5611e-05	69

Table 5.2: Time-independent estimates of relative error for $m = 1$

N	KSS	Lanczos	Lanczos Iterations
20	4.1118e-02	7.6304e-02	14
80	5.3879e-02	5.5219e-02	41
320	5.9915e-02	3.5040e-01	73

Table 5.3: Time-independent estimates of relative error for $m = 5$

N	KSS	Lanczos	Lanczos Iterations
20	1.0386e-02	1.2069e-02	6
80	1.4657e-02	1.8802e-02	24
320	7.3949e-03	6.1633e-01	73

Table 5.4: Time-independent estimates of relative error for $m = 10$

N	KSS	Lanczos	Lanczos Iterations
20	4.4631e-03	2.3181e-03	6
80	3.4187e-03	3.4365e-03	18
320	1.2808e-02	1.2969e-02	55

Tables 5.1 – 5.4 contain the time-independent results using low-frequency components. The asymptotic analysis in Chapter 3 is only for the high-frequency case, therefore its results are best not used for computing low-frequency components [21]. Notice that the number of iterations for Lanczos increases substantially as N increase, while the same accuracy was obtained with three iterations for KSS. The Lanczos method results for $N = 320$ in Tables 5.2 and 5.3 do not have the same order of accuracy. When the number of Lanczos iterations is larger than 73, the accuracy deteriorates.

5.1.2 Solving $\mathbf{x}' = A\mathbf{x}$

Table 5.5: Estimates of relative error for $m = 0$, $N = 20$

Δt	KSS	Lanczos (2)	Lanczos (4)
1	2.3648e-04	7.54572e-02	2.5399e-03
$1/2$	1.4928e-05	2.0904e-02	1.5748e-04
$1/4$	1.1787e-06	4.9484e-03	2.8968e-06
$1/8$	1.1736e-07	2.1593e-03	2.4413e-07

Table 5.6: Estimates of relative error for $m = 0$, $N = 80$

Δt	KSS	Lanczos (2)	Lanczos (4)
1	2.3648e-04	7.5457e-02	2.5399e-03
$1/2$	1.4928e-05	2.0904e-02	1.5628e-04
$1/4$	1.1787e-06	4.9484e-03	5.0867e-04
$1/8$	1.1736e-07	5.8363e-03	4.5024e-04

Table 5.7: Estimates of relative error for $m = 1$ and $N = 20$

Δt	KSS	Lanczos (2)	Lanczos (4)
$1/100$	5.0368e-06	8.4921e-02	2.1033e-04
$1/200$	6.0623e-0	4.4214e-0	2.7240e-05
$1/400$	7.4320e-08	2.2574e-02	3.4646e-06
$1/800$	9.2036e-09	1.1408e-02	4.3682e-07

Table 5.8: Estimates of relative error for $m = 1$ and $N = 80$

Δt	KSS	Lanczos (2)	Lanczos (4)
$1/100$	1.0321e-05	7.7712e-01	1.9314e-03
$1/200$	1.2463e-06	7.0728e-01	8.9913e-05
$1/400$	1.5303e-07	2.2345e-02	3.4043e-06
$1/800$	1.8950e-08	1.1287e-02	4.2917e-07

Table 5.9: Estimates of relative error for $m = 5$ and $N = 20$

Δt	KSS	Lanczos (2)	Lanczos (4)
$1/100$	2.4342e-01	3.7214e-01	4.7104e-03
$1/200$	3.3808e-02	2.1411e-01	6.7567e-04
$1/400$	4.4119e-03	1.1565e-01	9.0480e-05
$1/800$	5.6110e-04	6.0223e-02	1.1707e-05

Table 5.10: Estimates of relative error for $m = 5$ and $N = 80$

Δt	KSS	Lanczos (2)	Lanczos (4)
$1/100$	6.1828e+00	9.9741e-01	4.7932e-02
$1/200$	6.4629e+00	9.6505e-01	1.9885e-03
$1/400$	4.0200e+00	1.8508e-01	2.0236e-04
$1/800$	6.6176e-01	9.9458e-02	2.6323e-05

Table 5.11: Estimates of relative error for $m = 10$ and $N = 20$

Δt	KSS	Lanczos (2)	Lanczos (4)
$1/100$	7.1758e-04	4.6867e-01	1.4996e-02
$1/200$	1.4674e-05	1.9315e-01	2.4914e-03
$1/400$	1.0342e-05	1.0047e-01	3.6240e-04
$1/800$	1.9787e-06	5.1782e-02	4.8868e-05

Table 5.12: Estimates of relative error for $m = 10$ and $N = 80$

Δt	KSS	Lanczos (2)	Lanczos (4)
$1/100$	5.4918e+00	4.6889e-01	1.5006e-02
$1/200$	9.0088e+00	1.9316e-01	2.4917e-03
$1/400$	7.0997e+00	1.0047e-01	3.6245e-04
$1/800$	7.2146e+00	5.1784e-02	4.8875e-05

Tables 5.5 –5.12 contain the relative error estimates for the time-dependent problem $\mathbf{x}' = A\mathbf{x}$. By examining these tables, we can conclude that in the $m = 0$ case, the KSS method outperforms the Lanczos method. As we increase N, the accuracy of the Lanczos method deteriorates. In the $m \neq 0$ case, the KSS method doesn't perform well until the time step is made small enough, which is when the Lanczos method performs better. However, when the time step is larger, Lanczos method needs more iterations to obtain high accuracy. Usually KSS' strength is at larger time step, with larger matrices, but that is not the case here.

5.2 Solving $C\mathbf{x}' = -A\mathbf{x}$ using Crank-Nicolson and Backward Euler

Table 5.13: Estimates of error for backward Euler with $m = 0$, $K = 3$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.0093e00	1.0093e00	1.0093e00
0.1	1.3746e-01	1.3746e-01	1.3746e-01
0.01	4.3125e-02	4.3125e-02	4.3126e-02
0.001	2.3217e-01	2.3217e-01	2.3217e-01

Table 5.14: Estimates of error for Crank-Nicolson with $m = 0$, $K = 3$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	6.1769e-01	6.1779e-01	6.1797e-01
0.1	5.8057e-02	5.7341e-02	5.6739e-02
0.01	8.0611e-02	8.0741e-02	2.1963e-02
0.001	2.3957e-01	2.3957e-01	2.3957e-01

Table 5.15: Estimates of error for backward Euler with $m = 1$, $K = 3$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	9.0328e00	9.0335e00	9.0344e00
0.1	7.6784e-01	8.1193e-01	8.2591e-01
0.01	1.2236e-01	1.2236e-01	8.2010e-02
0.001	4.5779e-01	8.4076e-01	8.4076e-01

Table 5.16: Estimates of error for Crank-Nicolson with $m = 1$, $K = 3$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.6147e+01	1.6204e+01	1.6195e+01
0.1	8.2899e-01	1.0418e+00	2.8855e+00
0.01	1.5552e-01	2.6510e+02	5.8555e+11
0.001	8.0579e-01	6.3783e+54	1.3248e+77

Table 5.17: Estimates of error for Crank-Nicolson with $m = 3$, $K = 3$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	2.9896e+04	2.9896e+04	2.9896e+04
0.1	1.7425e+03	1.6777e+03	1.5795e+03
0.01	2.3868e+00	1.8809e+04	9.7379e+13
0.001	1.7425e+03	1.8947e+49	2.3224e+75

Table 5.18: Estimates of error for backward Euler with $m = 3$, $K = 3$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	3.1308e+03	3.1308e+03	3.1308e+03
0.1	2.5692e+01	2.5696e+01	2.5700e+01
0.01	5.9403e-01	5.9403e-01	6.1510e-01
0.001	5.4720e-01	5.0422e-01	5.0422e-01

Table 5.19: Estimates of error for backward Euler with $m = 10$, $K = 3$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.1665e+21	1.1665e+21	1.1665e+21
0.1	6.9410e+14	6.9410e+14	6.9410e+14
0.01	2.2701e+04	2.2701e+04	2.2701e+04
0.001	5.6092e-01	5.6090e-01	5.6090e-01

Table 5.20: Estimates of error for Crank-Nicolson with $m = 10$, $K = 3$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	9.3915e+22	9.3916e+22	9.4055e+22
0.1	1.7062e+22	1.7079e+22	1.6589e+22
0.01	1.0792e+21	1.1534e+21	5.9127e+28
0.001	9.6019e-01	1.9342e+31	1.4839e+72

Table 5.21: Estimates of error for Crank-Nicolson with $m = 0$, $K = 3$, a random function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	5.5390e+00	5.5390e+00	5.7028e+00
0.1	4.3148e+00	5.3553e+00	5.6738e+00
0.01	6.4461e-02	3.6326e+00	5.3175e+00
0.001	1.9786e-01	1.0508e-01	3.5396e+00

Table 5.22: Estimates of error for backward Euler with $m = 0$, $K = 3$, a random function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	9.6381e-01	9.6373e-01	9.6370e-01
0.1	1.3684e-01	1.3721e-01	1.3743e-01
0.01	3.1591e-02	3.1442e-02	3.2238e-02
0.001	1.9261e-01	1.9194e-01	1.9196e-01

Table 5.23: Estimates of error for backward Euler with $m = 0$, $K = 1$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	8.7086e-01	8.6922e-01	8.6911e-01
0.1	8.2056e-02	8.3597e-02	8.3699e-02
0.01	1.3479e-01	1.3560e-01	1.3565e-01
0.001	1.4754e-01	1.4834e-01	1.4839e-01

Table 5.24: Estimates of error for backward Euler with $m = 0$, $K = 2$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.0021e+00	1.0012e+00	1.0011e+00
0.1	1.2531e-01	1.2364e-01	1.2351e-01
0.01	2.3661e-02	2.9146e-02	2.9549e-02
0.001	4.2474e-02	4.9127e-02	4.9504e-02

Table 5.25: Estimates of error for backward Euler with $m = 0$, $K = 3$, a random smooth Function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.0085e+00	1.0080e+00	1.0075e+00
0.1	1.3735e-01	1.3716e-01	1.3472e-01
0.01	9.7796e-03	1.3875e-02	1.4624e-02
0.001	2.3721e-02	2.8241e-02	4.9504e-02

Table 5.26: Estimates of error for Crank-Nicolson with $m = 0$, $K = 1$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	5.6483e-01	5.6791e-01	5.6905e-01
0.1	9.6348e-02	1.0230e-01	1.3707e-01
0.01	2.3734e+01	8.3667e+00	1.0823e-01
0.001	1.0619e-01	5.5858e+06	8.5365e+01

When using columns of the eigenvectors of the matrix C, KSS has better accuracy than using standard basis vectors. The numerical results of KSS using Backward Euler and Crank Nicolson are computed using the more efficient KSS method described in Chapter 2. Crank-Nicolson is second-order accurate in time whereas backward Euler is first-order accurate. However, the backward Euler method outperforms Crank-Nicolson method in most cases.

Table 5.27: Estimates of error for Crank-Nicolson with $m = 0$, $K = 2$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	5.8355e-01	5.6066e-01	5.4674e-01
0.1	4.7992e-02	2.4340e-02	1.1200e-01
0.01	2.4036e-02	3.0335e-01	8.2610e-02
0.001	2.1323e-02	2.1062e+09	4.0920e+05

Table 5.28: Estimates of error for Crank-Nicolson with $m = 0$, $K = 3$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	6.0498e-01	5.9384e-01	5.8613e-01
0.1	5.7783e-02	4.3128e-02	3.3161e-02
0.01	9.7044e-03	8.8685e-02	1.4038e-02
0.001	1.2862e-02	5.2261e-01	4.3232e+08

Table 5.29: Estimates of error for Crank-Nicolson with $m = 1$, $K = 3$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.6161e+01	1.6187e+01	1.6198e+01
0.1	9.0309e-01	6.3421e+00	1.4311e+01
0.01	3.7047e-01	3.2005e+11	4.5626e+22
0.001	1.8229e-02	1.8275e+140	2.6700e+234

Table 5.30: Estimates of error for Crank-Nicolson with $m = 3$, $K = 3$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	2.9895e+04	2.9947e+04	2.9986e+04
0.1	1.8907e+03	9.3229e+02	5.4765e+03
0.01	7.8604e+02	4.9363e+09	3.2462e+21
0.001	1.2508e-02	7.4653e+104	2.9118e+184

Table 5.31: Estimates of error for Crank-Nicolson with $m = 10$, $K = 3$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	9.3926e+22	9.3897e+22	9.3836e+22
0.1	1.7039e+22	1.7175e+22	1.5170e+22
0.01	1.1237e+21	1.2335e+22	3.7232e+36
0.001	9.8645e+08	1.4972e+88	6.7650e+182

Table 5.32: Estimates of error for backward Euler with $m = 3$, $K = 3$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	3.1309e+03	3.1312e+03	3.1311e+03
0.1	2.5693e+01	2.5695e+01	2.5715e+01
0.01	6.5196e-01	6.5260e-01	6.5554e-01
0.001	3.1162e-02	2.9537e-02	2.9812e-02

Table 5.33: Estimates of error for backward Euler with $m = 10$, $K = 3$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.1674e+21	1.1665e+21	1.1668e+21
0.1	7.3319e+14	7.3470e+14	7.3493e+14
0.01	2.5849e+04	2.5869e+04	2.5889e+04
0.001	2.2632e+00	2.2613e+00	2.2607e+00

Table 5.34: Estimates of error for backward Euler with $m = 0$, $K = 3$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	9.5496e-01	9.4627e-01	9.4271e-01
0.1	1.3667e-01	1.3665e-01	1.3434e-01
0.01	1.0050e-02	1.4154e-02	1.4898e-02
0.001	2.1426e-02	2.5662e-02	2.6592e-02

Table 5.35: Estimates of error for Crank-Nicolson with $m = 0$, $K = 3$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	5.5132e+00	5.4890e+00	5.6309e+00
0.1	4.2834e+00	5.2543e+00	5.6456e+00
0.01	2.1072e-02	5.2203e+00	8.6097e+00
0.001	1.1697e-02	5.8422e-01	4.1773e+10

Table 5.36: Estimates of error for Crank-Nicolson with $m = 1$, $K = 3$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	5.2657e+01	6.7296e+02	6.4673e+01
0.1	6.9516e+01	6.7296e+02	3.7841e+03
0.01	3.0660e+02	1.3391e+13	7.7340e+24
0.001	1.7401e-02	8.1090e+141	3.6040e+231

Table 5.37: Estimates of error for backward Euler with $m = 1$, $K = 3$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	8.6574e+00	8.6288e+00	8.6175e+00
0.1	8.0239e-01	7.8123e-01	7.7512e-01
0.01	4.4807e-02	5.7012e-02	6.0081e-02
0.001	3.3665e-02	4.4720e-02	4.5097e-02

Table 5.38: Estimates of error for backward Euler with $m = 10$, $K = 3$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.2874e+21	1.2905e+21	2.4813e+01
0.1	7.3152e+14	7.3294e+14	2.2274e+00
0.01	2.5693e+04	2.5712e+04	2.5664e+04
0.001	2.2414e+00	2.2393e+00	2.2386e+00

5.3 Solving $C\mathbf{x}' = -A\mathbf{x}$ using KSS to Compute $e^{-C^{-1}A\Delta t}$

Table 5.39: Estimates of error for KSS with $m = 0$, $K = 1$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	8.0094e-01	9.6729e-01	9.9335e-01
0.1	6.7056e-01	7.6480e-01	8.9475e-01
0.01	7.4459e-01	9.4423e-01	9.7887e-01
0.001	9.9581e-01	9.6422e-01	7.8922e-01

Table 5.40: Estimates of error for KSS with $m = 0$, $K = 2$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	5.5029e-01	8.9257e-01	9.8125e-01
0.1	3.4057e-01	8.4260e-01	8.3932e-01
0.01	2.1990e-01	8.8723e-01	9.2830e-01
0.001	9.8558e-05	1.1775e+07	9.2830e-01

Table 5.41: Estimates of error for KSS with $m = 0$, $K = 3$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	3.7449e-01	8.1620e-01	9.6735e-01
0.1	6.9521e-02	6.8155e-01	8.2356e-01
0.01	1.1356e-04	8.7983e-01	9.2387e-01
0.001	2.7977e-06	9.7907e-01	9.0946e-01

Table 5.42: Estimates of error for KSS with $m = 0$, $K = 1$, a random function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	9.0341e-01	9.7462e-01	9.9365e-01
0.1	5.4207e-01	7.3791e-01	8.4850e-01
0.01	7.3015e-01	9.2873e-01	9.2481e-01
0.001	9.9503e-01	9.9153e-01	8.5607e-01

Table 5.43: Estimates of error for KSS with $m = 0$, $K = 2$, a random function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	7.2510e-01	9.3672e-01	9.8381e-01
0.1	3.8066e-01	7.6330e-01	8.3619e-01
0.01	2.8135e-01	9.6465e-01	9.3759e-01
0.001	1.6508e-03	4.1769e+04	8.7475e-01

Table 5.44: Estimates of error for KSS with $m = 0$, $K = 3$, a random function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	5.0875e-01	8.8661e-01	9.7315e-01
0.1	1.3269e-01	5.4087e-01	7.3300e-01
0.01	2.8581e-03	8.7015e-01	9.4461e-01
0.001	2.8366e-05	9.7785e-01	9.4277e-01

Table 5.45: Estimates of error for KSS with $m = 1$, $K = 3$, a random smooth function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	3.6532e-01	5.5999e-01	5.7317e-01
0.1	8.5295e-02	5.6086e-01	5.3209e-01
0.01	2.7475e-03	5.5642e-01	5.0815e-01
0.001	5.6796e-06	5.0796e-01	4.1729e-01

Table 5.46: Estimates of error for KSS with $m = 1$, $K = 3$, a random function, and columns of identity

Δt	$N = 20$	$N = 80$	$N = 320$
1	3.4354e-01	5.9038e-01	6.1003e-01
0.1	1.2177e-01	5.6461e-01	5.4232e-01
0.01	1.1695e-03	5.7615e-01	6.2295e-01
0.001	2.4366e-04	5.4236e-01	3.1266e-01

Table 5.47: Estimates of error for KSS with $m = 0$, $K = 2$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.2708e-01	1.7764e-01	1.9005e-01
0.1	1.0366e-02	1.0895e-02	.1.3415e-02
0.01	7.1146e-04	1.5594e-03	1.6763e-04
0.001	5.5240e-06	9.5356e-05	1.7421e-04

Table 5.48: Estimates of error for KSS with $m = 0$, $K = 3$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.0157e-01	1.7054e-01	1.8769e-01
0.1	6.7930e-03	1.0603e-02	1.3062e-02
0.01	1.8237e-04	1.2086e-03	1.3553e-03
0.001	7.4155e-07	6.4302e-05	1.5688e-04

Table 5.49: Estimates of error for KSS with $m = 1$, $K = 1$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	4.1550e-01	4.2816e-01	4.3355e-01
0.1	1.1887e-01	3.6967e-01	2.5724e-01
0.01	3.1797e-01	4.1613e-01	3.9872e-01
0.001	4.5431e-04	4.1630e-01	3.9885e-01

Table 5.50: Estimates of error for KSS with $m = 3$, $K = 3$,a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	5.1529e-01	5.3809e-01	5.3809e-01
0.1	5.9767e-02	2.1330e-01	5.3809e-01
0.01	2.7133e-03	4.0673e-01	5.3809e-01
0.001	4.4056e-06	4.3085e-01	4.5951e-01

Table 5.51: Estimates of error for KSS with $m = 5$, $K = 3$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	9.5834e-01	9.9598e-01	9.9725e-01
0.1	4.1838e-01	9.6772e-01	9.1036e-01
0.01	1.5331e-02	9.2582e-01	9.7937e-01
0.001	9.1578e-05	9.3784e-01	9.8524e-01

Table 5.52: Estimates of error for KSS with $m = 10$, $K = 3$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	8.3029e-01	9.9842e-01	1.0000e+00
0.1	9.9994e-01	9.9997e-01	1.0000e+00
0.01	1.0000e+00	1.0000e+00	1.0000e+00
0.001	1.3558e-03	1.0000 e+00	1.0000e+00

Table 5.53: Estimates of error for KSS with $m = 0$, $K = 1$, a random smooth function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.6435e-01	1.8573e-01	1.9340e-01
0.1	8.6320e-03	4.6330e-02	2.4866e-02
0.01	7.7870e-04	1.4718e-03	1.9334e-03
0.001	4.5418e-05	9.5481e-05	1.9626e-04

Table 5.54: Estimates of error for KSS with $m = 0$, $K = 1$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.8702e-01	1.8703e-01	1.8704e-01
0.1	6.9101e-03	3.7369e-02	2.2973e-02
0.01	2.2940e-03	1.0642e-03	6.9054e-03
0.001	2.1787e-04	3.5958e-04	3.5913e-04

Table 5.55: Estimates of error for KSS with $m = 0$, $K = 2$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.8645e-01	1.8703e-01	1.8704e-01
0.1	2.1058e-02	1.5243e-02	1.6840e-02
0.01	4.7218e-03	2.7190e-03	4.4225e-03
0.001	3.7749e-04	3.1036e-04	4.1755e-04

Table 5.56: Estimates of error for KSS with $m = 0$, $K = 3$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.8174e-01	1.8700e-01	1.8703e-01
0.1	2.6604e-02	1.1366e-02	1.7066e-02
0.01	4.4037e-03	2.7788e-03	1.8152e-03
0.001	5.1735e-05	2.2410e-04	4.3725e-04

Table 5.57: Estimates of error for KSS with $m = 1$, $K = 1$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	4.2342e-01	4.2364e-01	4.2364e-01
0.1	1.1114e-01	1.2295e-01	1.2186e-01
0.01	3.1406e-01	3.9561e-01	3.7494e-01
0.001	2.2495e-04	4.0155e-01	3.8292e-01

Table 5.58: Estimates of error for KSS with $m = 3$, $K = 3$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	5.1643e-01	5.4245e-01	5.9780 e-01
0.1	1.2139e-01	2.5955e-01	3.2585e-01
0.01	1.3007e-02	3.9591e-01	5.4344e-01
0.001	1.8660e-04	4.4504e-01	4.7220e-01

To compute the results in Tables 5.39 –5.60, we used the standard KSS method using symmetric block Lanczos for each component. This method approximates the exponential instead of discretizing in time.

When $m = 0$ and we use the standard basis vectors, we have decent accuracy using the KSS method for $N = 20$. In this case, the accuracy tends to increase as the time step gets smaller. As we increase the value of N to 80, the accuracy is first order except when the time

Table 5.59: Estimates of error for KSS with $m = 5$, $K = 3$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	9.9631e-01	9.9782e-01	9.9786e-01
0.1	7.1073e-01	9.6761e-01	9.2985e-01
0.01	5.9001e-02	9.2591e-01	9.8029e-01
0.001	7.0122e-04	9.3509e-01	9.8555e-01

Table 5.60: Estimates of error for KSS with $m = 10$, $K = 3$, a random function, and columns of eigenvectors of C

Δt	$N = 20$	$N = 80$	$N = 320$
1	1.0000e+00	1.0000e+00	1.0000e+00
0.1	9.9996e-01	9.9999e-01	1.0000e+00
0.01	1.0000e+00	1.0000e+00	1.0000e+00
0.001	9.7140e-03	1.0000e+00	1.0000e+00

step is 0.001. When $N = 320$, we have first order accuracy regardless of the time stepping size. Increasing the quadrature nodes, K, seems to have minimal effect on the accuracy when using standard basis vectors. KSS performs better when a smooth random function is used rather than a random function. However, the order of accuracy for each time step stays the same in both cases.

We will compare the eigenvectors of A to the eigenvectors of C for the $m = 0$ case. The matrix A is highly diagonally dominant, so its eigenvectors are mostly concentrated around a single entry. That is, most components of the eigenvectors are negligibly small. However, this is not the case for C. The measure of the Riemann-Stieltjes integral using $\mathbf{u} = \mathbf{q}_j$ is more concentrated than using $\mathbf{u} = \mathbf{e}_j$ or $A = C + \theta \Delta t M$, so we are able to achieve better accuracy with fewer quadrature nodes. On the other hand, $\mathbf{q}_j$ used as basis vectors are not concentrated (sparse), so it is more difficult to achieve decoupling into frequency-dependent and frequency-independent nodes when using KSS only on A.

We will define the symbols for the spectral decomposition of C as follows:

$$C = \mathbf{Q} \Lambda \mathbf{Q}^T$$

where $\mathbf{Q} = [q_1 \; q_2 \cdots q_N]$ consists of the eigenvectors of C as its columns, and Λ is the diagonal matrix containing the eigenvalues. If $mathbffq_N$ is used, then KSS works quite well because the measures are concentrated. An advantage in computing the eigenvectors is that a smooth solution can be computed using relatively few of them. Another advantage is that the eigenvectors can be computed just once, for a given N, and then re-used in every time step, even if adaptive time-stepping is used.

Multiplying a vector by the eigenvectors of C can be carried out rapidly because the eigenvectors of C comes from applying the symmetric QR algorithm to C. Because C is banded, it only requires $O(n)$ Givens rotations where each rotation takes only $O(1)$ flops on a vector [11]. That is, matrix-vector multiplication by the columns of the matrix C is possible in $O(n)$ operations instead of $O(n^2)$.

Figure 5.1: Graph of $d\alpha(\lambda)$ for $m = 5$ and $K = 2$

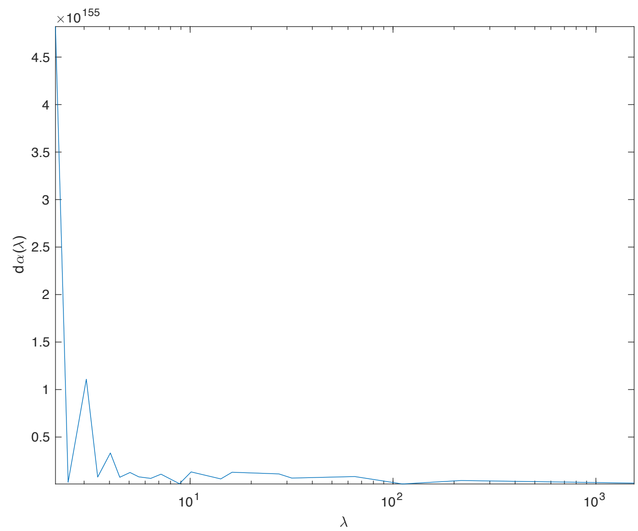
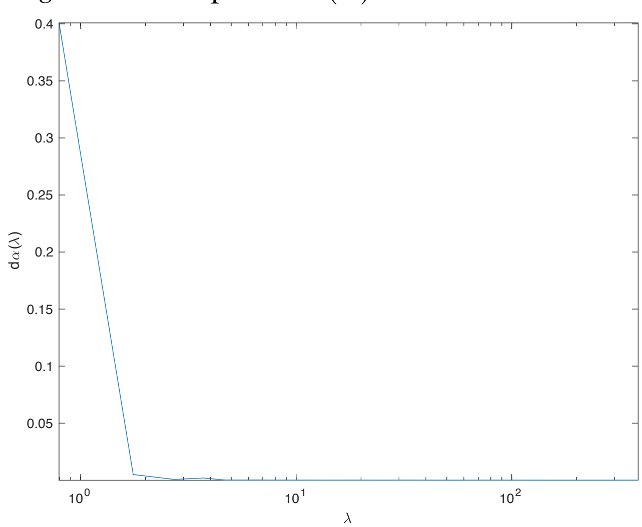


Figure 5.2: Graph of $d\alpha(\lambda)$ for $m = 0$ and $K = 3$



5.4 Results of GJPs for the Boundary Condition $p(1) = p'1 = 0$

Table 5.61 provides numerical results up to $N = 15$ for the boundary conditions $p(1) = p'(1) = 0$. For larger values of N, this method has difficulties due to roundoff error in the

Table 5.61: Estimates of Relative Error for $p(1) = p'(1) = 0$ using GJPs

N	Error	Iterations
5	1.4674 e-13	7
7	1.0266 e-11	6
10	2.5823 e-09	6
12	1.1534 e-06	6
15	4.4990 e-04	6

computation of GJP's using Rodriguez's formula. The number of secant iterations for $N = 5$ to $N = 15$ is 6. When $N = 5$ the number of secant iterations is 7.

Chapter 6

CONCLUSIONS

In conclusion, we have shown that the KSS method can be extended to a circular domain. In the time-independent case, we generalized KSS to the elliptic equation on a disk with analytic expressions for the frequency-dependent nodes. In the time-dependent case, we have taken the first steps in generalizing KSS to the parabolic PDE on a disk, but more work is needed to obtain a more efficient implementation. When $m = 0$, whether time-independent or time-dependent, the scalability observed in KSS for rectangular domains can also be observed when solving PDEs on circular domains. It is possible that the results can be improved by applying diagonal transformations.

We have obtained recurrence relations for generating orthogonal polynomials on the interval $(-1,1)$ that satisfy the boundary conditions (1) $p(1) = 0$, (2) $p(-1) = p(1) = 0$, and (3) $p(1) = p'(1) = 0$. These families of orthogonal polynomials can be used to easily implement transformation matrices between physical and frequency space for function spaces of interest for solving PDEs in polar and cylindrical geometries. While these polynomials are orthogonal with respect to the weight function $\omega(s) \equiv 1$, it has been shown that they can easily be modified to be orthogonal with respect to rational weight functions. When modified as such to obtain GJPs, recursion coefficients can be obtained with far greater efficiency than by computing the required inner products directly. Future work includes the development of numerical methods that make use of these families of orthogonal polynomials, or modifications thereof.

Appendix A

A.1

```
function u=kss_steady1(f,A,tdt,m,N,K,XC)
if nargin<7
    XC=eye(size(A));
end
format shorte
% this should be a function with argument M,m, returns u
plotmeasure(A,XC(:,1),f)
% solve (A + m^2 B + C)u = f
% let M = A + m^2 B + C

% K = number of Lanczos iterations (can be 2 or 3)

% outline:
% frequency-independent nodes:
% Lanczos on M, initial vector f, K iterations
% output [X,T] = [Lanczos vectors, Jacobi matrix]
% nodes = eigenvalues of T

% stdlanczos - performs K Lanczos iterations on M with initial vector f
% b0 = ||f||_2

[X,T]=Lanczos(A,f,K);
% these are the frequency-independent nodes, as a column vector
eg=eig(T(1:K,1:K));

% compute v = ||v||_2 X T^-1 e_1
e1=zeros(K,1);
e1(1)=1;
b0=norm(f);
```

```
v=b0*X(:,1:K)*(T(1:K,1:K)\e1);

% compute w = b1..bK X_K+1
tt=1;
for i=1:K
    tt=tt*T(i,i+1);
end
w=tt*X(:,K+1)*b0;
% row vector of frequency-independent nodes
ni=eg.';

% frequency-dependent nodes:
% use analysis, K iterations (K = 2 or 3)

%nf (# of components x K)
B=zeros(N,K);
beta1=zeros(N,1);
beta2=zeros(N,1);
if m==0
     alpha=2*(1:N)'+2;
    for j=1:N
        beta1(j,1)=sqrt((2*j)/((2*j-1)*(2*j+1))^2+(2*(j+2))/
        ((2*j+3)*(2*j+5))^2);
    end
    for j=1:N
        beta2(j,1)=1/2*sqrt((2*(2*(j-2))/((2*j-5)*(2*j-3)))^2+4*((2*j)/
        ((2*j-1)*(2*j+1)))^2 + 4*((2*(j+2))/((2*j+3)*(2*j+5)))^2+...
        2*((2*(j+4))/((2*j+7)*(2*j+9)))^2);
    end
    if K==2
        for j=1:N
            B(j,1)=sqrt((beta1(j))^2+(beta2(j))^2);
            B(j,2)=-B(j,1);
        end
    end
    if K==3
```

```
        for j=1:N
            B(j,1)=0;
            B(j,2)=sqrt((beta1(j))^2+(beta2(j))^2);
            B(j,3)=-B(j,2);
        end
    end
else
     alpha=4*(1:N)'+6;
    for j=1:N
        beta1(j,1)=sqrt((2*j+2)^2+(2*j+4)^2);
    end
    for j=1:N
        beta2(j,1)=1/2*sqrt((2*(2*j))^2+4*(2*j+2)^2+4*(2*j+4)^2+...
        2*(2*j+6)^2);
    end
    if K==2
        for j=1:N
            B(j,1)=sqrt((beta1(j))^2+(beta2(j))^2);
            B(j,2)=-B(j,1);
        end
    end
    if K==3
        for j=1:N
            B(j,1)=0;
            B(j,2)=sqrt((beta1(j))^2+(beta2(j))^2);
            B(j,3)=-B(j,2);
        end
    end
end
I=eye(N);
nf=zeros(N,K);
for j=1:N
    [~,Tj]=Lanczos(A,XC(:,j),K);
    nj=eig(Tj(1:K,1:K));
    nf(j,:)=reshape(nj,1,K);
end
```

```
% make matrix of all nodes, row = each component, 2K columns
ni=repmat(ni,size(nf,1),1);
ns=[ni nf];

% divdiff_lagrange(nodes) => divided differences
% nested_mult(2nd half of divided differences) =>
% power form coefficients
% how to perform interpolation
% K = number of each type of node
% tfun = integrand (tfun(x)=1./x for inverse)
% ns = matrix of interpolation points, one set per row
% frequency-independent nodes in FIRST columns
tfun=inline('1./x');
F=divdiff_lagrange(ns,tfun);
% select columns of F and ns corresponding to
% frequency-dependent nodes,
% those are the only ones to convert to power form
Q=powerform(F(:,K:-1:1),ns(:,K+1:2*K-1));
z=0;
m1=size(ns,2)/2;
for n=1:K
    Tw=XC'*w;
    z=z+Q(:,m1-n+1).*Tw;
    if n<K
        %w=M*w;
        w=A*w;
    end
end
u1=A\f;
u=v+XC*z;
nu=norm(u1,'inf');
err=norm(u1-u,'inf');


% compute Jacobi matrix from Section 4, up to degree n-2:
n=15;
```

```
[alpha,beta,gamma]=calcalphbetagam(n+2);
bg=sqrt(beta.*gamma);
% Junsym=diag(alpha)+diag(beta(1:n-1),-1)+diag(gamma(1:n-1,1));
% jump directly to symmetrized form
Jsym=diag(alpha)+diag(bg(1:n-1),1)+diag(bg(1:n-1),-1);
T=eye(n)-Jsym;
% solve fcheckmod3(x) = 0, 0 < x < 1/2 for xstar
% secant method
x0=0.5;
f0=fcheckmod3(x0);
x1=0.49;
f1=fcheckmod3(x1);
tol=1e-15;
while true
    x2=x1-f1*(x1-x0)/(f1-f0);
    err=abs(x2-x1)
    if err<tol || err>1e10
        break;
    end
    x0=x1;
    f0=f1;
    x1=x2;
    f1=fcheckmod3(x1);
end
xstar=x2;
% now that we have the correct \bar{\delta}_{n-1},
% compute Jacobi matrix for this value:
[~,J0]=fcheckmod3(xstar);

% check against true Jacobi matrix, constructed by computing
% inner products of polynomials directly, polynomials computed
% using Rodriguez formula implemented by makephit3
% P2 stores coefficients of polynomials, starting from degree 2
P2=zeros(n-1,n);
P2(1,1:3)=[ 1 -2 1 ];
for j=3:n
```

```
    P2(j-1,1:j+1)=makephit3(j);
end
% J3 = exact Jacobi matrix
J3=zeros(n-1);
for i=1:n-1
    for j=1:n-1
        Pi=P2(i,1:i+2);
        Pi=normpoly(Pi,3);
        Pj=P2(j,1:j+2);
        Pj=normpoly(Pj,3);
        PiPj=conv(Pi,Pj);
        tPiPj=conv([ 1 0 ],PiPj);
        % divide by 1-x^2
        [q,~]=deconv(tPiPj,[ 1 -2 1 ]);
        % anti-diff
        Iq=polyint(q);
        % plug in -1,1
        Ipipj=polyval(Iq,1)-polyval(Iq,-1);
        J3(i,j)=Ipipj;
    end
end
```

```
% function f for solving f(x)=0 using secant method
function [y,J0]=fcheckmod3(x)
% make unsymmetric Jacobi matrix J, n x n
n=15;
%Computes the values of alpha, beta and gamma
[alpha,beta,gamma]=calcalphbetagam(n+2);
bg=sqrt(beta.*gamma);
% Junsym=diag(alpha)+diag(beta(1:n-1),-1)+diag(gamma(1:n-1,1));
% jump directly to symmetrized form
Jsym=diag(alpha)+diag(bg(1:n-1),1)+diag(bg(1:n-1),-1);
T=eye(n)-Jsym;
% (n,n) entry of equation: Tnn = Lnn^2 + (x/Lnn)^2
% solve quadratic equation p(r) = 0, r = Lnn^2 where
% p=[ 1 -T(n,n) x^2 ]
% upper bound: 0 < |x| < 1/2 to ensure real roots
```

```
% quadratic formula:
r=(T(n,n) + sqrt(T(n,n)^2 - 4*x^2))/2;
T(n,n)=T(n,n)-x^2/r;
% now compute L from rearranged equation using "reverse Cholesky":
L=reversechol(T);
% final formula for \bar{J}_n:
J1=eye(n)-L*L';
% remove last row and column
J1=J1(1:n-1,1:n-1);
% first mod:
% this is \bar{T}_{n-1} from paper:
T1=eye(n-1)-J1;
% compute \hat{\delta}_{n-2}
pn=makephit3(n);
pn1=makephit3(n+1);
qn=normpoly(pn,3);
qn1=normpoly(pn1,3);
betan=ippolywt(conv([ 1 0 ],qn),qn1,3);
% solve (n,n) entry for r = \bar{L}(n-1,n-1):
r=(T1(n-1,n-1) + sqrt(T1(n-1,n-1)^2-4*betan^2))/2;
% isolate \bar{L}^T \bar{L}:
T1(n-1,n-1)=T1(n-1,n-1)-betan^2/r;
% reverse cholesky to get L1 = \bar{L}
L1=reversechol(T1);
% complete \hat{J}_{n-1}
J0=eye(n-1)-L1*L1';
% J0=L1*L1'-eye(n-1);
% delete last row and column
J0=J0(1:n-2,1:n-2);
% entry we want to match: (n-2,n-2) is our f-value:
y=J0(n-2,n-2);
% compute exact value (happens to be 0), since we're solving f(x)=0:
pnm1=makephit3(n-1);
alphan2=ippolywt(conv([ 1 0 ],pnm1),pnm1,3)/ippolywt(pnm1,pnm1,3);
% we want this to be 0
y=y-alphan2;
```

BIBLIOGRAPHY

[1] Richard C. Aiken. *Stiff Computation*. Oxford University Press, New York, NY, 1985.

[2] K. Atkinson. *An Introduction to Numerical Analysis*. Wiley, 2nd edition edition, 1989.

[3] L. Burden, Richard and J. Douglas Faires. *Numerical Analysis*. Thomson Brooks/Cole, 2005.

[4] J. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, 1987.

[5] A. Cibotarica, J.V. Lambers, and E. Palchak. Solution of nonlinear time-dependent pdes through componentwise approximation of matrix functions. *Journal of Computational Physics*, 321:1120–1143, 2016.

[6] H. Eisen, W. Heinrichs, and K. Witsch. Spectral collocation methods and polar coordinate singularities. *J. Comput. Phys.*, 96:241–257, 1991.

[7] S. Elhay and J. Kautsky. Jacobi matrices for measures modified by a rational factor. *Numerical Algorithms*, 6(2):205–227, 1994.

[8] B. Fornberg. A pseudospectral approach for polar and spherical geometries. *SIAM J. Sci. Comput.*, 16(195):1071–1081, 1994.

[9] W. Gautschi. The interplay between classical analysis and (numerical) linear algebra–a tribute to gene h. golub. *Electr. Trans. Num. Anal.*, 13:119–147, 2002.

[10] G. Golub and J. Kautsky. Calculation of gauss quadratures with multiple free and fixed knots. *Numericshe Mathematik*, 41:147–163, 1983.

[11] Gene H. Golub and Charles Van Loan. *Matrix Computations*. Third edition, 1996.

[12] Gene H. Golub and Gerard Meurant. Matrices, moments, and quadrature. In *Proceedings of the 15th Dundee Conference*. Longman Scientific and Technical, 1994.

[13] G.H. Golub and R. Underwood. The block lanczos method for computing eigenvalues. *J. Rice (Ed.), Mathematical Software III*, pages 361–377, 1977.

[14] D. Gottlieb and S. A. Orszag. *Numerical Analysis of Spectral Methods: Theory and Applications*. SIAM-CBMS, Philadelphia, PA, 1977.

[15] B.-Y. Guo, J. Shen, and L.L. Wang. Generalized jacobi polynomials/functions and their applications. *Applied Numerical Mathematics*, 59:1011–1028, 2009.

[16] M. Hochbruck and C. Lubich. On krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 34:1911–1925, 1996.

[17] M. Hochbruck, C. Lubich, and H. Selhofer. Exponential integrators for large systems of differential equations. *SIAM J. Sci. Comput.*, 19:1552–1574, 1998.

[18] W. Huang and D. M. Sloan. Pole condition for singular problems: the pseudospectral approximation. *J. Comput. Phys.*, 107:254–261, 1993.

[19] W.H. Hundsdorfer. Numerical solution of advection-diffusion-reaction equations. *CWI Report NM-N9603*, 1996.

[20] J.V. Lambers. Enhancement of krylov subspace spectral methods by block lanczos iteration. *Electron. T. Numer. Ana*, 31:86–109, 2008.

[21] E. M. Palchak, A. Cibotarica, and J. V. Lambers. Solution of time-dependent pde through rapid estimation of block gaussian quadrature nodes. *Linear Algebra and its Applications*, 468:233–359, 2015.

[22] M. Richardson and J. V. Lambers. Krylov subspace spectral methods for pdes in polar and cylindrical geometries. *in preparation*.

[23] M. Richardson and J.V. Lambers. Recurrence relations for orthogonal polynomials for pdes in polar and cylindrical geometries. *Springplus*, 1567(5), 2016.

[24] M. Sadkane. A block arnoldi-chebyshev method for computing the leading eigenpairs of large sparse unsymmetric matrices. *Numer. Math.*, 64:181–193, 1993.

[25] Jie Shen. Efficient spectral-galerkin methods iii: Polar and cylindrical geometries. *SIAM J. Sci. Comput.*, 18:1583–1604, 1997.

[26] Jie Shen. A new dual-petrov-galerkin method for third and higher odd-order differential equations: Application to the kdv equation. *SIAM J. Numer. Anal.*, 41(5):1595–1619, 2003.