Summer 6-2-2022

Sparse Spectral-Tau Method for the Two-Dimensional Helmholtz Problem Posed on a Rectangular Domain

Gabriella M. Dalton

Gabriella M. Dalton

Candidate

Mathematics Statistics

Department

This dissertation is approved, and it is acceptable in quality and form for publication: *Approved by the Dissertation Committee:*

Dr. Stephen R. Lau, Associate Professor

Dr. Deborah L. Sulsky, Emeritus Professor

Dr. Jacob B. Schroder, Associate Professor

Sparse Spectral-Tau Method for the Two-Dimensional Helmholtz Problem Posed on a Rectangular Domain

by

Gabriella M. Dalton

B.S., Applied Mathematics, University of New Mexico, 2018
B.A., Spanish, University of New Mexico, 2018

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Mathematics

The University of New Mexico

Albuquerque, New Mexico

July, 2022

Dedication

To my parents, Patrick and Bernadette, for their endless love and support.

To my siblings, Jennifer and June, who are my greatest blessing.

To my Eric, for your unwavering encouragement and love.

"Goodbye blue Monday." – Kurt Vonnegut

Acknowledgments

First and foremost, I would like to thank my advisor Dr. Stephen R. Lau for his invaluable support and kindness throughout the course of my research and graduate studies. No words are enough to express my gratitude towards him for his constant guidance and encouragement.

I also thank Professor Sulsky and Professor Schroder for their time and helpful feedback, as well as for serving on my thesis committee.

Special thanks also goes out to my family, close friends, Edward S. Jimenez, and many others for their guidance, reassurance, friendship, and for keeping me grounded during this journey.

Sparse Spectral-Tau Method for the Two-Dimensional Helmholtz Problem Posed on a Rectangular Domain

by

Gabriella M. Dalton

B.S., Applied Mathematics, University of New Mexico, 2018
B.A., Spanish, University of New Mexico, 2018

M.S., Mathematics, University of New Mexico, 2022

Abstract

Within recent decades, spectral methods have become an important technique in numerical computing for solving partial differential equations. This is due to their superior accuracy when compared to finite difference and finite element methods. For such spectral approximations, the convergence rate is solely dependent on the smoothness of the solution yielding the potential to achieve spectral accuracy. We present an iterative approach for solving the two-dimensional Helmholtz problem posed on a rectangular domain subject to Dirichlet boundary conditions that is well-conditioned, low in memory, and of sub-quadratic complexity. The proposed approach spectrally approximates the partial differential equation by means of modal Chebyshev integration matrices. Implementation of the boundary conditions is achieved through a technique known as "integration preconditioning," although we refer to the technique as integration sparsification. The spectral method presented represents certain partial differential operators in terms of sparse, banded integration matrices. In this work, there are $N+1$ Chebyshev modes associated with each coordinate direction. Therefore, there

are $n = (N+1)^2$ modes in total. For the truncations considered, our method empirically yields a linear set-up cost, followed by a sub-quadratic solve complexity of $\mathcal{O}(n^{1.6})$.

Contents

Chapter 1

Introduction

Partial differential equations (PDEs) model natural occurring phenomena such as heat, electrodynamics, sound, and quantum mechanics. This introduction follows [2] closely. Although these mathematical equations are an essential component to studying many real world applications in scientific fields such as, physics and engineering, closed form analytic solutions are often elusive. As a result, numerical methods are instead used to produce numerical solutions of PDEs. Methods such as the finite element method (FEM), finite volume method (FVM), finite difference method (FDM), and boundary integral method (BIM) are a few numerical methods designed for PDEs. Any choice of numerical method should be stable, consistent, and convergent.

As mentioned, there are different types of methods for numerically solving real-world problems. Particularly, finite element and finite difference methods are used to locally represent a function through low-order polynomials. Contrary to FEM and FDM, spectral methods globally represent a function by high-order polynomials or Fourier series. In other words, since these methods are global methods, this means that the computation is not solely dependent on a neighborhood of points, but rather on information obtained along the entire domain.

Within recent decades, spectral methods have become widely used for numerically solving PDEs. These methods work best when the solution is smooth across the entire domain. Compared with FEM, FVM, FDM methods, spectral methods yield a higher degree of accuracy and are generally computationally less expensive. The idea assumes that a smooth function can be approximated by a basis-function expansion with exponential convergence. Furthermore, spectral methods are also used in fields of applied mathematics and computational sciences to numerically solve diverse problems such as mesh compression and surface reconstruction [9].

Spectral methods are often categorized as either Galerkin, Tau, or collocation methods. The latter method is often referred to as the pseudo-spectral method. The difference between the aforementioned methods lie in working with (modal) expansion coefficients or (nodal) physical point values. This work considers the application of Chebyshev polynomials in spectral Tau-methods for the purpose of approximating the Helmholtz problem subject to Dirichlet boundary conditions posed on a rectangular domain. The presented work involves the adoption of a modal Chebyshev approach based on "integration preconditioning," a term coined by Coutsias et al. [3]. From the standpoint of this work this term is a misnomer, but their procedure does yield sparse linear systems. Therefore, we refer to this approach as *integration sparsification*, with the understanding that further (genuine) preconditioning is needed. These Chebyshev integration matrices are sparse and banded, and responsible for transforming linear partial differential operators with polynomial coefficients into Kronecker products of banded, integration operators. This work focuses on an iterative approach for solving the two-dimensional Helmholtz problem contrived through integration sparsification. The new aspect of this work is the construction of an additional (genuine) preconditioner on top of the "integration preconditioning."

The main results of this thesis concern the specification and study of a new modal-based preconditioner for the Helmholtz equation approximated by spectral-tau Chebyshev methods. Typically, preconditioning in spectral methods is carried out in the context of nodal

(pseudo-spectral) methods. For example, with nodal methods an approximate inversion often stems from inversion of lower order finite-difference or finite-element representations of a particular spectral representation of an operator. Modal-based preconditioning is not as well studied.

The outline of this work is as follows. Chapter 2 describes an overview of Chebyshev polynomials and the modal approximation of the two-dimensional Helmholtz equation; similar approximation of the Helmholtz problem can be achieved for higher spatial dimensions. Algorithmic details for the solution approach and preconditioning methods can be found in Chapter 3. Section 3.2 describes our modal-based preconditioner. Chapter 4 presents numerical results, in particular demonstrating that an iterative solution approach empirically yields a linear set-up cost followed by sub-quadratic solve complexity for the preconditioned system. Lastly, Chapter 5 summarizes the efforts of the presented work, as well as possible extensions of this work.

Chapter 2

Background

2.1 Overview of Chebyshev Polynomials

Chebyshev polynomials are essential in approximation theory and are widely used in the field of numerical analysis for numerical solutions of ordinary and partial differential equations through spectral or pseudo-spectral methods. Chebyshev polynomials of the first kind $T_n(x)$ are closely related to the cosine function.

2.1.1 Basic Properties of Chebyshev Polynomials

1. $T_n(x)$ has n distinct roots located on the closed interval $x \in [-1, 1]$ such that $T_n(t_k) = 0$ for $t_k = \cos\left(\frac{(2k-1)\pi}{2n}\right)$ for $k = 1, 2, ..., n$.

2. The leading coefficient of $T_n(x)$ is 2^{n-1}.

3. The maximal value of $T_n(x)$ occurs with alternating sign $(n + 1)$ times such that $|T_n(x_k)| = 1$, $T_n(x_k) = (-1)^k$, $\quad x_k = \cos\left(\frac{\pi k}{n}\right)$, $\quad k = 0, 1, 2, ..., n$.

Let us first consider the Chebyshev polynomial of the first kind of order n defined by,

$$T_n(x) = \cos(n\cos^{-1}(x)), \tag{2.1}$$

where x is on the interval $[-1, 1]$ for $n = 0, 1, 2, ...$ The sequence of polynomials $T_n(x)$ for $n \geq 2$ are obtained recursively by the three-term recurrence relation:

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \text{ for } n = 1, 2, ... \tag{2.2}$$

The first two terms of the recurrence relation are given by $T_0(x) = 1$ and $T_1(x) = x$.

From the above, we are able to deduce that $T_0(x) = T_1'(x)$ and $T_1(x) = \frac{1}{4}T_2'(x)$. Let us also note the following relations:

$$\begin{aligned} T_{n-1}'(x) &= (n-1)\sin\left((n-1)\cos^{-1}(x)\right)\omega(x), \text{ and} \\ T_{n+1}'(x) &= (n+1)\sin\left((n+1)\cos^{-1}(x)\right)\omega(x), \end{aligned} \tag{2.3}$$

where $\omega(x) = \frac{1}{\sqrt{x^2-1}}$ represents a weight function defined on the interval $x \in [-1, 1]$. Using what we know from (2.3), let us employ the sum-angle formula for sine to $\sin((n\pm1)\cos^{-1}(x))$ such that:

$$\begin{aligned} \implies \sin\left((n \pm 1)\cos^{-1}(x)\right) &= \sin(n\cos^{-1}(x) \pm \cos^{-1}(x)) \\ &= \sin(n\cos^{-1}(x))\cos(\cos^{-1}(x)) \pm \cos(n\cos^{-1}(x))\sin(\cos^{-1}(x)) \\ &= \sin(n\cos^{-1}(x))\cos(\cos^{-1}(x)) \pm T_n(x)\sqrt{1-x^2}. \end{aligned} \tag{2.4}$$

Now, with what we have discovered in (2.3) and (2.4), we are able to re-express the derivative of the Chebyshev polynomial of order $n \pm 1$ as the following:

$$\begin{aligned} \frac{T_{n-1}'(x)}{n-1} &= \frac{\sin(n\cos^{-1}(x))\cos(\cos^{-1}(x)) - T_n(x)\sqrt{1-x^2}}{\sqrt{1-x^2}}, \text{ and} \\ \frac{T_{n+1}'(x)}{n+1} &= \frac{\sin(n\cos^{-1}(x))\cos(\cos^{-1}(x)) + T_n(x)\sqrt{1-x^2}}{\sqrt{1-x^2}}, \end{aligned} \tag{2.5}$$

where $T_n(x)$ is the *nth* Chebyshev polynomial defined by (2.1). From the above relations, stems the key identity for $n \geq 2$:

$$T_n(x) = \frac{T_{n+1}'(x)}{2(n+1)} - \frac{T_{n-1}'(x)}{2(n-1)} = \frac{1}{2}\left[\frac{T_{n+1}'(x)}{n+1} - \frac{T_{n-1}'(x)}{n-1}\right]. \tag{2.6}$$

2.1.2 Differentiation of Chebyshev Polynomials

As a result of the previous relations, we are able to conclude the following derivative properties for Chebyshev polynomials:

$$\begin{aligned} T_0(x) &= T_1'(x), \\ T_1(x) &= \frac{1}{4}T_2'(x), \\ T_n(x) &= \frac{1}{2}\left[\frac{T_{n+1}'(x)}{n+1} - \frac{T_{n-1}'(x)}{n-1}\right],\ n \geq 2. \end{aligned} \tag{2.7}$$

Theorem 2.1.1. *We have the identity*

$$\frac{dT_n(x)}{dx} = 2n \sum_{\substack{p=0 \\ p+n\ odd}}^{n-1} \frac{T_p(x)}{c_p}, \tag{2.8}$$

for $n \geq 1$, where $c_0 = 2$ and $c_p = 1$ for $p \geq 1$. We interpret the formula as $\frac{dT_0(x)}{dx} = 0$ for $n = 0$.

Proof. Let us prove Theorem 2.1.1 by induction. **Base case:** For $n = 0$,

$$\frac{dT_0(x)}{dx} \overset{\checkmark}{=} 0. \tag{2.9}$$

For $n = 1$,

$$\frac{dT_1(x)}{dx} = 2 \sum_{\substack{p=0 \\ p+1\ \text{odd}}}^{0} \frac{T_p(x)}{c_p} = 2\frac{T_0(x)}{c_0} \overset{\checkmark}{=} T_0(x). \tag{2.10}$$

Thus, the base case has been established. Now, let us perform the inductive step.

Inductive step: For $n = n + 1$, using the last equation of (2.7),

$$\begin{aligned} \frac{dT_{n+2}(x)}{dx} &= 2(n+2)T_{n+1}(x) + \frac{(n+2)}{n}\frac{dT_n(x)}{dx} \\ &= 2(n+2)T_{n+1}(x) + \frac{(n+2)}{n}2n \sum_{\substack{p=0 \\ p+n\ \text{odd}}}^{n-1} \frac{T_p(x)}{c_p} \\ &\overset{\checkmark}{=} 2(n+2) \sum_{\substack{p=0 \\ p+n\ \text{odd}}}^{n+1} \frac{T_p(x)}{c_p} \end{aligned} \tag{2.11}$$

Thus, the result (2.9) for level n implies the result for level $n+2$. This proves the theorem since the formula for odd n is decoupled from the formula for even n (note that the derivative for an even $T_n(x)$ is odd, and the derivative of an odd $T_n(x)$ is even).

Conclusion: By the principle of induction, (2.8) is true for all integers $n \geq 0$. □

By Theorem 2.1.1, the following holds true:

$$\frac{d}{dx}[T_0(x), T_1(x), ..., T_N(x)] = [T_0(x), T_1(x), ..., T_N(x)]D, \tag{2.12}$$

where D is a dense, upper-triangular matrix known as the modal Chebyshev differentiation matrix. This relation implies that

$$\begin{aligned}\frac{d}{dx}[T_0(x), T_1(x), ..., T_N(x)] &= [T_0(x), T_1(x), ..., T_N(x)]D \\ &= [T_0(x), T_1(x), ..., T_N(x)] \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & \cdots & N \\ 0 & 0 & 4 & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & 2N \\ \vdots & \vdots & \vdots & \ddots & & & \vdots \\ \vdots & \vdots & \vdots & & \ddots & & 2N \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}\end{aligned} \tag{2.13}$$

for N-odd, where only nonzero entries are found above the main diagonal. The matrix structure in (2.13) has all zeros on and below the main diagonal. Above the main diagonal the matrix is full except for a staggered pattern of zeros. A similar formula is used for N-even. Here, each column of the differentiation matrix D represents the derivative of the Chebyshev polynomial of order n. As a result, the above relation illustrates that for a finite

expansion $u(x)$ and $\frac{du(x)}{dx}$ become:

$$\begin{aligned} u(x) &= \sum_{k=0}^{N} u_k T_k(x) \\ &= [T_0(x), T_1(x), ..., T_N(x)] \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix}, \end{aligned} \tag{2.14}$$

and

$$\begin{aligned} \frac{du(x)}{dx} &= \sum_{k=0}^{N} u'_k T_k(x) \\ &= [T_0(x), T_1(x), ..., T_N(x)] \begin{bmatrix} u'_0 \\ u'_1 \\ \vdots \\ u'_N \end{bmatrix}, \end{aligned} \tag{2.15}$$

where $u'_k = \sum_{l=0}^{N} D_{kl} u_l$ represents the expansion coefficients for the derivative.

2.1.3 Integration of Chebyshev Polynomials

To establish an integration method, recall the previously defined relation for the *nth* order Chebyshev polynomial with respect to its $n \pm 1$ derivatives for $n \geq 2$:

$$T_n(x) = \frac{T'_{n+1}(x)}{2(n+1)} - \frac{T'_{n-1}(x)}{2(n-1)} = \frac{1}{2}\left[\frac{T'_{n+1}(x)}{n+1} - \frac{T'_{n-1}(x)}{n-1}\right]. \tag{2.16}$$

The method of integration for Chebyshev polynomials is similar to differentiation. The Chebyshev integration operator is expressible as

$$B_{[1]} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & -\frac{1}{2} & 0 & \cdots & 0 \\ 0 & \frac{1}{4} & 0 & -\frac{1}{4} & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \frac{1}{2(N-1)} & 0 & -\frac{1}{2(N-1)} \\ 0 & 0 & 0 & 0 & \frac{1}{2N} & 0 \end{bmatrix}, \tag{2.17}$$

where $B_{[1]}$ is a $(N+1) \times (N+1)$. The notation [1] indicates that the first row has all zeros as entries. This matrix form is established from (2.7) such that the Chebyshev polynomials are represented by the following:

$$\begin{aligned} [T_0(x), T_1(x), ..., T_N(x)] &= [T_0'(x), T_1'(x), ..., T_N'(x)]B_{[1]} + \frac{T_{N+1}'(x)}{2(N+1)} e_N^T \\ &= [T_0'(x), T_1'(x), ..., T_N'(x)] \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & -\frac{1}{2} & 0 & \cdots & 0 \\ 0 & \frac{1}{4} & 0 & -\frac{1}{4} & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \frac{1}{2(N-1)} & 0 & -\frac{1}{2(N-1)} \\ 0 & 0 & 0 & 0 & \frac{1}{2N} & 0 \end{bmatrix} \\ &\quad + \frac{T_{N+1}'(x)}{2(N+1)} e_N^T. \end{aligned} \tag{2.18}$$

As a result, the integration of $u(x)$ yields the following relation provided $u_N = 0$:

$$\int u(x)dx = C + \sum_{k=0}^{N} \omega_k T_k(x), \tag{2.19}$$

where $\omega_k = \sum_{l=0}^{N} \left(B_{[1]}\right)_{kl} u_l$. Now, if $u_N \neq 0$ then the following expression is true:

$$\begin{aligned}
\int u(x)dx &= \int [T_0(x), T_1(x), ..., T_N(x)]dx \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{pmatrix} \\
&= C + [T_0(x), T_1(x), ..., T_N(x)]B_{[1]} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{pmatrix} + \frac{u_N}{2(N+1)} T_{N+1}(x). \qquad (2.20) \\
&= C + [T_0(x), T_1(x), ..., T_N(x)] \begin{pmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_N \end{pmatrix} + \frac{u_N}{2(N+1)} T_{N+1}(x),
\end{aligned}$$

where $\omega_k = \sum_{l=0}^{N} \left(B_{[1]}\right)_{kl} u_l$. The above results show that $B_{[1]}D = I_{[1]}$, where $I_{[1]}$ is the identity with the diagonal entry in the first row set to zero.

We showed that

$$\begin{aligned}
T_0(x) &= T_1'(x) \\
T_1(x) &= \frac{1}{4} T_2'(x) \qquad (2.21) \\
T_n(x) &= \frac{T_{n+1}'(x)}{2(n+1)} - \frac{T_{n-1}'(x)}{2(n-1)} \quad \text{for} \quad n \geq 2.
\end{aligned}$$

Although, we will not provide a proof, it can analogously be shown that

$$\begin{aligned}
T_0(x) &= \frac{1}{4}T_2''(x) \\
T_1(x) &= \frac{1}{24}T_3''(x) \\
T_2(x) &= \frac{1}{48}T_4''(x) - \frac{1}{6}T_2''(x) \\
T_n(x) &= \frac{T_{n+2}''(x)}{4(n+1)(n+2)} - \frac{T_n''(x)}{2(n^2-1)} + \frac{T_{n-2}''(x)}{4(n-1)(n-2)} \quad \text{for} \quad n \geq 3.
\end{aligned} \tag{2.22}$$

From the identities expressed in (2.22) or by squaring the expression for $B_{[1]}$ (as a matrix with infinite rows and columns) we have the following expression of the double Chebyshev integration matrix in the modal basis:

$$B_{[2]}^2 = \begin{bmatrix}
0 & 0 & 0 & & & & & & \\
0 & 0 & 0 & 0 & & & & & \\
\frac{1}{4} & 0 & -\frac{1}{6} & 0 & \frac{1}{24} & & & & \\
 & \frac{1}{24} & 0 & -\frac{1}{16} & 0 & \frac{1}{48} & & & \\
 & & \frac{1}{48} & 0 & -\frac{1}{30} & 0 & \frac{1}{80} & & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & & & \frac{1}{4(N-2)(N-3)} & 0 & -\frac{1}{2((N-2)^2-1)} & 0 & \frac{1}{4(N-2)(N-1)} \\
 & & & & & \frac{1}{4(N-1)(N-2)} & 0 & -\frac{1}{2((N-2)^2-1)} & 0 \\
 & & & & & & \frac{1}{4N(N-1)} & 0 & -\frac{1}{2(N^2-1)}
\end{bmatrix}. \tag{2.23}$$

The integration matrices $B_{[1]}$ and $B_{[2]}^2$ have been given for the standard interval $[-1, 1]$. For a general interval $[x_{\min}, x_{\max}]$ the corresponding matrices are obtained by

$$\begin{aligned}
B_{[1]} &\to \frac{1}{2}\left(x_{\max} - x_{\min}\right) B_{[1]} \\
B_{[2]}^2 &\to \frac{1}{4}\left(x_{\max} - x_{\min}\right)^2 B_{[2]}^2.
\end{aligned} \tag{2.24}$$

2.1.4 Modified Double Integration Matrix

We also have to consider the matrix $B^2_{[2]}$ presented in (2.23) with its first two columns set to zero:

$$\widetilde{B}^2_{[2]} = \begin{bmatrix} 0 & 0 & 0 & & & & & & \\ 0 & 0 & 0 & 0 & & & & & \\ 0 & 0 & -\frac{1}{6} & 0 & \frac{1}{24} & & & & \\ & 0 & 0 & -\frac{1}{16} & 0 & \frac{1}{48} & & & \\ & & \frac{1}{48} & 0 & -\frac{1}{30} & 0 & \frac{1}{80} & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & \frac{1}{4(N-2)(N-3)} & 0 & -\frac{1}{2((N-2)^2-1)} & 0 & \frac{1}{4(N-2)(N-1)} \\ & & & & & \frac{1}{4(N-1)(N-2)} & 0 & -\frac{1}{2((N-2)^2-1)} & 0 \\ & & & & & & \frac{1}{4N(N-1)} & 0 & -\frac{1}{2(N^2-1)} \end{bmatrix}. \tag{2.25}$$

Clearly, the difference between the above matrices is that the first two columns of (2.25) are set to zero, whereas the first two columns of (2.23) are not. As argued below, $\widetilde{B}^2_{[2]}$ is symmetrizable which means $\widetilde{B}^2_{[2]} = \widetilde{P}\Lambda_{[2]}\widetilde{P}^{-1}$ is diagonalizable by a simpler similarity transformation $\widetilde{P}$ than the one which diagonalizes $B^2_{[2]}$. This will be exploited later in our construction of a new modal-based preconditioner. Clearly, the canonical basis vectors $\mathbf{e}_0$ and $\mathbf{e}_1$ are eigenvectors of $\widetilde{B}^2_{[2]}$, which belong to the multiple eigenvalue $\lambda_0 = \lambda_1 = 0$. As a result, we may choose

$$\Lambda_{[2]} = \mathrm{diag}(0, 0, \lambda_2, ..., \lambda_N), \qquad \widetilde{P} = \widetilde{P}(0\colon N, 0\colon N) = \begin{bmatrix} 1 & & \\ & 1 & \\ & & \widetilde{P}(2\colon N, 2\colon N) \end{bmatrix}, \tag{2.26}$$

where the subscript [2] for $\Lambda_{[2]}$ indicates that the first two entries on the diagonal are zero. Here, $\widetilde{P}(2\colon N, 2\colon N)$ diagonalizes $\widetilde{B}^2_{[2]}(2\colon N, 2\colon N)$. From these formulas, it follows that

$$\widetilde{P} I_{[2]} \widetilde{P}^{-1} = I_{[2]}, \tag{2.27}$$

where $I_{[2]}$ is the identity matrix with its first two diagonal entries set to zero. The following argument explains why $\widetilde{B}^2_{[2]}$ is diagonalizable.

Let us consider the matrix

$$\widetilde{C} = S^{-1}\widetilde{B}^2_{[2]}S, \tag{2.28}$$

where $S^{-1} = \text{diag}(1, 1, \sqrt{2}, \sqrt{3}, ..., \sqrt{N})$. The matrix described in (2.28) is symmetric, thus diagonalized by a similarity transformation with an orthogonal matrix: $\widetilde{C} = Q\Lambda_{[2]}Q^T$. As a result, we have that

$$\widetilde{B}^2_{[2]} = (SQ)\Lambda_{[2]}(SQ)^{-1}, \tag{2.29}$$

where $\widetilde{P} = SQ$ and $\widetilde{P}^{-1} = (SQ)^{-1}$. Thus,

$$\widetilde{B}^2_{[2]} = \widetilde{P}\Lambda_{[2]}\widetilde{P}^{-1}. \tag{2.30}$$

Here,

$$\widetilde{P} = SQ = S[\mathbf{e}_0, \mathbf{e}_1, \mathbf{q}_2, ..., \mathbf{q}_N] = [\mathbf{e}_0, \mathbf{e}_1, S\mathbf{q}_2, ..., S\mathbf{q}_N]. \tag{2.31}$$

Since $Q = [\mathbf{e}_0, \mathbf{e}_1, \mathbf{q}_2, ..., \mathbf{q}_N]$ is orthogonal, $\mathbf{e}_0^T\mathbf{q}_k = 0 = \mathbf{e}_1^T\mathbf{q}_k$ for $k = 2, ..., N$. Thus, another calculation showing (2.27) is

$$\widetilde{P}I_{[2]}\widetilde{P}^{-1} = S\left(QI_{[2]}Q^T\right)S^{-1} = S\left(\sum_{k=2}^{N}\mathbf{q}_k\mathbf{q}_k^T\right)S^{-1} = SI_{[2]}S^{-1} = I_{[2]}. \tag{2.32}$$

Here, (2.32) exploits the fact that $I_{[2]} = QQ^T - \mathbf{e}_0\mathbf{e}_0^T + \mathbf{e}_1\mathbf{e}_1^T = \sum_{k=2}^{N}\mathbf{q}_k\mathbf{q}_k^T$. For a general interval $\widetilde{B}^2_{[2]} \to \frac{1}{4}\left(x_{\max} - x_{\min}\right)^2\widetilde{B}^2_{[2]}$.

2.2 Helmholtz Equation

For this work, let us consider the second-order Helmholtz problem posed on a rectangle subject to Dirichlet boundary conditions,

$$(\Delta \pm \kappa^2)u(x, y) = f(x, y) \qquad \text{and} \qquad u|_{\partial\Omega} = h, \tag{2.33}$$

where $(x, y) \in \Omega \equiv (x_{min}, x_{max}) \times (y_{min}, y_{max})$. In the above equation, the solution and source for the Helmholtz equation are represented by $u(x, y)$ and $f(x, y)$, respectively. More precisely, the Dirichlet boundary conditions, $u|_{\partial\Omega} = h$, are characterized by the following:

$$\begin{aligned} u(x_{min}, y) &= h_x^-(y), \qquad u(x_{max}, y) = h_x^+(y) \\ u(x, y_{min}) &= h_y^-(x), \qquad u(x, y_{max}) = h_y^+(x), \end{aligned} \tag{2.34}$$

with prescribed functions $h_x^{\pm}(y)$ and $h_y^{\pm}(x)$. We mention the possibility of "non-reflecting Sommerfeld boundary conditions" in the conclusion. The Laplacian is defined as $\Delta u(x, y) \equiv \partial_x^2 u(x, y) + \partial_y^2 u(x, y)$. Let us assume that $u(x, y)$ is evaluated by a Chebyshev series such that the solution is approximated as the following:

$$u(x, y) \approx \sum_{i=0}^{Nx} \sum_{j=0}^{Ny} u_{ij} T_i(\xi(x)) T_j(\eta(y)), \tag{2.35}$$

where u_{ij} are the modal coefficients and, $T_i(\xi(x))$ and $T_j(\eta(y))$ are Chebyshev basis functions. Moreover, $\xi(x)$ describes the linear mapping from $[x_{min}, x_{max}]$ to $[-1, 1]$. Likewise, for $\eta(y)$ and $[y_{min}, y_{max}]$, where N_x and N_y are the truncations corresponding to the respective directions. The bulk part of the Helmholtz equation presented in (2.33) can be approximated as

$$(D_x^2 \otimes I_y + I_x \otimes D_y^2 \pm \kappa^2 I_x \otimes I_y)\boldsymbol{u} = \boldsymbol{f}, \tag{2.36}$$

where D_x and D_y describe the modal differentiation matrices in the respective directions. Here, $\boldsymbol{u}$ is a finite collection of modal coefficients and is represented by a column vector of unknowns. The unknowns are indexed as

$$\boldsymbol{u}(\alpha) = u_{ij}, \qquad \alpha = i(N_y + 1) + j, \qquad 0 \le i \le N_x, \qquad 0 \le j \le N_y. \tag{2.37}$$

Moreover, the right-hand side of (2.36) represents the vector $\boldsymbol{f}$ of Chebyshev coefficients for a source function f. Thus, to further approximate (2.33), we must overwrite certain rows of (2.36) with *tau-conditions*, which correspond to the prescribed boundary conditions in (2.34). We achieve implementation of the boundary conditions through a technique known as integration sparsification[3].

2.2.1 Integration Sparsification

Let us adopt an alternative approach to approximating (2.35). More specifically, an integration sparsification method for spectral methods subject to *tau-conditions*. The proceeding work considers two matrices: D^k and $B^m_{[n]}$. Here, the matrix D^k is the modal-differentiation matrix, where k represents the order of differentiation. Additionally, the matrix $B^m_{[n]}$ is the modal integration matrix for Chebyshev polynomials, where m describes the m^{th}-order of integration in the Chebyshev basis. The subscript $[n]$ indicates that the first n rows of the matrix are empty. A list of properties for the aforementioned matrices is specified below:

1. The k^{th}-order modal differentiation matrix is a dense upper triangular matrix,
2. The m^{th}-order Chebyshev integration matrix is sparse and banded with upper and lower bandwidth, m,
3. $B^m_{[n]} D^k = B^{m-k}_{[n]}$ for $n \geq m \geq k$.

For the integration matrix, if $m = 0$ then $B^0_{[n]} \equiv I_{[n]}$, where $I_{[n]}$ is the identity matrix, with its first n entries on the diagonal set to zero.

While the matrix on the left-hand side of (2.36) has some sparsity due to its particular direct product structure, it involves factors which are dense and upper triangular. However, to achieve a fully sparse system of equations, we employ statement 3 from above and apply a Kronecker-product of Chebyshev integration matrices to Equation (2.36), thus

$$\mathcal{B}\boldsymbol{u} = \left(B^2_{x[2]} \otimes B^2_{y[2]}\right) \boldsymbol{f}. \tag{2.38}$$

Moreover, let us observe that $I_{[2]} = B^2_{[2]} D^2$. Thus, after integration sparsification the bulk operator simplifies to

$$\mathcal{B} = I_{x[2]} \otimes B^2_{y[2]} + B^2_{x[2]} \otimes I_{y[2]} \pm \kappa^2 B^2_{x[2]} \otimes B^2_{y[2]}. \tag{2.39}$$

The matrix $\mathcal{B}$ is a $n \times n$ matrix of rank $n-m$, where $n = (N_x+1)(N_y+1)$ and $m = 2(N_x+N_y)$. Therefore, the matrix $\mathcal{B}$ is of rank $(N_x - 1)(N_y - 1)$, which corresponds to the number of nonzero rows. The described process of applying $B^2_{x[2]} \otimes B^2_{y[2]}$ has been described as "integration preconditioning." However, our focus is on the sparsifying aspect of this process. In any case, the issue of conditioning is more subtle. As a result of this process, note that the matrix $\mathcal{B}$ can be applied to a vector with $\mathcal{O}(n)$ cost.

To implement the spectral tau-method, the zero rows in $\mathcal{B}$ are overwritten by *tau-conditions* which are responsible for enforcing the boundary conditions of the PDE along the edges of the rectangular domain. This process yields a nonsingular, linear system. Note that potential repetition of boundary data may arise, so some caution is needed. Figure 2.1 illustrates the Dirichlet boundary conditions imposed on the $xy-$faces of a rectangle to solve the 2D Helmholtz Equation.

As previously mentioned, Chebyshev polynomials of the first kind are chosen as basis functions to approximate the Helmholtz Equation, $T_i(\xi(x))$ and $T_j(\eta(y))$. Here,

$$\xi\colon [x_{min}, x_{max}] \mapsto [-1, 1] \qquad \text{and} \qquad \eta\colon [y_{min}, y_{max}] \mapsto [-1, 1]. \tag{2.40}$$

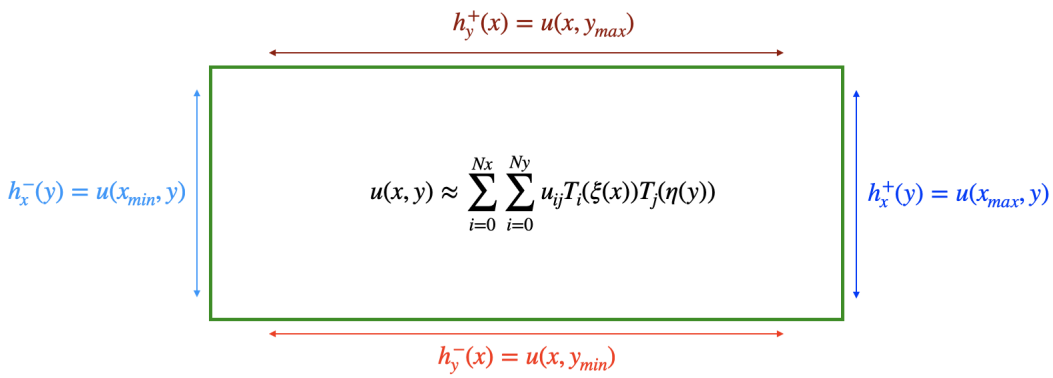

Figure 2.1: Dirichlet boundary conditions imposed along a rectangle to solve the 2D Helmholtz Equation.

The Dirichlet boundary conditions along the rectangular domain are approximated as

$$\begin{aligned} \sum_{j=0}^{N_y} u_{ij}\delta_j^{\pm} &= h_{yi}^{\pm}, \qquad i = 0, \ldots, N_x, \\ \sum_{i=0}^{N_x} u_{ij}\delta_i^{\pm} &= h_{xj}^{\pm}, \qquad j = 0, \ldots, N_y - 2 \end{aligned} \tag{2.41}$$

where all empty rows of (2.38) are replaced by these *tau-conditions.* Notice that it is the Chebyshev coefficients for expansions of $h_y^{\pm}(x)$ and $h_x^{\pm}(y)$ that appear on the right-side of (2.41). Moreover, the Dirichlet vectors are represented by $\delta^+ = [1, 1, ..., 1, 1]$ and $\delta^- = [1, -1, ..., -1, 1]$. Furthermore, from the equations in (2.41), if $N_x \sim N \sim N_y$ then filling each tau-row with a Dirichlet vector amounts to $\mathcal{O}(N)$ nonzero entries. Since there are $\mathcal{O}(N)$ tau-rows, overall enforcement of the Dirichlet boundary conditions contributes $\mathcal{O}(N^2) = \mathcal{O}(n)$ nonzero entries to the coefficient matrix.

For example, if

$$u(x, y) \approx \sum_{i=0}^{Nx} \sum_{i=0}^{Ny} u_{ij} T_i(\xi(x)) T_j(\eta(y)) \tag{2.42}$$

then

$$\begin{aligned} u(x_{min}, y) &= \sum_{j=0}^{Ny} \left\{ \sum_{i=0}^{Nx} u_{ij}\delta_i^- \right\} T_j(\eta(y)) \\ &= \sum_{j=0}^{Ny} h_{xj}^- T_j(\eta(y)) \\ &= h_x^-(y). \end{aligned} \tag{2.43}$$

The full set of *tau-conditions* is the following (four more equations than in (2.41)):

$$\begin{aligned} \sum_{j=0}^{N_y} u_{ij}\delta_j^{\pm} &= h_{yi}^{\pm}, \qquad i = 0, \ldots, N_x, \\ \sum_{i=0}^{N_x} u_{ij}\delta_i^{\pm} &= h_{xj}^{\pm}, \qquad j = 0, \ldots, N_y. \end{aligned} \tag{2.44}$$

Therefore, the number of possible equations $2(N_x + N_y + 2)$. However, this number is four more than the number of rows with zero entries in $\mathcal{B}$. However, there are 4 linear dependencies amongst the set of equations, which correspond to the fact that the edges share common corner values. The four corner relations can be expressed by the following:

$$\begin{aligned} h_x^+(y_{max}) &= h_y^+(x_{max}), \qquad h_x^+(y_{min}) = h_y^-(x_{max}) \\ h_x^-(y_{min}) &= h_y^-(x_{min}), \qquad h_x^-(y_{max}) = h_y^+(x_{min}). \end{aligned} \tag{2.45}$$

The *tau-conditions* in (2.44) are a linearly independent set of equations.

2.2.2 Coefficient Matrix

The coefficient matrix for the approximation of (2.33) is expressed as

$$\mathcal{M} = \mathcal{B} + \mathcal{U}\mathcal{V}^T, \tag{2.46}$$

where the rank-augmenting perturbation matrix enforces Dirichlet boundary conditions. The matrix $\mathcal{U}$ has as its columns canonical basis vectors which insert the corresponding rows of $\mathcal{V}^T$ into the tau-rows of $\mathcal{M}$. The rows of $\mathcal{V}^T$ stem from the Dirichlet vectors. The rank-augmenting matrix, $\mathcal{U}\mathcal{V}^T$, is of rank:

$$rank(\mathcal{U}) = m = 2(N_x + N_y) = rank(\mathcal{V}).$$

To describe the coefficient matrix we will adopt the clumped index notation such that

$$\mathcal{M}(\alpha, \beta) = \mathcal{M}(\alpha_{ij}, \beta_{pq}) = \mathcal{M}(i(N_y + 1) + j, p(N_y + 1) + q), \tag{2.47}$$

where $\alpha = i(N_y + 1) + j$ and $\beta = p(N_y + 1) + q$. Let us recall the *tau-conditions* expressed in (2.41). The index structure presented in Section 2.2.1, prevents the adoption of redundant boundary conditions. Thus, the *tau-conditions* used to enforce the Dirichlet boundary conditions are expressed in (2.44).

Based on the row-filling method of *tau-conditions*, our approximation of (2.33) is

$$\begin{aligned}\mathcal{M}(\alpha_{ij},:)\boldsymbol{u} = B^2_{x[2]}(i,:)\otimes B^2_{y[2]}(j,:)\boldsymbol{f} + \delta_{j0}h^-_{yi} + \delta_{j1}h^+_{yi} \\ + \delta_{i0}(1-\delta_{j0})(1-\delta_{j1})h^-_{x,j-2} + \delta_{i1}(1-\delta_{j0})(1-\delta_{j1})h^+_{x,j-2},\end{aligned} \tag{2.48}$$

where Chebyshev projections of the boundary data are present on the right-hand side. Moreover, let us account for the Kronecker-products involving Dirichlet vectors of the form for $j = 0$ and $j = 1$:

$$\begin{aligned}\mathcal{M}(\alpha_{i0},\beta_{pq}) &= \delta_{ip}\delta^-_q \\ \mathcal{M}(\alpha_{i1},\beta_{pq}) &= \delta_{ip}\delta^+_q.\end{aligned} \tag{2.49}$$

Thus, if $j = 0$ then

$$\begin{aligned}\mathcal{M}(\alpha_{i0},:)\boldsymbol{u} &= \sum_{p=0}^{N_x}\sum_{q=0}^{N_y}\mathcal{M}(\alpha_{i0},\beta_{pq})\boldsymbol{u}\left(p(N_y+1)+q\right) \\ &= \sum_{p=0}^{N_x}\sum_{q=0}^{N_y}\mathcal{M}(\alpha_{i0},\beta_{pq})u_{pq} \\ &= \sum_{q=0}^{N_y}\delta^-_q u_{iq}.\end{aligned} \tag{2.50}$$

The above relation yields the left-hand-side of the first equation in (2.41). A similar argument can be shown for $j = 1$. Now, take $i = 0$ and $i = 1$ for $j = 2, ..., N_y$ such that

$$\begin{aligned}\mathcal{M}(\alpha_{0j},\beta_{pq}) &= \delta_{j-2,q}\delta^-_p \\ \mathcal{M}(\alpha_{1j},\beta_{pq}) &= \delta_{j-2,q}\delta^+_p\end{aligned} \tag{2.51}$$

.

Now, if $i = 0$ then

$$\begin{aligned}\mathcal{M}(\alpha_{0j},:)\boldsymbol{u} &= \sum_{p=0}^{N_x}\sum_{q=0}^{N_y}\mathcal{M}(\alpha_{0j},\beta_{pq})\boldsymbol{u}\left(p(N_y+1)+q\right) \\ &= \sum_{p=0}^{N_x}\sum_{q=0}^{N_y}\mathcal{M}(\alpha_{0j},\beta_{pq})u_{pq} \\ &= \sum_{p=0}^{N_x}\delta^-_p u_{p,j-2}.\end{aligned} \tag{2.52}$$

The above relation yields the right-hand-side of the first equation in (2.44). Alternatively, if $j = 0, ..., N_y - 2$ then

$$\begin{aligned} \mathcal{M}(\alpha_{0j+2}, \beta_{pq}) &= \delta_{jq}\delta_q^- \\ \mathcal{M}(\alpha_{1j+2}, \beta_{pq}) &= \delta_{jq}\delta_q^+. \end{aligned} \tag{2.53}$$

To summarize our modal spectral approximation of (2.33) and (2.34) is

$$\mathcal{M}\boldsymbol{u} = \boldsymbol{g}, \tag{2.54}$$

where $\mathcal{M}$ is described above and $\boldsymbol{g}$ is the right-hand side of (2.48), that is the right-hand side of (2.38) supplemented with the boundary values.

Chapter 3

The Generalized Minimal Residual Method: Solution Approach

Suppose $A\mathbf{x} = \mathbf{b}$, where A is an $n \times n$ invertible matrix and $\mathbf{b}$ is an $n \times 1$ vector. Instead of solving the system directly, let us employ an iterative method known as the generalized minimal residual method (GMRES). This method is used to find the best approximate solution, which minimizes the residual over the k-th order Krylov subspace. Let us denote the k-th order Krylov subspace by $\mathcal{K}_k = \mathcal{K}_k(A, \mathbf{r}_0)$, where $\mathbf{r}_0$ represents the zeroth residual. Here, the residual is denoted by

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k. \tag{3.1}$$

The idea is to use GMRES to approximate the exact solution of $\mathbf{x} = A^{-1}\mathbf{b}$ by the k-th iterate $\mathbf{x}_k \in x_0 + \mathcal{K}_k$ such that the Euclidean norm of the residual is minimized. Ideally, an exact solution to the linear system is returned when $k \ll n$. Though routinely, the algorithm returns an approximate solution once the residual is sufficiently small.

The k-th Krylov subspace is defined by

$$\mathcal{K}_k(A, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, ..., A^{k-1}\mathbf{r}_0\}, \tag{3.2}$$

where the initial error is represented by $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ for $k = 0$. For the following computations, let us assume that the initial guess is equal to zero. Clearly, if $\mathbf{x}_0 = 0$ then $\mathbf{r}_0 = \mathbf{b}$. Fast matrix-vector products are needed to efficiently generate the Krylov sequence. The k-th GMRES iterate $\mathbf{x}_k$ is the minimizer of the problem

$$\mathbf{x}_k = \underset{\mathbf{x}\in\mathcal{K}_k}{\arg\min} \|\mathbf{b} - A\mathbf{x}\|_2 . \tag{3.3}$$

From the nested property $\mathcal{K}_k(A, \mathbf{b}) \subseteq \mathcal{K}_{k+1}(A, \mathbf{b})$, the best solution approximate to the linear system tends to become more accurate with each iteration. However, the basis vectors encompassing $\mathcal{K}_k(A, \mathbf{b})$ may become nearly colinear as k increases making the problem ill-conditioned. To remedy this situation, the algorithm is instead implemented with the Arnoldi process to construct an orthonormal basis for $\mathcal{K}_k(A, \mathbf{b})$ as the iterations progress.

Let the orthonormal basis vectors of $\mathcal{K}_k(A, \mathbf{b})$ be stored as column vectors in the matrix $Q_k \in \mathbb{R}^{n\times k}$. Thus, $\mathbf{x}_k = Q_k\mathbf{y}_k$ and the minimization problem becomes

$$\mathbf{y}_k = \underset{\mathbf{y}\in\mathbb{R}^k}{\arg\min} \|\mathbf{b} - AQ_k\mathbf{y}\|_2 . \tag{3.4}$$

From the Arnoldi method, the matrix A is reduced to $AQ_k = Q_{k+1}H_k$ by a partial Hessenberg reduction, where $H_k \in \mathbb{R}^{(k+1\times k)}$. Therefore,

$$\underset{\mathbf{y}\in\mathbb{R}^k}{\arg\min} \|\mathbf{b} - AQ_k\mathbf{y}\|_2 = \underset{\mathbf{y}\in\mathbb{R}^k}{\arg\min} \|\mathbf{b} - Q_{k+1}H_k\mathbf{y}\|_2 = \underset{\mathbf{y}\in\mathbb{R}^k}{\arg\min} \left\| Q_{k+1}^T\mathbf{b} - H_k\mathbf{y} \right\|_2 . \tag{3.5}$$

Since the first orthonormal basis vector of Q_k is $\mathbf{q}_1 = \dfrac{\mathbf{b}}{\|\mathbf{b}\|_2}$, we have that $Q_{k+1}^T\mathbf{b} = \|\mathbf{b}\|_2 \mathbf{e}_1 \in \mathbb{R}^{k+1}$. As a result, (3.5) becomes

$$\mathbf{y}_k = \underset{\mathbf{y}\in\mathbb{R}^k}{\arg\min} \|\beta\mathbf{e}_1 - H_k\mathbf{y}\|_2 . \tag{3.6}$$

Here, $\beta = \|\mathbf{b}\|$ and the vector $\mathbf{e}_1 = (1, 0, 0, ..., 0)^T$ is of length $k + 1$. Thus, the minimizer of Equation (3.6) is $\mathbf{y}_k$ and the k-th GMRES iterate is expressed as $\mathbf{x}_k = Q_k\mathbf{y}_k$ for $\mathbf{x}_0 = \mathbf{0}$. An outline of the algorithm can be found in Section 3.1.

3.1 The GMRES Algorithm

The GMRES algorithm operates on a vector $\mathbf{b}$ of length n and A, where A is typically a large sparse, nonsingular $n \times n$ matrix. The algorithm typically iterates over a Krylov subspace for k steps or until the residual is less than the tolerance. Thereupon, an approximate solution to $A\mathbf{x} = \mathbf{b}$ is returned along with the relative residual.

Algorithm 1 The GMRES Algorithm.

Input: $A, \mathbf{b}, M^{-1}, \mathbf{x}_0, k, TOL$
Output: $\mathbf{x}_k, \rho$
$Q \leftarrow zeros(size(\mathbf{b}), k)$
$H \leftarrow zeros(k+1, k)$
$\mathbf{r}_0 \leftarrow \mathbf{b} - A\mathbf{x}_0$
$Q(:, 0) = \mathbf{r}_0 / \|\mathbf{r}_0\|_2$
$\beta = \|\mathbf{r}_0\|$
$\rho = \beta$
$i = 0$
while $i < k$
$i = i + 1$
$\mathbf{q}_{i+1} = A\mathbf{q}_i$ (if preconditioner applied, also $\mathbf{q}_{i+1} \leftarrow M^{-1}\mathbf{q}_{i+1}$)
for $j = 1 \ldots i$
$h_{ji} = \mathbf{q}_{i+1}^T \mathbf{q}_j$
$\mathbf{q}_{i+1} \leftarrow \mathbf{q}_{i+1} - h_{ji}\mathbf{q}_j$
end
$h_{i+1,i} = ||\mathbf{q}_{i+1}||_2$
$\mathbf{q}_{i+1} \leftarrow \mathbf{q}_{i+1}/h_{i+1,i}, \quad Q(:, i+1) = \mathbf{q}_{i+1}$
$\mathbf{e}_1 = (1, 0, 0, ..., 0)^T \in \mathbb{R}^{i+1}$
$\mathbf{y}_i = \underset{\mathbf{y} \in \mathbb{R}^i}{\arg\min} \|\beta\mathbf{e}_1 - H_i\mathbf{y}\|$
$\rho = \left\|\beta\mathbf{e}_1 - H_i\mathbf{y}_i\right\|$
end
$\mathbf{x}_k = \mathbf{x}_0 + Q(:, 1{:}\,k)\mathbf{y}_k$

As demonstrated above, a main expense of the algorithm originates from the matrix-vector product of $A\mathbf{q}_i$ or $M^{-1}A\mathbf{q}_i$ with a left preconditioner, which is computed once per iteration. Generically, the cost of k-iterations is $\mathcal{O}(kn^2)$. However, if A is sparse then the cost is $\mathcal{O}(kn)$, assuming either a sparse preconditioner or no preconditioner. For our problem, we have a

sparse matrix A and a preconditioner M^{-1}, which costs $\mathcal{O}(n^{3/2})$ to apply. Thus, we expect a total cost of $\mathcal{O}(kn^{3/2})$ for k matrix-vector multiplies.

The cost for k-iterations of the Arnoldi iteration process is $\mathcal{O}(k^2n)$, as shown by the following argument. The cost for a single iteration of the for loop between lines 13 and 16 is $\mathcal{O}(in)$. Thus, the cumulative cost for k-iterations is $\mathcal{O}(k^2n)$.

Lastly, we consider the cost of the least squares solves, as described in line 20. Here, this cost cannot be inferred from the given algorithm provided. However, Kelley [4] shows that if information from the previous least squares solve is used for the current least squares solve, then the cost is $\mathcal{O}(i)$ for a single iteration. Thus, the cumulative cost is $\mathcal{O}(k^2)$. If performed efficiently, the cost of performing the least squares solves is subdominant compared to the Gram-Schmidt/ Arnoldi process.

In Section 3.2, a preconditioner with a cost of $\mathcal{O}(n^{3/2})$ is considered. The cumulative cost of the unpreconditioned and preconditioned system for k-iterations of GMRES breaks down to the following:

	Matrix-vector	Arnoldi Iteration	Least Squares
Unpreconditioned:	$\mathcal{O}(kn)$	$\mathcal{O}(k^2n)$	$\mathcal{O}(k^2)$
Preconditioned:	$\mathcal{O}(kn^{3/2})$	$\mathcal{O}(k^2n)$	$\mathcal{O}(k^2)$

Table 3.1: Cumulative cost for k iterations of GMRES

As demonstrated above, the efficiency of the algorithm is dependent on the number of iterations it takes to solve the linear system.

3.2 Preconditioning

3.2.1 GMRES Preconditioning

Fundamentally, GMRES computes the best approximate solution $x_k \in \mathcal{K}_k$ to $A\mathbf{x} = \mathbf{b}$ over a Krylov subspace for $\mathbf{x}_0 = \mathbf{0}$. The algorithm is said to be monotonically convergent since

$\|\mathbf{r}_{k+1}\| \leq \|\mathbf{r}_k\|$. For large linear systems, a preconditioner is often used to increase the rate convergence and ensure the accuracy of the best approximate solution in terms of the forward error. For our problem, the coefficient matrix is represented by

$$\mathcal{M} = \mathcal{B} + \mathcal{U}\mathcal{V}^T \tag{3.7}$$

where, the bulk operator is defined by

$$\mathcal{B} = I_{x[2]} \otimes B^2_{y[2]} + B^2_{x[2]} \otimes I_{y[2]} \pm \kappa^2 B^2_{x[2]} \otimes B^2_{y[2]}. \tag{3.8}$$

The rank-augmenting perturbation $\mathcal{U}\mathcal{V}^T$ is responsible for enforcing the Dirichlet boundary conditions. Here, $\mathcal{U}$ and $\mathcal{V}$ are both sparse matrices with $\mathcal{U}\mathcal{V}^T$ having $\mathcal{O}(n)$ nonzero entries. The preconditioner $\widetilde{\mathcal{M}}^{-1}$ involves inversion of an approximation

$$\widetilde{\mathcal{M}} = \widetilde{\mathcal{B}} + \mathcal{U}\mathcal{V}^T \tag{3.9}$$

of the coefficient matrix $\mathcal{M}$. Here, the bulk operator of the approximation is defined by

$$\widetilde{\mathcal{B}} = I_{x[2]} \otimes \widetilde{B}^2_{y[2]} + \widetilde{B}^2_{x[2]} \otimes I_{y[2]} \pm \kappa^2 \widetilde{B}^2_{x[2]} \otimes \widetilde{B}^2_{y[2]}, \tag{3.10}$$

where the $\widetilde{B}^2_{[2]}$ matrices stem from (2.25) appropriately scaled by the interval length. Thus, an efficient way to apply the inverse of (3.9) to a vector is needed. Before inverting (3.9), let us diagonalize $\widetilde{\mathcal{B}}$. Let us use the similarity transformation defined by $\widetilde{\mathcal{P}} = \widetilde{P}_x \otimes \widetilde{P}_y$. As discussed in Section 2.1.6, $\widetilde{B}^2_{x[2]}$ and $\widetilde{B}^2_{y[2]}$ are diagonalizable by

$$\widetilde{B}^2_{x[2]} = \widetilde{P}_x \Lambda_{x[2]} \widetilde{P}_x^{-1} \quad \text{and} \quad \widetilde{B}^2_{y[2]} = \widetilde{P}_y \Lambda_{y[2]} \widetilde{P}_y^{-1}. \tag{3.11}$$

As a result,

$$\begin{aligned}
\widetilde{\mathcal{B}} &= \left(\widetilde{B}^2_{x[2]} \otimes I_{y[2]}\right) + \left(I_{x[2]} \otimes \widetilde{B}^2_{y[2]}\right) \pm \left(\kappa^2 \widetilde{B}^2_{x[2]} \otimes \widetilde{B}^2_{y[2]}\right) \\
&= \left(\widetilde{P}_x \Lambda_{x[2]} \widetilde{P}_x^{-1} \otimes I_{y[2]}\right) + \left(I_{x[2]} \otimes \widetilde{P}_y \Lambda_{y[2]} \widetilde{P}_y^{-1}\right) \pm \left(\kappa^2 \widetilde{P}_x \Lambda_{x[2]} \widetilde{P}_x^{-1} \otimes \widetilde{P}_y \Lambda_{y[2]} \widetilde{P}_y^{-1}\right) \\
&= \widetilde{P}_x \otimes \widetilde{P}_y \left(\Lambda_{x[2]} \otimes I_{y[2]} + I_{x[2]} \otimes \Lambda_{y[2]} \pm \kappa^2 \Lambda_{x[2]} \otimes \Lambda_{y[2]}\right) \widetilde{P}_x^{-1} \otimes \widetilde{P}_y^{-1} \\
&= \widetilde{\mathcal{P}} \mathbf{\Lambda} \widetilde{\mathcal{P}}^{-1},
\end{aligned} \tag{3.12}$$

is true. The calculations in (3.12) rely on (2.27). Similarly, the pseudoinverse of the bulk operator can be expressed as: $\widetilde{\mathcal{B}}^{\dagger} = \widetilde{\mathcal{P}}\mathbf{\Lambda}^{\dagger}\widetilde{\mathcal{P}}^{-1}$. Here, the pseudoinverse of $\mathbf{\Lambda}^{\dagger}$ is found by taking the reciprocal of the nonzero elements of $\mathbf{\Lambda}$ on the diagonal.

Theorem 3.2.1. *The inverse of* $\widetilde{\mathcal{M}}$ *is*

$$\widetilde{\mathcal{M}}^{-1} = [\widetilde{\mathcal{B}}^{\dagger} - \mathcal{U}(\mathcal{V}^T\mathcal{U})^{-1}\mathcal{V}^T\widetilde{\mathcal{B}}^{\dagger} + \mathcal{U}(\mathcal{V}^T\mathcal{U})^{-1}\mathcal{U}^T]. \tag{3.13}$$

This is the generalized version of the Woodbury matrix identity. This theorem and its proof is from [6].

Proof. First, let us observe the following relation between the bulk operator and the pseudoinverse of the bulk operator,

$$\begin{aligned}\widetilde{\mathcal{B}}^{\dagger}\widetilde{\mathcal{B}} &= \widetilde{\mathcal{P}}\mathbf{\Lambda}^{\dagger}\mathbf{\Lambda}\widetilde{\mathcal{P}}^{-1} \\ &= \left(\widetilde{P}_x \otimes \widetilde{P}_y\right)\left(I_{x[2]} \otimes I_{y[2]}\right)\left(\widetilde{P}_x^{-1} \otimes \widetilde{P}_y^{-1}\right) \\ &= I_{x[2]} \otimes I_{y[2]}.\end{aligned} \tag{3.14}$$

The last equality stems from the fact that $\widetilde{P}_x I_{x[2]} \widetilde{P}_x^{-1} = I_{x[2]}$ as seen in (2.27). This shows that the orthogonal projector onto the column space of $\widetilde{\mathcal{B}}$ is given by

$$\widetilde{\mathcal{B}}^{\dagger}\widetilde{\mathcal{B}} = \widetilde{\mathcal{B}}\widetilde{\mathcal{B}}^{\dagger} = \mathcal{I} - \mathcal{U}\mathcal{U}^{\mathcal{T}}, \tag{3.15}$$

which represents an orthogonal splitting of two complementary subspaces. Moreover, since $\mathcal{U}\mathcal{U}^{\mathcal{T}}$ represents the orthogonal projector onto $\mathrm{col}(\widetilde{\mathcal{B}})^{\perp}$ we have that

$$\begin{aligned}\widetilde{\mathcal{M}}\widetilde{\mathcal{M}}^{-1} &= [\widetilde{\mathcal{B}} + \mathcal{U}\mathcal{V}^T][\widetilde{\mathcal{B}}^{\dagger} - \mathcal{U}(\mathcal{V}^T\mathcal{U})^{-1}\mathcal{V}^T\widetilde{\mathcal{B}}^{\dagger} + \mathcal{U}(\mathcal{V}^T\mathcal{U})^{-1}\mathcal{U}^T] \\ &= \widetilde{\mathcal{B}}\widetilde{\mathcal{B}}^{\dagger} - \widetilde{\mathcal{B}}\mathcal{U}(\mathcal{V}^T\mathcal{U})^{-1}\mathcal{V}^T\widetilde{\mathcal{B}}^{\dagger} + \widetilde{\mathcal{B}}\mathcal{U}(\mathcal{V}^T\mathcal{U})^{-1}\mathcal{U}^T + \mathcal{U}\mathcal{V}^T\widetilde{\mathcal{B}}^{\dagger} \\ &\quad - \mathcal{U}\mathcal{V}^T\mathcal{U}(\mathcal{V}^T\mathcal{U})^{-1}\mathcal{V}^T\widetilde{\mathcal{B}}^{\dagger} + \mathcal{U}\mathcal{V}^T\mathcal{U}(\mathcal{V}^T\mathcal{U})^{-1}\mathcal{U}^T \\ &= \widetilde{\mathcal{B}}\widetilde{\mathcal{B}}^{\dagger} - \widetilde{\mathcal{B}}\mathcal{U}(\mathcal{V}^T\mathcal{U})^{-1}\mathcal{V}^T\widetilde{\mathcal{B}}^{\dagger} + \widetilde{\mathcal{B}}\mathcal{U}(\mathcal{V}^T\mathcal{U})^{-1}\mathcal{U}^T + \mathcal{U}\mathcal{V}^T\widetilde{\mathcal{B}}^{\dagger} \\ &\quad - \mathcal{U}\mathcal{V}^T\widetilde{\mathcal{B}}^{\dagger} + \mathcal{U}\mathcal{U}^T \\ &= \widetilde{\mathcal{B}}\widetilde{\mathcal{B}}^{\dagger} + \mathcal{U}\mathcal{U}^T.\end{aligned} \tag{3.16}$$

The above relation makes use of the fact that the product of $\widetilde{\mathcal{B}}\mathcal{U}$ is an $n \times m$ matrix filled entirely of zeros. Moreover, the $m \times m$ matrix $\mathcal{V}^T\mathcal{U}$ is a nonsingular, as shown later. □

3.2.2 Fast Application of Kronecker Products

Suppose $\mathcal{T} = T_x \otimes T_y$. The application of $\mathcal{T}$ or $\widetilde{\mathcal{P}}$ in the analysis above to a vector $\boldsymbol{x}$ involves a computation of the form

$$\boldsymbol{z} = T_x \otimes T_y \boldsymbol{x}, \tag{3.17}$$

such that

$$\boldsymbol{z}(i(N_y+1)+j) = \sum_{p=0}^{N_x}\sum_{q=0}^{N_y} T_x(i,p)T_y(j,q)\boldsymbol{x}(p(N_y+1)+q). \tag{3.18}$$

Provided $N_x \approx N_y$, the matrix-vector products $\mathcal{T}\boldsymbol{x}$ and $\mathcal{T}^{-1}\boldsymbol{x}$ are computable at $\mathcal{O}(N^4) = \mathcal{O}(n^2)$ complexity.

Theorem 3.2.2. *The product* (3.17) *can be alternatively computed at* $\mathcal{O}(n^{3/2})$ *complexity.*

Proof. Instead of computing the product (3.17) as a matrix-vector product of two sums, let us split the computation into two sub-components such that

$$\boldsymbol{y}(i(N_y+1)+j) = \sum_{q=0}^{N_y} T_y(j,q)\boldsymbol{x}(i(N_y+1)+q), \tag{3.19}$$

$$\boldsymbol{z}(i(N_y+1)+j) = \sum_{p=0}^{N_x} T_x(i,p)\boldsymbol{y}(p(N_y+1)+j). \tag{3.20}$$

Consequently, each matrix-vector product is computable at $\mathcal{O}(N^3) = \mathcal{O}(n^{3/2})$ complexity. □

3.2.3 Inversion of $\mathcal{V}^T\mathcal{U}$ at $\mathcal{O}(n^{1/2})$ Cost

Let us recall the inverse of the matrix $(\mathcal{V}^T\mathcal{U})^{-1}$ appearing in Theorem (3.2.1). Let us consider the linear system,

$$\mathcal{V}^T\mathcal{U}\boldsymbol{y} = \boldsymbol{z}. \tag{3.21}$$

Let us apply $\mathcal{U}$ to both sides of equation (3.21) such that:

$$\mathcal{V}^T\mathcal{U}\boldsymbol{y} = \boldsymbol{z} \iff \mathcal{U}\mathcal{V}^T\boldsymbol{u} = \boldsymbol{f}, \tag{3.22}$$

where $\boldsymbol{u} = \mathcal{U}\boldsymbol{y}$ and $\boldsymbol{f} = \mathcal{U}\boldsymbol{z}$ are both in the column space of $\mathcal{U}$. Here, the matrix $\mathcal{U}\mathcal{V}^T$ is responsible for enforcing the *tau-conditions*. Moreover, the vectors $\boldsymbol{u}$ and $\boldsymbol{f}$ have all zero entries except for those entries corresponding to the *tau-conditions*. This indicates that $\boldsymbol{u}(\alpha) = \boldsymbol{u}(i(N_y+1)+j) = u_{ij} = 0$ for all entries except when either of i, j is 0 or 1, or both are. The boundary conditions for the linear system described in (3.22) are expressible as

$$\sum_{j=0}^{N_y} u_{ij}\delta_j^- = f_{i0}, \qquad \sum_{j=0}^{N_y} u_{ij}\delta_j^+ = f_{i1}, \qquad i = 0, \ldots, N_x, \tag{3.23}$$

$$\sum_{i=0}^{N_x} u_{ij}\delta_i^- = f_{0,j+2}, \qquad \sum_{i=0}^{N_x} u_{ij}\delta_i^+ = f_{1,j+2}, \qquad j = 0, \ldots, N_y - 2. \tag{3.24}$$

To efficiently solve the linear system, the equations above correspond to a particular row-filling pattern illustrated in Figure 3.1. First, the coefficients in *region A* are recovered then *region B* and *region C*, and lastly *region D*.

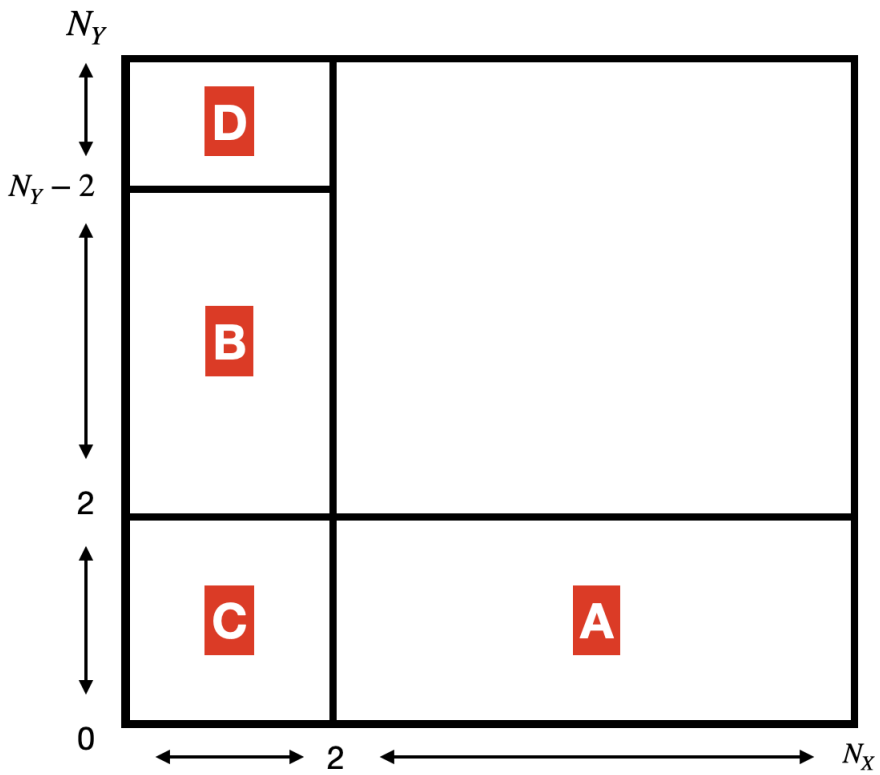

Figure 3.1: Recovery of Modal Coefficients.

To begin we will consider the coefficients in *region A*. Assuming $2 \leq i \leq N_x$ from (3.23), the j sums must range over only 0 and 1 (other terms in the sum are zero). These unknowns are then recoverable from (3.23) expressed as

$$u_{i0}\delta_0^- + u_{i1}\delta_1^- = f_{i0}, \qquad u_{i0}\delta_0^+ + u_{i1}\delta_1^+ = f_{i1}. \tag{3.25}$$

This 2×2 system is solved immediately for all coefficients corresponding to *region A*. Next, we will consider the coefficients in *region B*. Assuming $2 \leq j \leq N_y - 2$ from (3.24), the i sums must range over only 0 and 1 (other terms in the sum are zero). These unknowns are then recoverable from (3.24) expressed as

$$u_{0j}\delta_0^- + u_{1j}\delta_1^- = f_{0,j+2}, \qquad u_{0j}\delta_0^+ + u_{1j}\delta_1^+ = f_{1,j+2}. \tag{3.26}$$

Now, let us consider the coefficients in *region C*. Assuming $i = 0, 1$ and $j = 0, 1$ from (3.24) when $i = 0, 1$, the unknown coefficients are recoverable from (3.24) expressed as:

$$u_{0j}\delta_0^- + u_{1j}\delta_1^- = f_{0,j+1} - \sum_{i=2}^{N_x} u_{ij}\delta_i^-, \tag{3.27}$$

$$u_{0j}\delta_0^+ + u_{1j}\delta_1^+ = f_{1,j+1} - \sum_{i=2}^{N_x} u_{ij}\delta_i^+. \tag{3.28}$$

Notice that the source terms on the right-hand side involve coefficients already recovered in *region A*. Lastly, let us consider the coefficients in *region D*. Assuming $i = 0, 1$ and $j = N_y - 1, N_y$ from (3.23), the unknown coefficients are recoverable from (3.23) expressed as:

$$u_{i,N_y-1}\delta_{N_y-1}^- + u_{i,N_y}\delta_{N_y}^- = f_{i0} - \sum_{j=0}^{N_y-2} u_{ij}\delta_j^-, \tag{3.29}$$

$$u_{i,N_y-1}\delta_{N_y-1}^+ + u_{i,N_y}\delta_{N_y}^+ = f_{i1} - \sum_{j=0}^{N_y-2} u_{ij}\delta_j^+. \tag{3.30}$$

Again, the source terms on the right-hand side involve already recovered coefficients. Recovery of the coefficients of each region costs at most $\mathcal{O}(n^{1/2})$. Thus, the system $\mathcal{V}^T\mathcal{U}\boldsymbol{y} = \boldsymbol{z}$ is solvable at $\mathcal{O}(n^{1/2})$ complexity. As a result, the matrix-vector multiply of $\widetilde{\mathcal{M}}^{-1}\boldsymbol{x}$ has a cost of $\mathcal{O}(n^{3/2})$.

Chapter 4

Numerical Experiments

This chapter presents a numerical experiment for solving the Helmholtz equation using a matrix laboratory, specifically `MATLAB R2020a`. In order to minimize the resource storage and poor operation counts, an iterative method coupled with a preconditioner is used to numerically solve the modal spectral approximation of the Helmholtz equation described in early chapters. A preconditioner is used to increase the rate of convergence and enhance the accuracy of the numerical solution. The following sections present a numerical investigation of the preconditioned Helmholtz problem.

For this section alone, all indexing is from 1 instead of 0 for the convenience of reporting the numerical results. The reader should note that the change of indexing does not affect the analysis of algorithmic complexities. The goal of this analysis is to empirically determine the complexity of our iterative scheme for numerically solving our Chebyshev approximations of the Helmholtz equation.

4.1 Analysis of the Helmholtz Equation

The presented modal Chebyshev spectral approximation of (2.33) and (2.34) is based on integration sparsification. The approximation of the Helmholtz equation is solved iteratively by way of a preconditioned GMRES, with full details given in Section 3.9. The source code `gmres.m` solves the linear system using the Generalized Minimal Residual method with restarts. The syntax employed is

$$\texttt{x} = \texttt{gmres}(\texttt{A}, \texttt{b}, \texttt{restart}, \texttt{tol}, \texttt{maxit}, \texttt{M}),$$

which specifies a preconditioner M, and computes x, the modal Chebyshev expansion coefficients, by effectively solving the preconditioned system. To determine how the number `iter` of GMRES iterations scales with resolution, a tolerance of `1e-13` was chosen as the tolerance for all GMRES solves. In order to numerically solve the Helmholtz equation, the solution and source employed are defined by the following equations:

$$u(x,y) = \cos(mx)e^{(ly)} \qquad \text{and} \qquad f(x,y) = (-m^2 + l^2 + \kappa^2)u(x,y), \tag{4.1}$$

where m and l describe the parameters for the manufactured solution and $+\kappa^2$ (the plus choice) is the parameter in the Helmholtz equation.

4.1.1 Accuracy Test

In previous chapters, the modal Chebyshev expansion coefficients corresponding to a numerical solution have been represented by (2.37). Since we are changing the index convention, these modal coefficients would now be represented as

$$\boldsymbol{u}(\alpha) = u_{ij}, \qquad \alpha = (i-1)N_y + j, \qquad 1 \le i \le N_x, \qquad 1 \le j \le N_y. \tag{4.2}$$

Regardless, our convention has been to use $\boldsymbol{u}$ and u_{ij} to represent the *modal* expansion coefficients. Therefore, we have to be careful with the notation for representing the *nodal* values of a Chebyshev numerical solution.

Let $u_c(x, y)$ represent the right-hand side of (2.35) and $u_{ref}(x, y)$ denote an exact reference solution. Now, let

$$(x_J, y_J) \qquad \text{for} \qquad 1 \leq J \leq J_{max} \tag{4.3}$$

be a uniform reference grid. Although a single index, J enumerates a two-dimensional array of points. Then,

$$u_c(x_J, y_J) \qquad \text{for} \qquad 1 \leq J \leq J_{max} \tag{4.4}$$

is a collection of nodal values stemming from the numerical solution. Likewise,

$$u_{ref}(x_J, y_J) \qquad \text{for} \qquad 1 \leq J \leq J_{max} \tag{4.5}$$

is a collection of exact reference nodal values. The relative errors are computed by evaluating the following:

$$\epsilon_{RL2} = \sqrt{\frac{\sum_{J=1}^{J_{max}} \left| u_c(x_J, y_J) - u_{ref}(x_J, y_J) \right|^2}{\sum_{J=1}^{J_{max}} \left| u_{ref}(x_J, y_J) \right|^2}}, \tag{4.6}$$

where ϵ_{RL2} represents the relative L_2-norm. The J_{max} is typically determined with 51 points in each direction.

The relative error is evaluated for a given GMRES tolerance of `1e-13` . We define thirteen truncation values incremented by two from 12 to 36 with parameter values defined as $m = 2.3$, $l = 1.5$, and $\kappa^2 = 0.777$. This is an initial test scenario which allows us to confirm accuracy and complexity, but further tests, especially, with varied κ^2, are needed. The evidence shows that the number of iterations needed to achieve a fixed tolerance of `1e-13` increases minimally with resolution. Moreover, Table 4.1 describes the accuracy comparisons of three different strategies for solving the Helmholtz equation: unpreconditioned, preconditioned, and Gaussian Elimination. Figure 4.1 shows the error convergence for the middle column (preconditioned system) of Table 4.1.

truncation		Unpreconditioneed	Preconditioned	Gaussian Elimination
N_x	N_y	ϵ_{RL2}	ϵ_{RL2}	ϵ_{RL2}
12	13	2.1927e-03	2.1927e-03	2.1927e-03
14	15	1.2133e-04	1.2133e-04	1.2133e-04
16	17	4.9628e-06	4.9628e-06	4.9628e-06
18	19	1.5668e-07	1.5668e-07	1.5668e-07
20	21	3.9289e-09	3.9290e-09	3.9290e-09
22	23	8.0396e-11	7.9655e-11	7.9656e-11
24	25	2.3024e-11	1.3348e-12	1.3319e-12
26	27	3.3216e-11	8.6240e-14	2.0533e-14
28	29	3.1769e-11	1.3032e-13	4.7119e-15
30	31	3.5794e-11	5.3718e-14	4.8168e-15
32	33	3.3285e-11	5.3525e-14	2.1181e-15
34	35	3.5484e-11	6.6973e-14	2.3330e-15
36	37	2.8854e-11	7.3607e-14	2.1975e-15

Table 4.1: Accuracy comparison of strategies for numerically solving the Helmholtz equation for thirteen prescribed truncations, along with a set tolerance of `1e-13`.

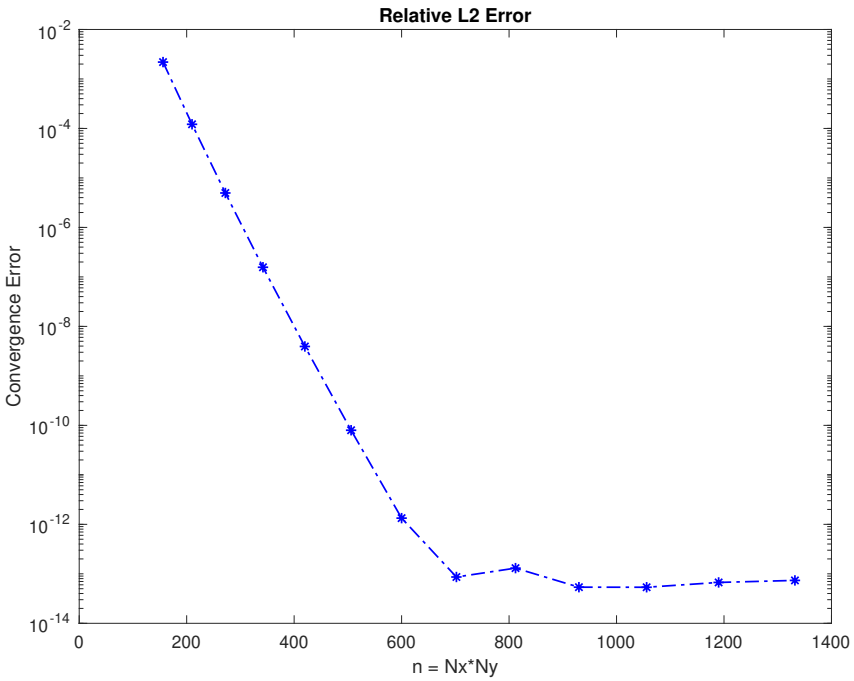

Figure 4.1: The plot above is plotted in `MATLAB R2020a` using the `semilogy` feature for thirteen prescribed truncations, along with a set tolerance of `1e-13`. The convergence error for the preconditioned system indicates exponential convergence with resolution.

4.1.2 Complexity Verification

As previously mentioned, a given tolerance of `1e-13` is investigated to observe the relation between the truncations and the number `iter` of GMRES iterations. Since the details of this relationship is unknown, let us assume a power-law dependency of the form

$$p = \alpha n^{\beta}, \tag{4.7}$$

where n represents the product of the unknowns in each Cartesian direction (this α is a power law constant and not an index). Let us linearize the above equation by taking the log (base-ten) of both sides,

$$\log(p) = \log(\alpha) + \beta \log(n). \tag{4.8}$$

Define $y \equiv \log(p)$ and $x \equiv \log(n)$. Substituting these into (4.8) and rearranging, we get

$$y = \beta x + \log(\alpha), \tag{4.9}$$

which is indeed a linear relationship. The slope of this straight line is the unknown exponent β and the value of the intercept is $\log(\alpha)$. Given that we are evaluating the relationship between the product of the truncations corresponding to the two coordinate directions and the number of iterations, let us find a linear polynomial that best fits the data using `MATLAB's` built-in `polyfit` routine. For a fixed tolerance of `1e-13` and truncation values incremented by two from 60 to 100, the first degree polynomial returned that best approximates the data is defined by:

$$y = 0.2069x + 0.9483. \tag{4.10}$$

Now, if we take the base-10 exponentiate of (4.10) then the relation of the power law is defined by:

$$p = 10^{0.9483} n^{0.2069}. \tag{4.11}$$

Thus, the value that best fits the data is 0.2069.

The number of required iterations empirically scales nearly as $\mathcal{O}(n^{1/5})$. Assuming this is true, the cumulative cost for k-iterations of GMRES breaks down to the following:

1. Arnoldi iteration: $\mathcal{O}(n^{7/5}) = \mathcal{O}(n^{1.4})$.
2. Matrix-vector Multiply (with preconditioner): $\mathcal{O}(n^{(1/5+3/2)}) = \mathcal{O}(n^{17/10}) = \mathcal{O}(n^{1.7})$.
3. Least Squares: $\mathcal{O}(n^{2/5}) = \mathcal{O}(n^{0.4})$.

As a result, the cumulative cost scales like $\mathcal{O}(n^{1.7})$. The linear behavior observed in Figure 4.2, illustrates that the number of required iterations scales like a power-law with respect to the number of truncations. Overall, the method appears to have a sub-quadratic solve complexity.

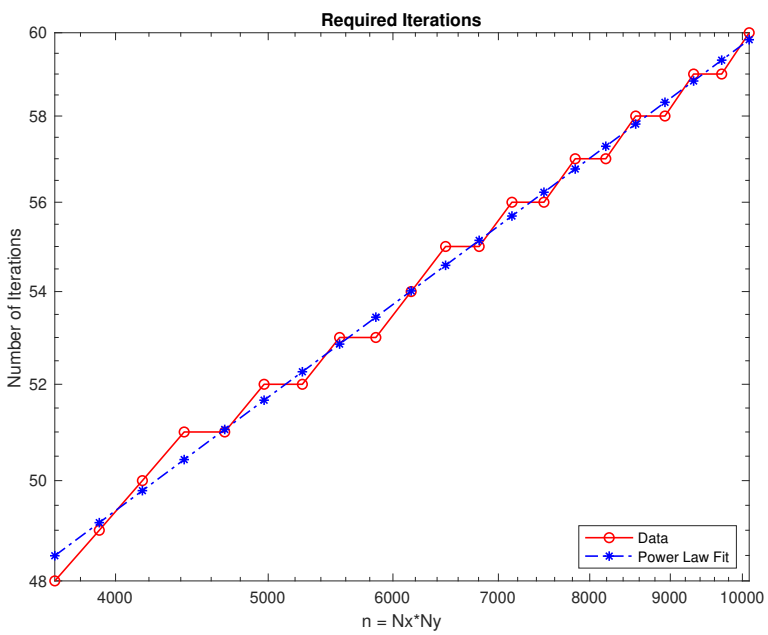

Figure 4.2: The plot above is plotted in `MATLAB R2020a` using the `Loglog` feature for twenty-one prescribed truncation values, along with a set tolerance of `1e-13`. The linear behavior illustrates a power law relationship between the truncations and required iterations when a GMRES preconditioner is used.

Similarly, the power-law that empirically describes the relationship between the truncations and total time is the following:

$$p = 10^{-6.3337} n^{1.5962}. \tag{4.12}$$

The value that best fits the data is 1.5962. Moreover, since the total time scales like $\mathcal{O}(n^{1.6})$, the cumulative cost for k-iterations of GMRES, also scales like $\mathcal{O}(n^{1.6})$. This observed scaling is a bit better than the $\mathcal{O}(n^{1.7})$ prediction made on the last page. Again, the linear behavior observed in Figure 4.2 illustrates that the total time scales like a power law with respect to the number of truncations. For the sake of simplicity, only comparison plots for the number of required iterations and total time for the preconditioned system were generated for the prescribed truncation values.

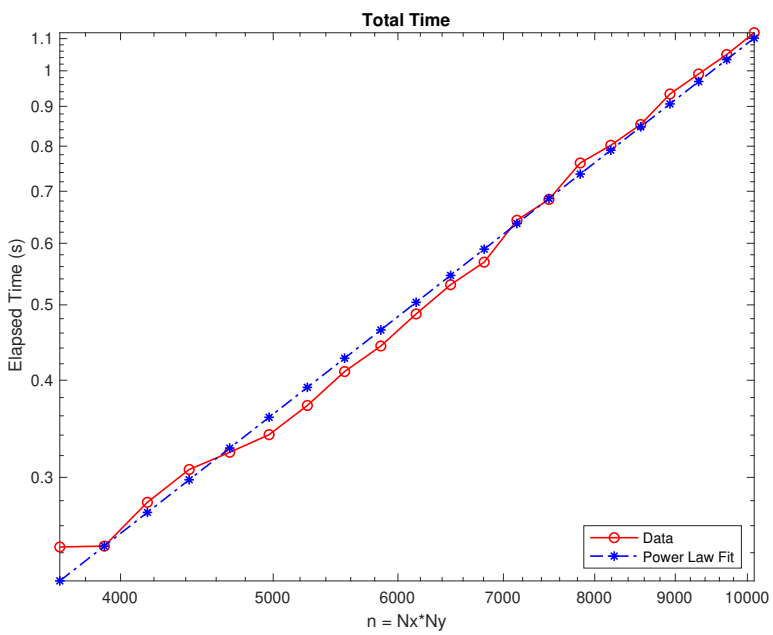

Figure 4.3: The plot above is plotted in `MATLAB R2020a` using the `Loglog` feature for twenty-one prescribed truncation values, along with a set tolerance of `1e-13`. The linear behavior illustrates a power law relationship between the truncations and total time required to arrive at a solution when a GMRES preconditioner is used.

truncation		Unpreconditioned		Preconditioned		Gaussian Elimination	
N_x	N_y	ϵ_{RL2}	timing	ϵ_{RL2}	timing	ϵ_{RL2}	timing
60	61	1.3798e-10	3.6532e+00	1.4450e-13	3.2672e-01	3.5373e-15	3.4429e-01
70	71	1.0212e-10	7.9053e+00	1.1181e-13	4.5317e-01	4.7278e-15	8.6248e-01
80	81	1.8784e-10	1.1400e+01	9.3577e-14	6.6254e-01	3.2036e-15	1.8067e+00
90	91	1.3202e-10	1.5416e+01	1.4966e-13	1.0009e+00	2.8595e-15	3.1349e+00
100	101	3.0402e-10	2.0849e+01	1.5725e-13	1.3864e+00	2.1859e-15	5.8221e+00
110	111	2.6503e-10	2.6033e+01	1.4899e-13	1.8604e+00	4.1492e-15	1.0380e+01
120	121	3.3494e-10	3.2508e+01	1.7620e-13	2.4987e+00	2.6592e-15	1.6859e+01
130	131	3.1014e-10	4.2055e+01	1.0784e-13	3.1951e+00	4.0950e-15	2.9219e+01
140	141	3.8236e-10	4.8768e+01	1.4475e-13	4.0162e+00	2.3107e-15	4.2288e+01
150	151	3.3007e-10	5.9189e+01	1.1417e-13	5.0592e+00	4.7963e-15	9.1595e+01

Table 4.2: Testing results for solving (2.33). Each timing in this table corresponds to an average over 10 runs.

Table 4.2 lists three testing result comparisons of solving the Helmholtz equation via unpreconditioned GMRES, preconditioned GMRES, and Gaussian elimination for ten prescribed truncation values, along with a set tolerance of `1e-13`. The table indicates the effectiveness of implementing a preconditioner for the spectral solution of this PDE. For small truncation values, Gaussian elimination outperforms both the unpreconditioned and preconditioned methods. However, as seen in Table 4.2, a preconditioner is useful for effectively solving the Helmholtz equation. Additionally, to investigate the solve complexity of each method, a set tolerance of `1e-13` was used for all GMRES solves. For the truncations considered, the preconditioned method empirically yields a linear set-up cost followed by a sub-quadratic solve complexity of $\mathcal{O}(n^{1.6})$. Moreover, the number of iterations needed to achieve a fixed tolerance of `1e-13` scales dependently with resolution nearly as $\mathcal{O}(n^{1/5})$.

Chapter 5

Conclusion

For this work, we presented a method for solving the two-dimensional Helmholtz problem posed on a rectangular domain subject to Dirichlet boundary conditions based on integration sparsification with further additional (genuine) preconditioning. Our approach entailed using a spectral approximation of the Helmholtz problem involving a modal integration matrix for Chebyshev polynomials. The total number of variables for the spectral approximation of the preconditioned system is $n = (N+1)^2$, where $N+1$ is the number of Chebyshev modes associated with both Cartesian direction. Implementation of a preconditioner to solve the spectral solution of the aforementioned problem empirically demonstrated a linear set-up cost followed by a sub-quadratic solve complexity of $\mathcal{O}(n^{1.6})$. For the truncations considered, our approach demonstrated spectral accuracy and is empirically well-conditioned. For small truncation values, we found that Gaussian elimination outperformed both the unpreconditioned and preconditioned systems. However, as truncation values increased, we found that implementation of a preconditioner was advantageous for effectively solving the Helmholtz problem. Thus, the described method is indeed effective for the presented modal Chebyshev approximation of the Helmholtz equation.

Potential future avenues of research include extension to harder problems with Neumann

or mixed Dirichlet-Neumann, more complicated domains, as well as extension to higher spatial dimensions and other PDEs. Firstly, the change of boundary conditions entails the replacement of Dirichlet vectors with either Neumann vectors or Dirichlet-Neumann vectors. For example, for Neumann boundary conditions this would involve replacing Dirichlet vectors

$$\delta^{\pm} = [T_0(\pm 1), T_1(\pm 1), T_2(\pm 1), T_3(\pm 1), T_4(\pm 1), ...] = [1, \pm 1, \pm 1, \pm 1, ...]$$

with Neumann vectors

$$\nu^{\pm} = [T_0'(\pm 1), T_1'(\pm 1), T_2'(\pm 1), T_3'(\pm 1), T_4'(\pm 1), ...] = [0, 1, \pm 4, 9, \pm 16, ...].$$

In this case, the problem with pure Neumann boundary conditions is indeed singular.

Sommerfeld conditions are mixed Dirichlet-Neumann conditions. In the context here they are $(i\kappa u + \partial u/\partial n)|_{\partial\Omega} = h$, where $\partial/\partial n$ denotes differentiation normal to the boundary. Clearly, such conditions will be approximated by Dirichlet and Neumann vectors. But these boundary conditions also lead to a complex solution, with the boundary conditions coupling the real and imaginary parts. These issues are beyond the scope of this thesis, but one might try to extend our work to this setting. Sommerfeld conditions (radiation conditions) are often used when studying the wave equation, or its time-harmonic version, the Helmholtz equation.

A possibility for treating more complicated domains is shown in Figure 5.1 which depicts an irregular domain with a curved boundary embedded inside a rectangle. The PDE can be solved on the full rectangle, but with boundary conditions enforced on the curved boundary. The tau methods described in this thesis allow for such a formulation. The figure below also suggests a straightforward way of partitioning the boundary which could be used with the tau approach. With further work, some aspects of this thesis may generalize to this more complicated scenario.

As a different line of research, a plausible method for direct inversion of the coefficient matrix stems from a direct method for solving the coefficient matrix through a Woodbury identity matrix. Thus, viewing

$$\mathcal{M} = \widetilde{\mathcal{M}} + \widetilde{\mathcal{U}}\widetilde{\mathcal{V}}^T,$$

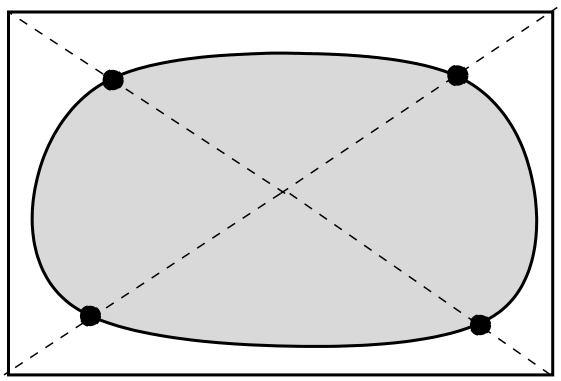

Figure 5.1: Irregular domain (shaded) with a curved boundary embedded inside a rectangle.

where $\mathcal{M}$ is the coefficient matrix and $\widetilde{\mathcal{M}}$ is the preconditioner. The perturbation $\widetilde{\mathcal{U}}\widetilde{\mathcal{V}}^T = \mathcal{B} - \widetilde{\mathcal{B}}$ in terms of (2.39) and (3.10). Here we can use the Woodbury formula to invert $\mathcal{M}$ (inversion of $\widetilde{\mathcal{M}}$ is studied in this thesis). The formula here uses the $\widetilde{\mathcal{U}}$ and $\widetilde{\mathcal{V}}^T$ from above. By the Woodbury matrix identity

$$\mathcal{M}^{-1} = \widetilde{\mathcal{M}}^{-1} - \widetilde{\mathcal{M}}^{-1}\widetilde{\mathcal{U}}\mathcal{C}^{-1}\widetilde{\mathcal{V}}^T\widetilde{\mathcal{M}}^{-1}, \tag{5.1}$$

where the *capacitance matrix* is

$$\mathcal{C} = I + \widetilde{\mathcal{V}}^T\widetilde{\mathcal{M}}^{-1}\widetilde{\mathcal{U}}. \tag{5.2}$$

Here, the capacitance matrix is $m \times m$, where m is given after (2.39). Finally, for higher dimensions, the ideas presented in this thesis can almost certainly be carried out for any n. However, for $n > 2$ the complexity for the strategy (preconditioned GMRES) presented is likely quadratic or worse. Lastly, other linear PDEs, for example those with Laplacian principal part and first derivative terms should be considered.

Bibliography

[1] Michele Benzi, *Preconditioning techniques for large linear systems: A survey*, Journal of Computational Physics 182, 418–477 (2002).

[2] John P. Boyd, *Chebyshev and Fourier spectral methods*, Courier Corporation, 2001.

[3] J. S. Hesthaven E. A. Coutsias, T. Hagstrom and D. Torres, *Integration preconditioners for differential operators in spectral-methods*, Proceedings of the Third International Conference on Spectral and High Order Methods (1996).

[4] Carl T. Kelley, *Iterative methods for linear and nonlinear equations*, Society for Industrial and Applied Mathematics, 1995.

[5] Balaram Khatri Ghimire, *Hybrid Chebyshev polynomial scheme for the numerical solution of partial differential equations*, The University of Southern Mississippi, 2016.

[6] Stephen R. Lau, *Direct, low-memory, spectral solution of harmonic problems on a block at near optimal complexity*, in preparation. (2022).

[7] Sahuck Oh, *An efficient spectral method to solve multi-dimensional linear partial differential equations using Chebyshev polynomials*, Mathematics **7.1** (2019).

[8] Per-Gunnar Martinsson Sijia Hao, *A direct solver for elliptic pdes in three dimensions based on hierarchical merging of Poincaré–Steklov operators*, Journal of Computational and Applied Mathematics (2016).

[9] Kaick-O.V. Dyer R. Zhang, H., *Spectral mesh processing*, Proceedings of Eurographics State-of-the-art Report. (2007).