Absorbing Boundary Conditions and Numerical Methods for the Linearized Water Wave Equation in 1 and 2 Dimensions

by

David K. Prigge

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Applied and Interdisciplinary Mathematics)
in The University of Michigan
2016

Doctoral Committee:

Professor Smadar Karni, Co-Chair
Professor Rémi Abgrall, University of Zurich, Co-Chair
Professor Robert Krasny
Professor Emeritus Jeffrey B Rauch
Professor Philip L Roe

To my mother and father

ACKNOWLEDGEMENTS

I would like to thank the Mathematics Department at the University of Michigan for giving me the opportunity to study applied math here. In particular, I would like to thank Peter Miller for all of his help in my early years here at Michigan. I would also like to thank the staff in the graduate student office, especially Stephanie Carroll and Tara McQueen, for always being there for me when I had a question or needed to talk to someone. I want to thank my undergraduate advisor, Jeffery DiFranco, for encouraging me to apply to Michigan.

I want to thank my advisor, Smadar Karni, for all of her help throughout my time here. My mathematical training and interests were greatly influenced by her and I am glad that I was able to work with her. Thank you for taking the time to develop me into the mathematician I am today. I also want to thank my co-advisor, Rémi Abgrall, for helping me during my time in Zürich and for all your help on my thesis problem. I also want to thank Sean Carney for all of his work on the uniform and nonuniform grid formulations for the One Way Water Wave Equation, as well as his support during Summer 2014 at the SIAM Annual Conference. His help made the work on the 1-D problem much more complete.

I am grateful to Jeffrey Rauch for all his work on this project. The numerous talks we had were always helpful at the time and would often lead to me discovering something new about this problem weeks, sometimes even months, later. I am thankful to the rest of my committee, Robert Krasny and Phil Roe, for their

help and comments on my work. I also want to thank Karen Smith for her kindness throughout the years.

My time here in Ann Arbor would not have been nearly as enjoyable as it was without my friends. The memories we created will stay with me a lifetime. I especially want to thank the following people (note: not an exhaustive list) for all the talks, mathematical and personal, we have had throughout the years: Chris Fraser, Ivy Wei, Dan Hathaway, Will Drobny, Ari Shnidman, Becky Hoai, Alex Mueller, Pedro Acosta, Adam Kaye, Dan DeWoskin, Joe Roberts, Jen Beichman, Kurt Mishchenko, Zach Scherr, Wendy Shang, Brandon Carter, Jeremy Hoskins, Daniel Barter. I especially want to thank Chris for being my roommate for four years and still being my friend through all of that! I want to thank my friends in Seattle for all the support and fun we have had for almost a decade now: Tymon Silva, Vince Canlas, Jason Foster.

I want to thank my family for always supporting me and believing in me. My mom and dad always encouraged me to be me and I can't state enough how much that has meant to be as I have grown older. My sister Rachel was always there to help me with schoolwork and my brother Adam was always there when I wanted to play as a kid.

Most importantly, I want to thank my wife Olivia Martin for all of her love and support. Without her, I would not be the person I am today and I am thankful for all the great times we have had together. I am excited to see what the future has in store for us!

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Absorbing Boundary Conditions and Numerical Methods for the Linearized Water Wave Equation in 1 and 2 Dimensions

by
David Prigge

Co-Chair: Professor Smadar Karni

Co-Chair: Professor Rémi Abgrall, University of Zurich

The linearized water wave equation (WWE) models incompressible, irrotational, inviscid free surface flows in deep water. We will investigate the WWE in both one and two spatial dimensions and derive nonreflecting boundary conditions for both. We will calculate numerical solutions for a fractional PDE arising as a nonreflecting boundary condition to the 1-D and 2-D WWE and discuss convergence and stability of the numerical methods. The nonreflecting boundary conditions will be implemented in a boundary layer around the computational domain.

CHAPTER I

Introduction

When modeling water waves, it is common to be interested in their behavior in unbounded domains. This can be illustrated, for example, by modeling the surface of the ocean far away from any body of land. Though the boundaries are not infinitely far away, the effects from them are insignificant. Ideally, there would be a way to model the water waves in the entire unbounded domain but computational limitations will require that the domain be truncated to an arbitrary finite size, introducing an artificial boundary. It may be assumed that no waves originate outside this restricted domain. Boundary conditions that simulate the behavior of the water waves on the unbounded domain and make the boundary undetectable to outgoing waves would be preferred. If the boundary conditions are not properly prescribed, then reflections of outgoing waves off the artificial boundary towards the interior of the domain can occur. In general, there should be reflections, even if they are delayed.

Among the ways to impose these "absorbing" or "non-reflecting" boundary conditions are to apply them directly on the boundary or to apply them in a matched layer outside of the computational domain, where they will be damped before they reach the end of the layer. We provide a brief introduction to each technique and give a few examples of their use in different types of problems.

1.1 Absorbing boundary conditions

In 1977, Engquist and Majda developed perfectly absorbing boundary conditions for the 2-D wave equation [6]. Note that the wave equation is non-dispersive and has constant wave propagation speed. The water wave equations that will be analyzed in Chapters II and III are dispersive and have waves with propagation speed ranging across all real numbers.

They consider solutions to the 2-D wave equation

$$\frac{\partial^2}{\partial t^2}u(x,y,t) = \frac{\partial^2}{\partial x^2}u(x,y,t) + \frac{\partial^2}{\partial y^2}u(x,y,t) \tag{1.1}$$

in the half-space $x \geq 0$. Consider solutions of the form $w(x,y,t) = e^{i(\xi_1 x + \xi_2 y + \omega t)}$. Without loss of generality, assume $\omega < 0$. For these types of solutions, the dispersion relation is

$$(i\omega)^2 = (i\xi_1)^2 + (i\xi_2)^2 \Longrightarrow \omega^2 = \xi_1^2 + \xi_2^2. \tag{1.2}$$

The goal is to derive absorbing boundary conditions for waves propagating in the negative x-direction ($\xi_1 < 0$).

Assuming $\omega^2 - \xi_2^2 > 0$,

$$\left(\frac{du}{dx} - i\sqrt{\omega^2 - \xi_2^2}u\right)\Bigg|_{x=0} = 0 \tag{1.3}$$

is a boundary condition that annihilates these left-moving waves. (1.3) is an exact absorbing boundary condition for (1.1). However, it is nonlocal in space and time, making its use impractical from a computational aspect.

To prevent having to use a boundary condition that is nonlocal in time, look at the dispersion relation for (1.3):

$$i\xi_1 - i\sqrt{\omega^2 - \xi_2^2} = 0. \tag{1.4}$$

To create local in time boundary conditions, we approximate (1.4) at normal incidence to the x-boundary ($\xi_2 = 0$).

Using a Taylor expansion for $\sqrt{\omega^2 - \xi_2^2}$ when $\xi_2 = 0$, we have the dispersion relation for the first absorbing boundary condition:

$$i\xi_1 - i\omega = 0. \tag{1.5}$$

In physical space, this corresponds to

$$\frac{\partial u}{\partial x} - \frac{\partial u}{\partial t} = 0. \tag{1.6}$$

Taking a higher order Taylor expansion for $\sqrt{\omega^2 - \xi_2^2}$ when $\xi_2 = 0$, we have the dispersion relation for the second absorbing boundary condition:

$$i\xi_1 - i\left(\omega - \frac{\xi_2^2}{2\omega}\right) = 0. \tag{1.7}$$

Multiplying (1.7) by $i\omega$, we get

$$(i\xi_1)(i\omega) - (i\omega)^2 + \frac{(i\xi_2)^2}{2} = 0. \tag{1.8}$$

In physical space, this corresponds to

$$\frac{\partial^2 u}{\partial t \partial x} - \frac{\partial^2 u}{\partial t^2} + \frac{1}{2}\frac{\partial^2 u}{\partial y^2} = 0. \tag{1.9}$$

Numerical experiments in [6] show that for waves with a 45° angle of incidence, (1.6) reflects approximately 17% of the amplitude of the incident wave and (1.9) reflects approximately 3% of the amplitude of the incident wave. To get more absorption on the boundary, higher order approximations of $\sqrt{\omega^2 - \xi_2^2}$, when ξ_2 is near zero, are taken. It is shown in [6] that a higher order Taylor expansion leads to an ill-posed problem, so a Padé approximation of $\sqrt{\omega^2 - \xi_2^2}$ around $\xi_2 = 0$ is used to get a third absorbing boundary condition:

$$\frac{\partial^3 u}{\partial t^2 \partial x} - \frac{\partial^3 u}{\partial t^3} - \frac{1}{4}\frac{\partial^3 u}{\partial x \partial y^2} + \frac{3}{4}\frac{\partial^3 u}{\partial t \partial y^2} = 0. \tag{1.10}$$

Other types of approximations, such as Chebyshev, can also be used to create absorbing boundary conditions. Given that a computer has finite memory, it is not reasonable to use the nonlocal in time boundary conditions. We note that [8] implemented a nonlocal in time nonreflecting boundary condition for time-dependent wave problems.

For more papers that use approximations of perfect absorbing boundary conditions to absorb outgoing waves, see [1, 7, 10].

1.2 Matched layers

Another technique to reduce reflections of outgoing waves at the boundary of the domain is to surround the computational domain with a layer designed to artificially damp or decay the outgoing waves. The resulting outgoing waves are ideally completely damped before they reach the end of the computational domain. For example, [14] introduces a layer near the boundary to provide viscous damping to outgoing waves for hyperbolic problems. However, because matched layers rely on artificial damping and decay to reduce the size of the outgoing waves before they reach the boundary, there is a possibility that there will be reflections when the outgoing waves do reach the boundary or the layer interface. This can influence the computation in the interior of the domain. Moreover, if the interface between the layer and the computational domain is not properly treated, then this approach can lead to reflections.

1.2.1 Perfectly matched layers

In [2], Berenger introduces a new technique for deriving an absorbing boundary layer called a perfectly matched layer (PML). The PML formulations are derived by computing the solution for a wave incident on the interface between the layer and

the domain at an arbitrary angle. From here, one solves for conditions to ensure the outgoing waves are not reflected off this interface. The benefit from deriving a PML is that when a wave enters the absorbing layer, it decays exponentially. If there are outgoing waves that reflect off the boundary, they are exponentially small. The PML formulation is derived to specifically make sure there are no reflections off the interface between domain and layer, which is a problem when just matching an absorbing boundary layer.

1.3 Overview

The focal point of this thesis is to derive and numerically implement equations for use in an absorbing boundary layer for the Linearized Water Wave Equations (WWE) in one and two space dimensions. These equations model inviscid, incompressible, irrotational fluid flow in two and three space dimensions by solving for the velocity on the surface of the infinitely deep fluid. The remainder of this dissertation is organized as follows.

In Chapter II, we give a brief derivation of the 1-D WWE. We also derive the one-way water wave equations (OWWWE), an absorbing boundary condition for the 1-D WWE that supports one-way wave propagation. Properties, numerical methods, and numerical results for the OWWWE are discussed. We also implement the OWWWE as a boundary layer for the 1-D WWE and show its effectiveness. The goal here is to gain understanding of how the absorbing boundary conditions are derived for the 1-D WWE and use this insight for deriving absorbing boundary conditions for the 2-D WWE.

In Chapter III, we give some details of the derivation of the 2-D WWE. We state some properties of the 2-D WWE and we discuss the numerical method used to solve

the 2-D WWE and the reflections that arise from the boundary of our domain when there are not absorbing boundary conditions implemented.

In Chapter IV, we derive a hierarchy of equations for use in an absorbing boundary layer for the 2-D WWE based on the ideas in [6]. We discuss numerical implementations of the equations.

In Chapter V, we discuss some further directions of study.

CHAPTER II

The 1-D Linearized Water Wave Equations and its Absorbing Boundary Conditions

We first provide an analysis of the 1-D WWE and the derivation of its absorbing boundary conditions. For further discussion and analysis, see [11, 12, 13].

2.1 Definitions and Notation

We define the Fourier transform in n spatial dimensions as

$$\hat{f}(\vec{\xi}) = \int_{\mathbb{R}^n} f(\vec{x}) e^{-i\vec{x}\cdot\vec{\xi}}\, d\vec{x} \tag{2.1}$$

and the inverse Fourier transform in n spatial dimensions as

$$f(\vec{x}) = \frac{1}{(2\pi)^n} \int_{\mathbb{R}^n} \hat{f}(\vec{\xi}) e^{i\vec{x}\cdot\vec{\xi}}\, d\vec{\xi}. \tag{2.2}$$

For this chapter, we take $n = 1$.

Let $H : L^2(\mathbb{R}) \to L^2(\mathbb{R})$, the Hilbert transform, be defined as

$$Hf(x) = \frac{1}{\pi} \text{P.V.} \int_{-\infty}^{\infty} \frac{f(y)\, dy}{x - y}. \tag{2.3}$$

Note that the Hilbert transform has the following Fourier symbol

$$\widehat{Hf}(\xi) = -i \operatorname{sgn}(\xi) \hat{f}(\xi). \tag{2.4}$$

$|D|$ denotes the operator defined by

$$|D|u(x) = \frac{1}{\pi} \frac{\partial}{\partial x} \text{P.V.} \int_{-\infty}^{\infty} \frac{f(y)\, dy}{x - y}. \tag{2.5}$$

Note that $|D|$ has the following Fourier symbol

$$\widehat{|D|f}(\xi) = |\xi|\hat{f}(\xi). \tag{2.6}$$

2.2 The 1-D Linearized Water Wave Equations

To derive the 1-D WWE, the derivation in [18, 16] by Wu is followed. Consider the motion of the interface separating an incompressible, irrotational, inviscid fluid in deep water from a region of zero density in two-dimensional space. Assume the surface tension on the interface is zero, the density of the fluid is one, and the gravitational field is $(0, -1)$. For time $t \geq 0$, denote the free interface as $\Gamma(t)$ and the region the fluid occupies, below the interface, as $\Omega(t)$. The motion of the fluid is given by the 2-D incompressible, irrotational, inviscid Euler equations

$$\vec{v}_t + (\vec{v} \cdot \nabla)\vec{v} = (0, -1) - \nabla p \qquad x \in \Omega(t), t \geq 0 \tag{2.7}$$

$$\nabla \cdot \vec{v} = 0 \qquad x \in \Omega(t), t \geq 0 \tag{2.8}$$

$$\nabla \times \vec{v} = 0 \qquad x \in \Omega(t), t \geq 0 \tag{2.9}$$

where $\vec{v} = (v_1, v_2)$ is the fluid velocity and p is the fluid pressure. The boundary conditions are

$$\begin{aligned} &p \text{ is constant on } \Gamma(t) \\ &(1, \vec{v}) \text{ is tangent to the free surface } (t, \Gamma(t)) \end{aligned} \tag{2.10}$$

We are interested in solutions to (2.7)–(2.10) such that for every fixed $t \geq 0$, $\Gamma(t)$ approaches the x-axis at infinity and $|\vec{v}(x, y, t)| \to 0$ as $|(x, y)| \to \infty$. In [18, 16], Wu shows the existence and uniqueness of solutions to the above system (2.7)–(2.10) for any interface $\Gamma(t)$ that is non self-intersecting.

2.2.1 Deriving the 1-D Water Wave Equation (WWE)

Writing (2.7)–(2.10) in terms of a Lagrangian variable α, the interface $\Gamma(t)$ can be described in terms of a curve in the complex plane $z(\alpha, t) = x(\alpha, t) + iy(\alpha, t)$, for $-\infty < \alpha < \infty$. The velocity of the fluid along the surface $\Gamma(t)$ can be written as

$$z_t(\alpha, t) = \vec{v}(z(\alpha, t), t) = v_1(\alpha, t) + iv_2(\alpha, t). \tag{2.11}$$

Though we are solving for the velocity of the fluid on the interface $\Gamma(t)$, the velocity of the fluid in $\Omega(t)$ can be recovered by solving Laplace's equation

$$\begin{aligned} \Delta\vec{v} &= 0 \qquad \text{in } \Omega(t) \\ \vec{v} &= z_t \qquad \text{on } \Gamma(t) \end{aligned}$$

Next, the fluid domain $\Omega(t)$ is mapped into the lower half plane. Because $\Gamma(t)$ is non self-intersecting, it divides the complex plane into two simply connected regions: an upper region of zero density and a lower region, $\Omega(t)$, in which the fluid lies.

This mapping into the lower half plane allows (2.7)–(2.10) to be written as a quasilinear system that may be reduced to the WWE. See [16] or [18] for the full quasilinear system.

The quasilinear system can be written in the form of the WWE

$$u_{tt} + |D|u = f(u, x, t) \tag{2.12}$$

where $f(u, x, t)$ is a function that is at least quadratic in u, the horizontal velocity on the free surface. Because we are interested in the solution near the boundary, it makes sense to linearize (2.12) around the zero solution to get

$$u_{tt} + |D|u = 0. \tag{2.13}$$

Recall that $|D|$ is the operator defined by (2.5) with Fourier symbol (2.6). For convenience, we re-label the Lagrangian variable α as x, keeping in mind it is related, but not identical, to the x-coordinate. In physical space, (2.13) is expressed as

$$\frac{\partial^2}{\partial t^2}u(x,y,t) + \frac{1}{\pi}\frac{\partial}{\partial x}P.V.\int_{-\infty}^{\infty}\frac{u(y,t)\,dy}{x-y} = 0. \tag{2.14}$$

Plane wave solutions to (2.13) of the form $\psi(x,t) = e^{i(\xi x - \omega t)}$ satisfy the dispersion relation

$$(-i\omega)^2 + |\xi| = 0 \rightarrow \omega^2 = |\xi| \tag{2.15}$$

which is plotted in Figure 2.1.

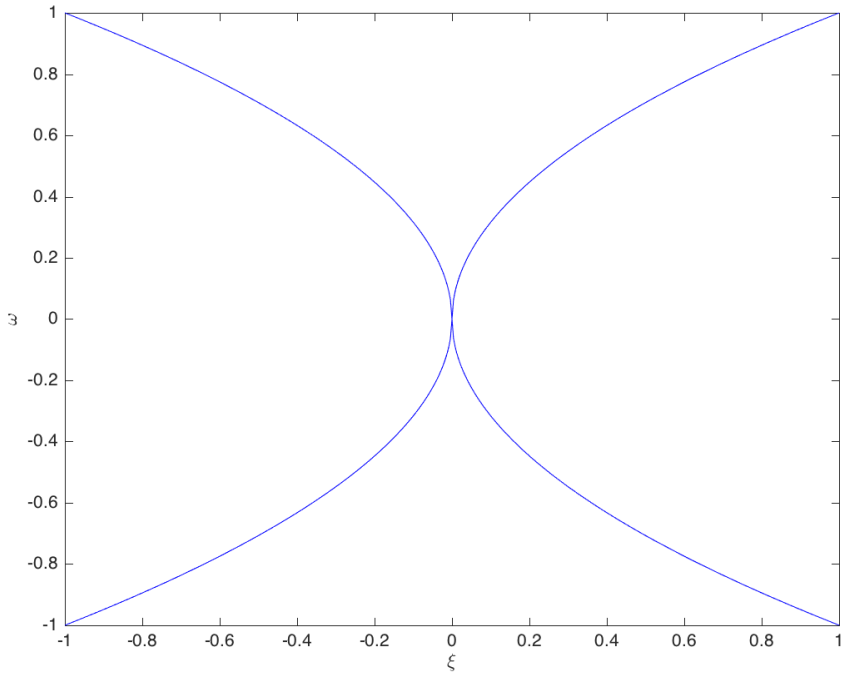

Figure 2.1: Dispersion relation, $\omega^2 = |\xi|$, for 1-D WWE

Differentiating (2.15), the group velocity, $\omega'(\xi)$, satisfies

$$2\omega\omega'(\xi) = \mathrm{sgn}(\xi). \tag{2.16}$$

The group velocity $\omega'(\xi)$ admits velocities of both signs, which means (2.13) supports the propagation of both left and right moving waves. Because the group velocity is not constant, (2.13) is dispersive, meaning waves corresponding to different wave numbers will propagate at different speeds. In this case, waves corresponding to small wave numbers will propagate infinitely fast as $\xi \to 0$.

If (2.13) is solved without properly prescribing absorbing boundary conditions, there will be numerical reflections, see Figure 2.2.

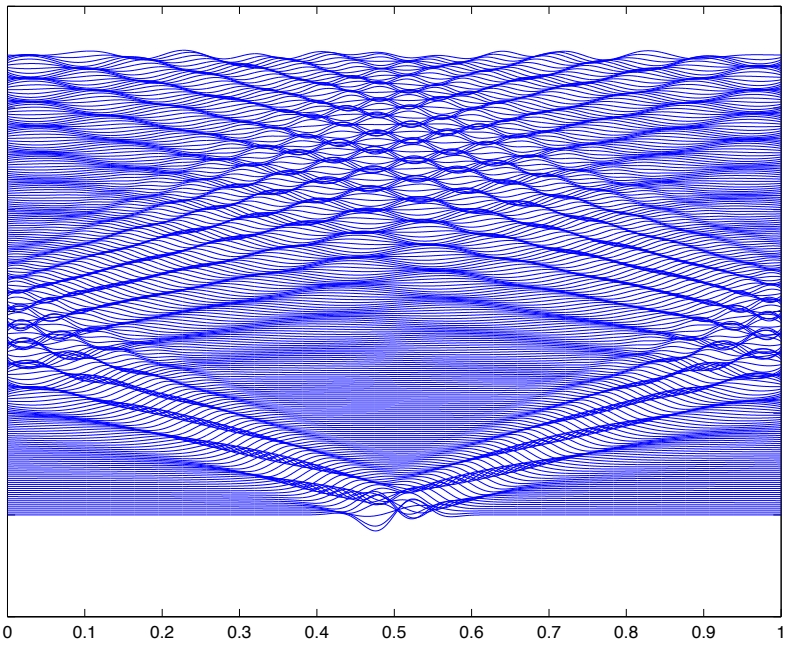

Figure 2.2: Solution to 1-D WWE for $x \in [0, 1]$ without absorbing boundary conditions. Time is on the vertical axis for $t \in [0, 20]$.

2.2.2 Deriving the One-Way Water Wave Equation

To prevent outgoing waves from reflecting off the boundary, we must derive absorbing boundary conditions. As motivation, we consider the 1-D wave equation

$$w_{tt} - w_{xx} = 0 \tag{2.17}$$

and look for absorbing boundary conditions. The goal is to gain insight as to how to proceed in deriving absorbing boundary conditions for the 1-D WWE.

Note that (2.17) can be factored into a product of left and right moving wave operators

$$w_{tt} - w_{xx} = (\partial_t + \partial_x)(\partial_t - \partial_x)w = 0.$$

We seek a similar factorization of (2.13). We write the desired factorization as

$$u_{tt} + |D|u = (\partial_t - \Phi_1)\,(\partial_t + \Phi_2)\,u = 0. \tag{2.18}$$

To get each factor of (2.18), recall the 1-D WWE's dispersion relation (2.15).

If we take the square root of (2.15), we get

$$\omega = \pm\sqrt{|\xi|} \tag{2.19}$$

which is plotted in Figure 2.3.

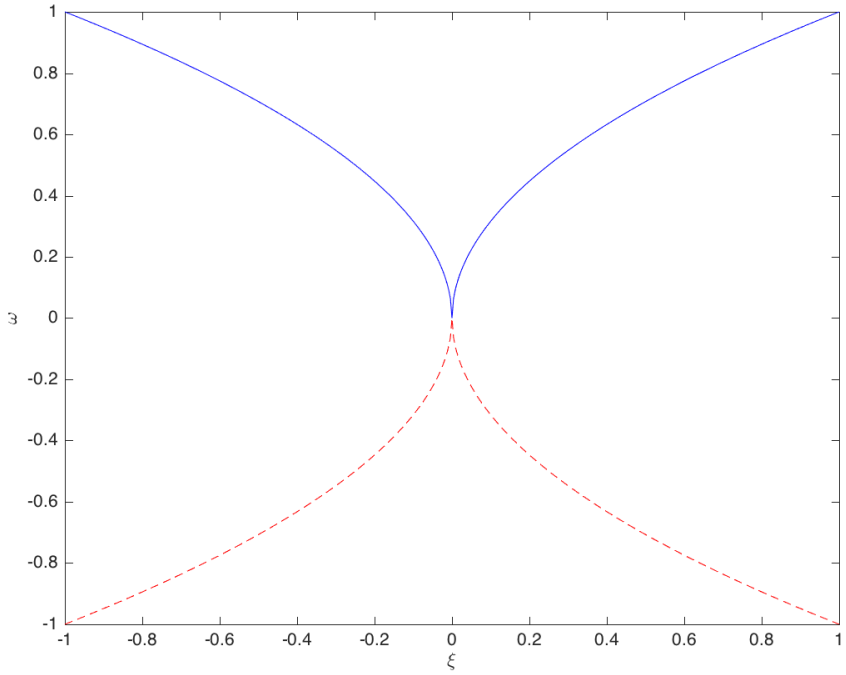

Figure 2.3: Dispersion relation (2.19). $\omega = \sqrt{|\xi|}$ in blue (solid), $\omega = -\sqrt{|\xi|}$ in red (dashed)

This particular solution for ω yields group velocities of

$$\omega'(\xi) = \pm\frac{\operatorname{sgn}(\xi)}{2\sqrt{|\xi|}} \tag{2.20}$$

which still support wave propagation in both positive and negative x directions.

A specific choice of the square root of (2.15) is taken to ensure that there are group velocities, $\omega'(\xi)$, that support one way wave propagation.

$$\omega(\xi) = \pm \text{sgn}(\xi)\sqrt{|\xi|}. \tag{2.21}$$

See Figure 2.4 for a plot of (2.21).

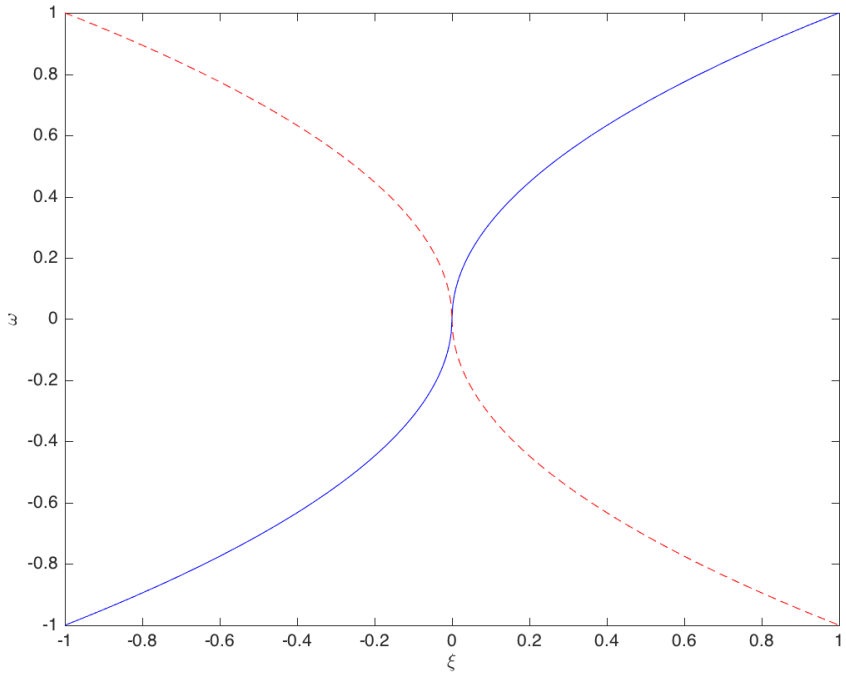

Figure 2.4: Dispersion relation (2.21). $\omega = \text{sgn}(\xi)\sqrt{|\xi|}$ in blue (solid), $\omega = -\text{sgn}(\xi)\sqrt{|\xi|}$ in red (dashed)

This particular solution for ω yields group velocities of

$$\omega'(\xi) = \pm\frac{1}{2\sqrt{|\xi|}} \tag{2.22}$$

which supports one way wave propagation, making this the correct choice for factorization in (2.18).

Consider the right moving waves, the plus sign in (2.21). Multiplying (2.21) by i,

we have

$$i\omega = i\,\mathrm{sgn}(\xi)\sqrt{|\xi|} \Longrightarrow -i\omega + \frac{i\xi}{\sqrt{|\xi|}} = 0. \tag{2.23}$$

Recall the Hilbert transform H has Fourier symbol (2.4). We note the operators $|D|^{\pm 1/2}$ have Fourier symbols $|\xi|^{\pm 1/2}$, respectively, and ∂_x has Fourier symbol $i\xi$. This turns (2.18) into

$$(\partial_t - H|D|^{1/2})(\partial_t + H|D|^{1/2})u = 0. \tag{2.24}$$

Note $H|D|^{1/2}$ has the same Fourier symbol as $-\partial_x|D|^{-1/2}$, meaning (2.24) can be rewritten as

$$(\partial_t + \partial_x|D|^{-1/2})(\partial_t - \partial_x|D|^{-1/2})u = 0. \tag{2.25}$$

$|D|^{-1/2}$ has the integral representation

$$|D|^{-1/2}u(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{u(y)\,dy}{\sqrt{|x-y|}}. \tag{2.26}$$

Thus, the One-Way Water Wave Equation (OWWWE) can be written as

$$\frac{\partial u(x,t)}{\partial t} \pm \frac{\partial}{\partial x} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{u(y,t)\,dy}{\sqrt{|x-y|}} = 0 \tag{2.27}$$

where $+/-$ corresponds to right/left moving waves. Note that the OWWWE is a fractional partial differential equation. Some areas of application where fractional partial differential equations arise are sedimentation [3] and hydrology [20]. Note that the operator $|D|^{-1/2}$ is a fractional derivative using the left- or right-side Riemann-Liouville derivatives:

$$(D_{-\infty+}^{-1/2}u)(x) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{x} \frac{u(y)\,dy}{\sqrt{x-y}}$$
$$(D_{\infty-}^{-1/2}u)(x) = \frac{1}{\sqrt{\pi}} \int_{x}^{\infty} \frac{u(y)\,dy}{\sqrt{y-x}}$$

2.3 Numerical Method

For the rest of the chapter, the right moving waves will be the point of discussion. The analysis for the left moving waves is similar. To solve the OWWWE numerically, we view (2.27) as a conservation law

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial}{\partial x} f(x;u) = 0 \tag{2.28}$$

where

$$f(x;u) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{u(y,t)\,dy}{\sqrt{|x-y|}}. \tag{2.29}$$

(2.28) is a conservation law with a linear nonlocal flux f that involves an integrable square root singularity.

Let our spatial domain, $x \in [0, L]$, be divided uniformly into N intervals $I_j = [x_{j-1/2}, x_{j+1/2}]$, with $|I_j| = \Delta x = L/N$, for $j = 1, \ldots, N$. Let $x_{m-1/2} = (m-1)\Delta x$, for $m = \{1, \ldots, N+1\}$ and let x_j be the center of I_j. For the purpose of this chapter, let $L = 1$. Let $\overline{u}_j(t)$ denote the cell average of $u(\cdot, t)$ on interval I_j

$$\overline{u}_j(t) = \frac{1}{\Delta x} \int_{I_j} u(y,t)\,dy. \tag{2.30}$$

Integrating (2.28) over I_j and dividing by the length of I_j, Δx, gives the semi-discrete system of ODEs

$$\frac{d}{dt}\overline{u}_j(t) = -\frac{1}{\Delta x}\left(f(x_{j+1/2}; u(\cdot,t)) - f(x_{j-1/2}; u(\cdot,t)) \right). \tag{2.31}$$

First, approximate $u(\cdot, t)$ by a local polynomial on I_j. This process will be described in section 2.3.1. Then the flux integral (2.29) can be integrated exactly because the square root singularity is integrable. The system of ODEs simplifies to

$$\frac{d}{dt}\overline{U}(t) = -\frac{1}{\sqrt{\Delta x}} A\overline{U}(t) \tag{2.32}$$

where A is a dense $N \times N$ matrix that arises from the nonlocal flux integral (2.29) and $\overline{U} = (\overline{u}_1, \overline{u}_2, \ldots, \overline{u}_N)^T$ is the vector of cell averages of the numerical solution.

2.3.1 Constructing the coefficient matrix A

To construct the coefficient matrix A, the flux function $f(u,x)$ needs to be evaluated.

$$\begin{aligned} f(x;u) =& \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{u(y,t)\,dy}{\sqrt{|x-y|}} = \frac{1}{\sqrt{2\pi}} \sum_{j=-\infty}^{\infty} \int_{I_j} \frac{u(y,t)\,dy}{\sqrt{|x-y|}} \\ \approx& \frac{1}{\sqrt{2\pi}} \sum_{j=1}^{N} \int_{I_j} \frac{u(y,t)\,dy}{\sqrt{|x-y|}} \approx \frac{1}{\sqrt{2\pi}} \sum_{j=1}^{N} \int_{I_j} \frac{P_j(y,t)\,dy}{\sqrt{|x-y|}} = F(P,x) \end{aligned} \tag{2.33}$$

where $P_j(y)$ is a local polynomial approximation of $u(y,t)$ on interval I_j.

For the rest of this chapter, consider $P_j(y)$ to be a polynomial of degree 1. We choose $P_j(y)$ so that it has the following properties:

- $\dfrac{1}{\Delta x} \displaystyle\int_{I_{j-1}} P_j(y)\,dy = \overline{u}_{j-1}$
- $\dfrac{1}{\Delta x} \displaystyle\int_{I_j} P_j(y)\,dy = \overline{u}_j$

We use basis functions to express $P_j(y)$, i.e. $P_j(y) = \overline{u}_{j-1}\varphi_j^{-1}(y) + \overline{u}_j\varphi_j^0(y)$, where $\varphi_j^{-1}(y), \varphi_j^0(y)$ satisfy the following conditions

$$\begin{aligned} \frac{1}{\Delta x}\int_{I_{j-1}} \varphi_j^{-1}(y)\,dy = 1 \qquad & \frac{1}{\Delta x}\int_{I_j} \varphi_j^{-1}(y)\,dy = 0 \\ \frac{1}{\Delta x}\int_{I_{j-1}} \varphi_j^{0}(y)\,dy = 0 \qquad & \frac{1}{\Delta x}\int_{I_j} \varphi_j^{0}(y)\,dy = 1 \end{aligned} \tag{2.34}$$

A coordinate transformation is used to shift and rescale the intervals to be unit length. Let $z = (y - x_j)/\Delta x$. This change of variables turn (2.34) to

$$\begin{aligned} \int_{-3/2}^{-1/2} \phi_j^{-1}(z)\,dz = 1 \qquad & \int_{-1/2}^{1/2} \phi_j^{-1}(z)\,dz = 0 \\ \int_{-3/2}^{-1/2} \phi_j^{0}(z)\,dz = 0 \qquad & \int_{-1/2}^{1/2} \phi_j^{0}(z)\,dz = 1 \end{aligned} \tag{2.35}$$

where $\phi_j^{-1}(z)$ and $\phi_j^0(z)$ are scaled basis functions. Solving (2.35), the scaled basis functions are

$$\phi_j^{-1}(z) = 1 - (z+1) \qquad \phi_j^0(z) = 1 + z$$

For more examples and stencils for the local polynomial reconstruction, see Appendix B.

Scaling each interval I_j in the numerical flux (2.33) and evaluating at a cell interface $x_{i+1/2}$, $F(P, x_{i+1/2})$ can be evaluated

$$\begin{aligned}
F(P, x_{i+1/2}) &= \frac{1}{\sqrt{2\pi}} \sum_{j=1}^{N} \int_{I_j} \frac{P_j(y)\, dy}{\sqrt{|x_{i+1/2} - y|}} = \frac{1}{\sqrt{2\pi}} \sum_{j=1}^{N} \int_{I_j} \frac{(\overline{u}_{j-1}\varphi_j^{-1}(y) + \overline{u}_j \varphi_j^0(y))\, dy}{\sqrt{|x_{i+1/2} - y|}} \\
&= \frac{1}{\sqrt{2\pi}} \sum_{j=1}^{N} \int_{I_j} \frac{(\overline{u}_{j-1}\varphi_j^{-1}(y) + \overline{u}_j \varphi_j^0(y))\, dy}{\sqrt{|i\Delta x - y|}} \\
&= \frac{1}{\sqrt{2\pi}} \sum_{j=1}^{N} \overline{u}_j \left(\int_{I_{j+1}} \frac{\varphi_{j+1}^{-1}(y)\, dy}{\sqrt{|i\Delta x - y|}} + \int_{I_j} \frac{\varphi_j^0(y)\, dy}{\sqrt{|i\Delta x - y|}} \right) \\
&= \frac{\sqrt{\Delta x}}{\sqrt{2\pi}} \sum_{j=1}^{N} \overline{u}_j \left(\int_{1/2}^{3/2} \frac{\phi_{j+1}^{-1}(z)\, dz}{\sqrt{|i - j + 1/2 - z|}} + \int_{-1/2}^{1/2} \frac{\phi_j^0(z)\, dz}{\sqrt{|i - j + 1/2 - z|}} \right) \\
&= F_{i+1/2}
\end{aligned}$$

At the intervals near the boundary, the stencil will require data from outside the computational domain. The above expression must be modified by (i) assuming the cell averages outside the domain are zero or (ii) using a shifted stencil to ensure that only data from inside the domain is used. For (ii), this means using the linear downwind stencil near the right boundary, which will change a few terms in the sum for $F_{i+1/2}$.

For the linear upwind stencil, this gives the modified expression (i)

$$\begin{aligned}
F_{i+1/2} = \frac{\sqrt{\Delta x}}{\sqrt{2\pi}} \Bigg[& \sum_{j=1}^{N-1} \overline{u}_j \left(\int_{1/2}^{3/2} \frac{\phi_{j+1}^{-1}(y)\, dy}{\sqrt{|i - j + 1/2 - y|}} + \int_{-1/2}^{1/2} \frac{\phi_j^0(y)\, dy}{\sqrt{|i - j + 1/2 - y|}} \right) \\
& + \overline{u}_N \int_{-1/2}^{1/2} \frac{\phi_N^0(y)\, dy}{\sqrt{|i - N + 1/2 - y|}} \Bigg]
\end{aligned}$$

and the modified expression (ii)

$$
\begin{aligned}
F_{i+1/2} \;=\; & \frac{\sqrt{\Delta x}}{\sqrt{2\pi}}\Bigg[\overline{u}_1\left(\int_{1/2}^{3/2}\frac{\phi_2^{-1}(y)\,dy}{\sqrt{|i-1+1/2-y|}}+\int_{-1/2}^{1/2}\frac{\tilde{\phi}_1^{0}(y)\,dy}{\sqrt{|i-1+1/2-y|}}\right)\\
& +\overline{u}_2\int_{-3/2}^{-1/2}\frac{\tilde{\phi}_1^{1}(y)\,dy}{\sqrt{|i-2+1/2-y|}}\\
& +\sum_{j=2}^{N-1}\overline{u}_j\left(\int_{1/2}^{3/2}\frac{\phi_{j+1}^{-1}(y)\,dy}{\sqrt{|i-j+1/2-y|}}+\int_{-1/2}^{1/2}\frac{\phi_j^{0}(y)\,dy}{\sqrt{|i-j+1/2-y|}}\right)\\
& +\overline{u}_N\int_{-1/2}^{1/2}\frac{\phi_N^{0}(y)\,dy}{\sqrt{|i-N+1/2-y|}}\Bigg]
\end{aligned}
$$

where $\tilde{\phi}_j^0(y)$ and $\tilde{\phi}_j^1(y)$ are basis functions for the linear downwind polynomial approximation.

Factoring out the cell averages $\overline{u}_j$ leads to the matrix A. For the unshifted stencil (i), the $N\times N$ matrix A has the following form

$$
A=\frac{1}{\sqrt{2\pi}}\left(\Phi_{-1}^0+\Phi_0^0-\Phi_{-1}^{-1}-\Phi_0^{-1}\right) \tag{2.36}
$$

where

$$
\Phi_{-1}^k=\begin{pmatrix}
I_{-1}(k) & I_{-1}(k-1) & \cdots & I_{-1}(k-(N-2)) & 0\\
I_{-1}(k+1) & \ddots & \ddots & I_{-1}(k-(N-3)) & \vdots\\
I_{-1}(k+2) & \ddots & \ddots & I_{-1}(k-(N-4)) & \vdots\\
\vdots & \ddots & \ddots & \vdots & \vdots\\
I_{-1}(k+N-1) & I_{-1}(k+N-2) & \cdots & I_{-1}(k+1) & 0
\end{pmatrix}
$$

and

$$
\Phi_0^k=\begin{pmatrix}
I_0(k) & I_0(k-1) & \cdots & I_0(k-(N-2)) & I_0(k-(N-1))\\
I_0(k+1) & \ddots & \ddots & \ddots & \ddots\\
I_0(k+2) & \ddots & \ddots & \ddots & \ddots\\
\vdots & \ddots & \ddots & \ddots & I_0(k-1)\\
I_0(k+N-1) & I_0(k+N-2) & \cdots & I_0(k+1) & I_0(k)
\end{pmatrix}
$$

where

$$I_{-1}(k) = \int_{1/2}^{3/2} \frac{\phi_{j+1}^{-1}(y)\,dy}{\sqrt{|k+1/2-y|}} \qquad\qquad I_0(k) = \int_{-1/2}^{1/2} \frac{\phi_j^0(y)\,dy}{\sqrt{|k+1/2-y|}}$$

The matrix A corresponding to the shifted stencil (ii) can be obtained in a similar manner.

2.3.2 Numerical Stability

The matrix A will depend on the choice of polynomial reconstruction. Putting (2.32) into a time-integration scheme will lead to stability bounds in terms of the parameter ν

$$\nu = \frac{\Delta t}{\sqrt{\Delta x}}. \tag{2.37}$$

For a given Runge-Kutta method, let $g(z)$ be the associated stability function. To test the stability of the stencil, $-\nu\lambda_k$ must be within the stability region of the Runge-Kutta method used for integration, where λ_k are the eigenvalues of the matrix A that is associated with the chosen stencil for local polynomial approximation of $u(x,t)$. To find the stability range for ν, the following condition must be satisfied

$$\max_k |g(-\nu\lambda_k)| \leq 1.$$

As the grid is refined, the eigenvalues of A converge to a well-defined curve, which was used to determine the stability range for ν.

In Table 2.1, the stable range for ν is listed to six significant digits when reconstructing with unshifted stencils. In Table 2.2, the stable range for ν is listed to six significant digits when reconstructing with shifted stencils.

Comparing Table 2.1 and 2.2, there does not seem to be a significant difference between using shifted and unshifted stencil reconstructions.

Stencil	Time-Integrator	Stable ν range
Linear upwind	RK4	$0 \leq \nu \leq 2.20803$
Linear upwind	Butcher 6	$0 \leq \nu \leq 2.13109$
Linear upwind	Cooper-Verner RK8	$0 \leq \nu \leq 3.43865$
Quadratic upwind	Butcher 6	$0 \leq \nu \leq 1.40582$
Quadratic upwind	Cooper-Verner RK8	$0 \leq \nu \leq 2.06010$
Cubic slightly upwind	Cooper-Verner RK8	$0 \leq \nu \leq 3.37398$

Table 2.1: Stable ν ranges for various orders, unshifted stencils, and time-integrators

Stencil	Time-Integrator	Stable ν range
Linear Upwind	RK4	$0 \leq \nu \leq 2.20813$
Linear Upwind	Butcher 6	$0 \leq \nu \leq 2.13127$
Linear Upwind	Cooper-Verner RK8	$0 \leq \nu \leq 3.43878$
Quadratic Upwind	Butcher 6	$0 \leq \nu \leq 1.40028$
Quadratic Upwind	Cooper-Verner RK8	$0 \leq \nu \leq 2.06188$
Cubic Slightly Upwind	Cooper-Verner RK8	$0 \leq \nu \leq 3.37410$

Table 2.2: Stable ν ranges for various orders, shifted stencils, and time-integrators

Figure 2.6 shows the eigenvalues of A when using a linear upwind polynomial reconstruction with the corresponding Runge-Kutta stability region and a plot of ν versus $\max_k |g(-\nu\lambda_k)|$ is also shown. Figure 2.7 shows the eigenvalues of A when using a cubic very upwind polynomial reconstruction with the corresponding Runge-Kutta stability region and a plot of ν versus $\max_k |g(-\nu\lambda_k)|$ is also shown. The cubic very upwind polynomial reconstruction is visualized in Figure 2.5.

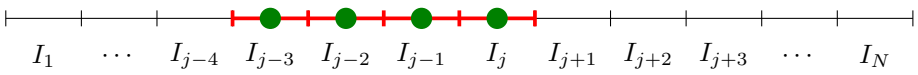

Figure 2.5: Visualization of cubic very upwind stencil

Note that Figure 2.7 shows that a cubic very upwind polynomial reconstruction is unstable, regardless of the choice of ν.

In Figure 2.8, the eigenvalues of some stable stencils are plotted. For the quadratic centered stencil, the computed eigenvalues lie on or near the imaginary axis, a property of the OWWWE (the spectrum of the OWWWE is the entire imaginary axis.) We note that polynomial reconstructions based on downwind or heavily upwind stencils are found to be unstable. Define L as the number of intervals on the left of I_j

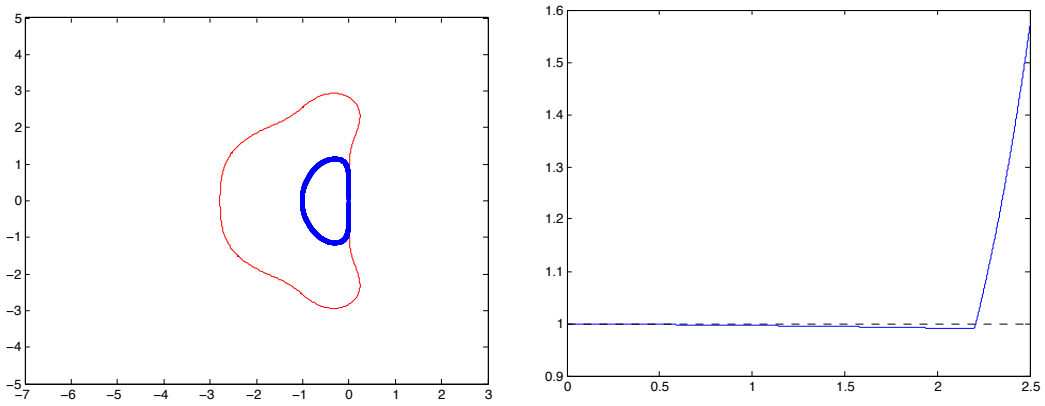

Figure 2.6: RK4 stability region plotted with computed eigenvalues of linear upwind stencil (left figure), Plot of ν versus $\max_k |g(-\nu\lambda_k)|$ (right figure)

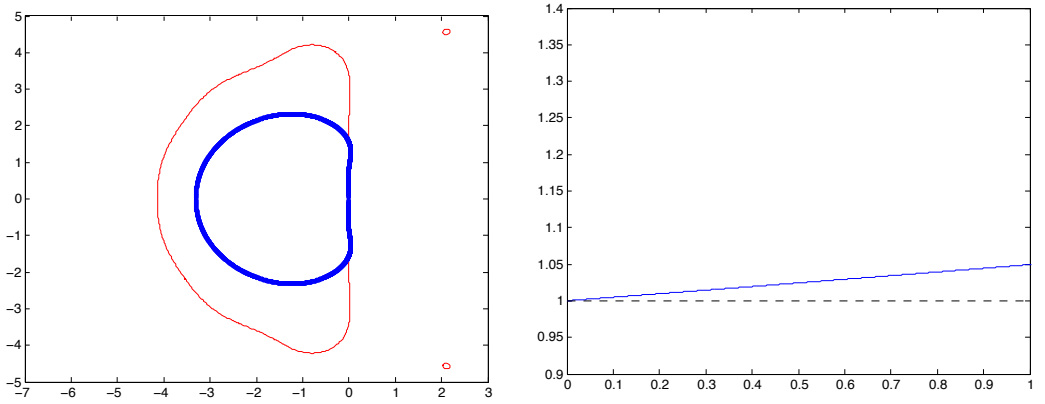

Figure 2.7: Cooper Verner 8 stability region plotted with computed eigenvalues of cubic very upwind stencil (left figure), Plot of ν versus $\max_k |g(-\nu\lambda_k)|$ (right figure)

that the stencil matches cell average values on and R as the number of intervals on the right of I_j that the stencil matches cell average values on. If $0 \leq L - R \leq 2$, then the stencil is stable. If $L - R > 2$, then the stencil is unstable. Similar results are known for the advection equation [9].

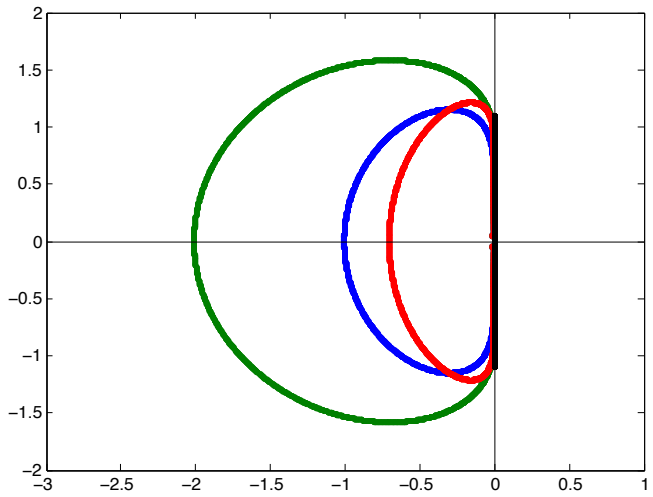

Figure 2.8: Plot of the computed eigenvalues for stable stencils; linear upwind (blue), quadratic upwind (green), quadratic centered (black), cubic slightly upwind (red)

2.3.3 Numerical Accuracy

We want to investigate how accurate a degree p polynomial reconstruction is when evaluating the flux at a point $x_{i+1/2}$. A simple estimate shows

$$\begin{aligned}
|f(u, x_{i+1/2}) - F(P, x_{i+1/2})| &= \left| \sum_{j=1}^{N} \frac{1}{\sqrt{2\pi}} \int_{I_j} \frac{(u(y) - P_j(y))\, dy}{\sqrt{|x_{i+1/2} - y|}} \right| \\
&\leq \sum_{j=1}^{N} \frac{1}{\sqrt{2\pi}} \int_{I_j} \frac{|u(y) - P_j(y)|\, dy}{\sqrt{|x_{i+1/2} - y|}} \leq \sum_{j=1}^{N} \frac{C}{\sqrt{2\pi}} \int_{I_j} \frac{\Delta x^{p+1}\, dy}{\sqrt{|x_{i+1/2} - y|}} \\
&\leq \frac{C\Delta x^{p+1}}{\sqrt{2\pi}} \left(\sum_{j=1}^{i} \int_{I_j} \frac{dy}{\sqrt{x_{i+1/2} - y}} + \sum_{j=i+1}^{N} \int_{I_j} \frac{dy}{\sqrt{y - x_{i+1/2}}} \right)
\end{aligned}$$

Evaluating each integral exactly, the estimate becomes

$$\begin{aligned}
|f(u, x_{i+1/2}) - F(P, x_{i+1/2})| &\leq \frac{C\Delta x^{p+1}}{\sqrt{2\pi}} \left(\sum_{j=1}^{i} \int_{I_j} \frac{dy}{\sqrt{x_{i+1/2} - y}} + \sum_{j=i+1}^{N} \int_{I_j} \frac{dy}{\sqrt{y - x_{i+1/2}}} \right) \\
&= \frac{2C\Delta x^{p+3/2}}{\sqrt{2\pi}} \Big(\sqrt{i} - \sqrt{i-1} + \sqrt{i-1} - \sqrt{i-2} + \cdots \\
&\qquad + \sqrt{N-i} - \sqrt{N-i-1} \Big) \\
&= \frac{2C\Delta x^{p+3/2}}{\sqrt{2\pi}} \left(\sqrt{i} + \sqrt{N-i} \right) \\
&= O(\Delta x^{p+1}).
\end{aligned}$$

Thus, this estimate shows that a degree p polynomial reconstruction is $O(\Delta x^{p+1})$ accurate when evaluating the flux at a point $x_{i+1/2}$.

However, a more careful estimate reveals that $|f(u, x_{i+1/2}) - F(P, x_{i+1/2})| = O(\Delta^{p+3/2})$, giving an extra 1/2 in convergence rate. This extra convergence is from using cell averages in the polynomial reconstructions.

Theorem II.1. *Given a degree p polynomial reconstruction for $u(x)$*

$$P(x) = \sum_j P_j(x)\mathcal{X}_{I_j}(x) = u(x) + \mathcal{O}(\Delta x^{p+1})$$

The accuracy of the numerical flux approximation at a point $x = x_{i+1/2}$ is

$$|f(u, x_{i+1/2}) - F(P, x_{i+1/2})| = O(\Delta x^{p+3/2}). \tag{2.38}$$

Proof. Evaluate the flux at the point $x_{i+1/2}$

$$|f(u, x_{i+1/2}) - F(P, x_{i+1/2})| = \left| \sum_{j=1}^{N} \frac{1}{\sqrt{2\pi}} \int_{I_j} \frac{(u(y) - P_j(y))\, dy}{\sqrt{|x_{i+1/2} - y|}} \right| \leq E_1 + E_2 + E_3 + E_4$$

where

$$E_1 = \left| \frac{1}{\sqrt{2\pi}} \int_{I_{i+1}} \frac{(u(y) - P_{i+1}(y))\, dy}{\sqrt{|x_{i+1/2} - y|}} \right| \qquad E_2 = \left| \frac{1}{\sqrt{2\pi}} \int_{I_i} \frac{(u(y) - P_i(y))\, dy}{\sqrt{|x_{i+1/2} - y|}} \right|$$

$$E_3 = \left| \frac{1}{\sqrt{2\pi}} \sum_{j=i+2}^{N} \int_{I_j} \frac{(u(y) - P_j(y))\, dy}{\sqrt{|x_{i+1/2} - y|}} \right| \qquad E_4 = \left| \frac{1}{\sqrt{2\pi}} \sum_{j=1}^{i-1} \int_{I_j} \frac{(u(y) - P_j(y))\, dy}{\sqrt{|x_{i+1/2} - y|}} \right|$$

First, we estimate E_1. The estimate for E_2 is obtained similarly.

$$E_1 \leq C_1 \Delta x^{p+1} \int_{I_i} \frac{1}{\sqrt{|x_{i+1/2} - y|}} dy = C_1 \Delta x^{p+1} \sqrt{\Delta x} = C \Delta x^{p+3/2}.$$

Next, we estimate E_3. The estimate for E_4 is obtained similarly. We look at the j^{th} term of the sum. By integration by parts, we have

$$\left[\frac{\int_{x_{j-1/2}}^{y} (u(s) - P(s)) ds}{\sqrt{y - x_{i+1/2}}} \right]_{y=x_{j-1/2}}^{y=x_{j+1/2}} + \frac{1}{2} \int_{x_{j-1/2}}^{x_{j+1/2}} \frac{\int_{x_{j-1/2}}^{y} (u(s) - P(s)) ds}{(y - x_{i+1/2})^{3/2}} dy$$

$$= \frac{1}{2} \int_{x_{j-1/2}}^{x_{j+1/2}} \frac{\int_{x_{j-1/2}}^{y} (u(s) - P(s)) ds}{(y - x_{i+1/2})^{3/2}} dy$$

The estimate for E_3 becomes

$$\begin{aligned}
E_3 &\leq \sum_{j=i+2}^{N} \left| \frac{1}{\sqrt{2\pi}} \int_{I_j} \frac{(u(y) - P(y))\, dy}{\sqrt{|x_{i+1/2} - y|}} \right| = \frac{1}{\sqrt{2\pi}} \sum_{j=i+2}^{N} \frac{1}{2} \left| \int_{I_j} \frac{\int_{x_{j-1/2}}^{y} (u(s) - P(s)) ds}{(y - x_{i+1/2})^{3/2}} dy \right| \\
&\leq \sum_{j=i+2}^{N} C_2 \Delta x^{p+2} \frac{1}{2} \int_{x_{j-1/2}}^{x_{j+1/2}} \frac{dy}{(y - x_{i+1/2})^{3/2}} = \frac{C_2}{2} \Delta x^{p+2} \int_{x_{i+3/2}}^{L} \frac{dy}{(y - x_{i+1/2})^{3/2}} \\
&= C_2 \Delta x^{p+2} \left(\frac{1}{\sqrt{x_{i+3/2} - x_{i+1/2}}} - \frac{1}{\sqrt{L - x_{i+1/2}}} \right) \leq C_2 \Delta x^{p+3/2}
\end{aligned}$$

Combining all four estimates, we have

$$|f(u, x_{i+1/2}) - F(P, x_{i+1/2})| \leq E_1 + E_2 + E_3 + E_4 \leq \widetilde{C} \Delta x^{p+3/2}.$$

□

Assuming the local polynomial reconstruction has order of accuracy $\Delta x^{p+3/2}$ and the time-integration scheme has order of accuracy Δt^q, the overall order of accuracy for the time-dependent problem is

$$\mathcal{O}(\Delta t^q + \Delta x^{p+3/2}) = \mathcal{O}((\sqrt{\Delta x})^q + \Delta x^{p+3/2}) = \mathcal{O}(\Delta x^{q/2} + \Delta x^{p+3/2}). \tag{2.39}$$

Thus, to be able to test convergence of the time-dependent solution, the chosen time-integration scheme should be, in terms of order of accuracy, twice as accurate as the spatial order of accuracy. For example, a degree 1 polynomial reconstruction has order of accuracy $\Delta x^{2.5}$. This means to test convergence of the time-dependent solution, a time integration method must have order of accuracy Δt^5. The approximate numerical order of accuracy for some stencils are listed in Table 2.3.

Stencil	Order of accuracy
Linear upwind	≈ 2.5
Quadratic upwind	≈ 3.5
Cubic slightly upwind	≈ 4.5

Table 2.3: Approximate orders of accuracy for each stencil in evaluating the flux

2.4 Time-dependent solution

Section 2.3.3 was concerned with the accuracy of the flux function using a local polynomial reconstruction. The accuracy of solving (2.27) at a time T is the concern of this section. After choosing the polynomial reconstruction and time-integrator, one can numerically solve (2.27). From (2.39), the order of accuracy for time integration should be twice as accurate as the order of accuracy for the polynomial reconstruction.

Figure 2.9 shows the solution to the OWWWE that supports right-moving wave propagation on domain $[0, 1]$ with time on the vertical axis for $t \in [0, 10]$. We see the waves propagating towards the right boundary with minimal reflections off the boundary.

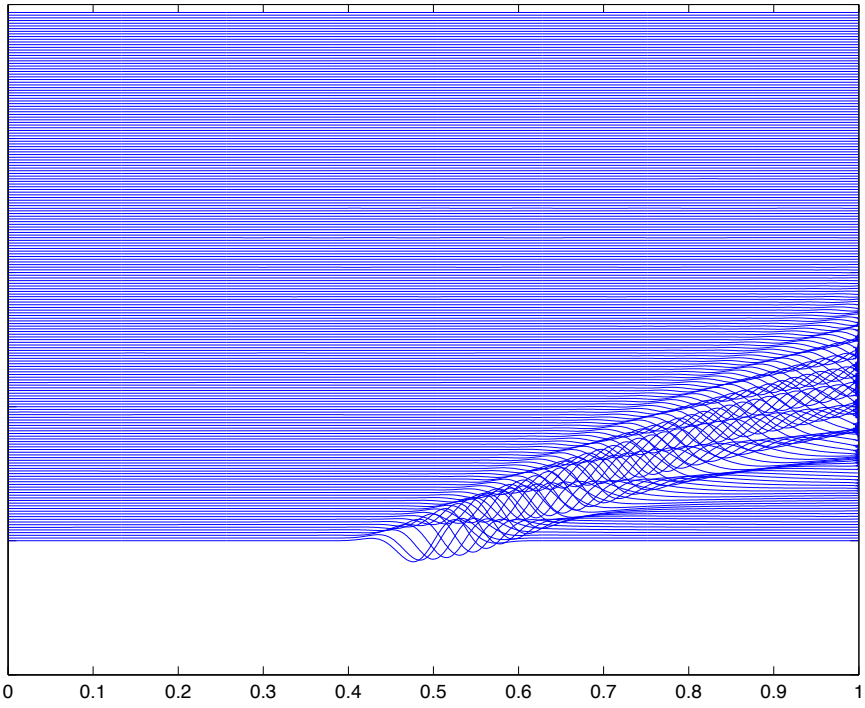

Figure 2.9: Solution to OWWWE for $x \in [0, 1]$. Time is on the vertical axis for $t \in [0, 10]$.

2.4.1 Plots

Figures 2.10, 2.11, and 2.12 are plots of $u(x,2)$ given the initial condition

$$u(x,0) = \begin{cases} \cos^6\left(\frac{20\pi}{6}\left(x-\frac{1}{2}\right)\right)\sin\left(\frac{100\pi}{6}\left(x-\frac{1}{2}\right)\right) & , \quad \frac{7}{20} \leq x \leq \frac{13}{20} \\ 0 & , \quad \text{elsewhere} \end{cases} \tag{2.40}$$

The function (2.40) is chosen because it is an element of $C^6([0,1])$ and is used to test the convergence of the solutions to (2.27) when using polynomial reconstructions of high degree.

A variety of polynomial reconstructions are used on the domain $x \in [0,1]$. The results obtained using the unshifted and shifted stencils exhibited similar convergence behavior. Only the unshifted stencil is shown in Figures 2.10, 2.11, and 2.12.

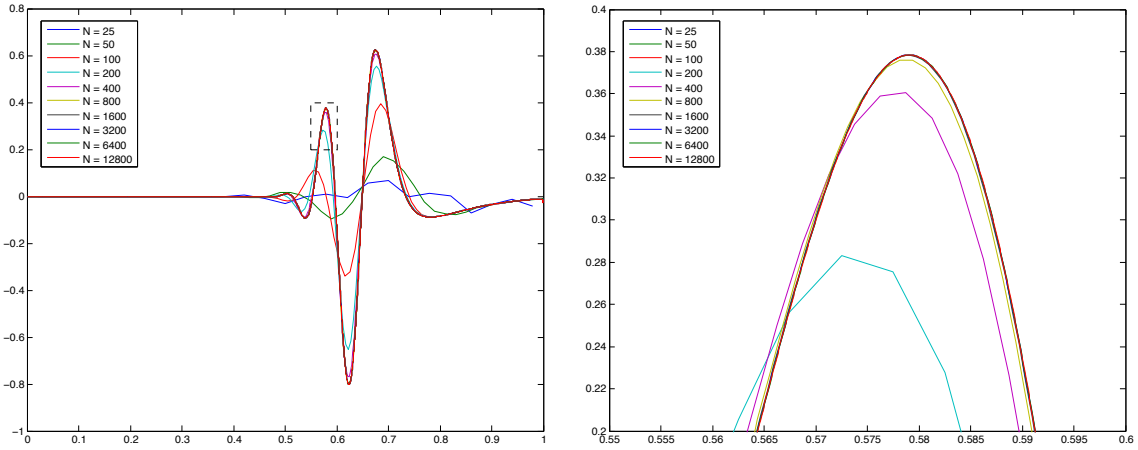

Figure 2.10: $u(x,2)$ using an unshifted linear upwind reconstruction with $\nu = 2.1$

In each case, the numerical rate of convergence is calculated by finding the error between subsequent grid refinements

$$e^N = \Delta x \sum_{j=1}^{N} \left| \overline{u}_j^{\Delta x} - \left(\frac{\overline{u}_{2j-1}^{\Delta x/2} + \overline{u}_{2j}^{\Delta x/2}}{2} \right) \right| \tag{2.41}$$

where $\overline{u}_j^{\Delta x}$ is the cell average on interval I_j using intervals of length Δx. Calculating $p = \log_2(e^N/e^{2N})$ will give the numerical rate of convergence.

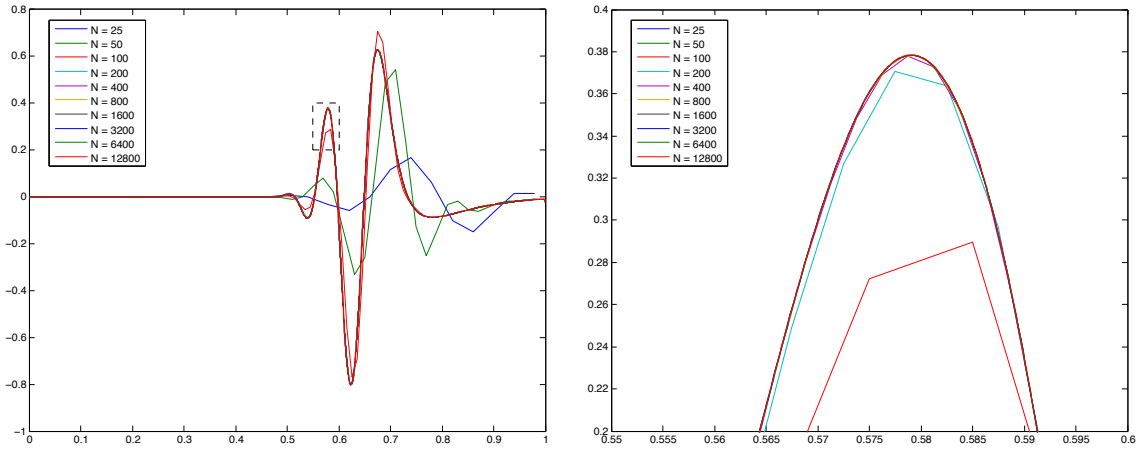

Figure 2.11: $u(x, 2)$ using an unshifted quadratic upwind reconstruction with $\nu = 1.3$

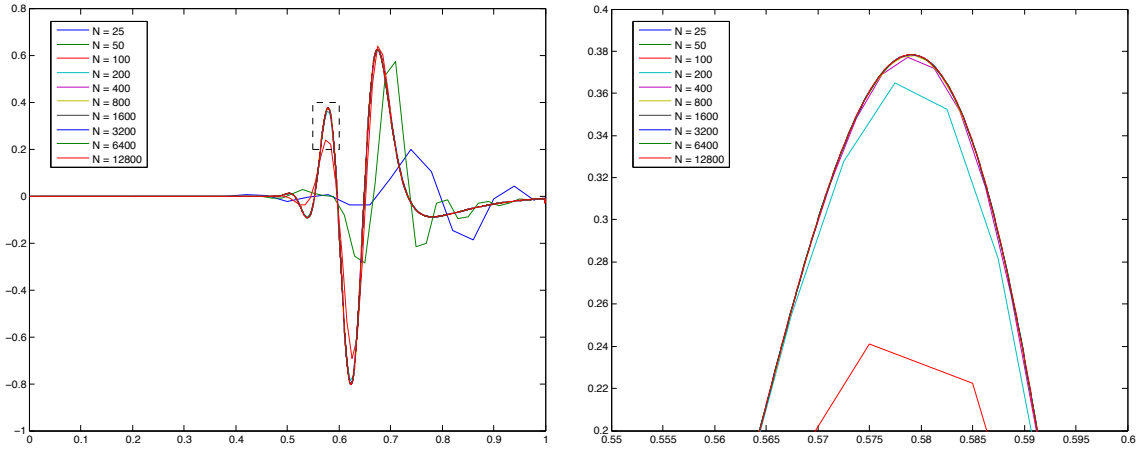

Figure 2.12: $u(x, 2)$ using an unshifted cubic slightly upwind reconstruction with $\nu = 3.3$

Included in Table 2.4 is the numerical error and convergence rate for $u(x, 2)$ using the unshifted quadratic upwind stencil for $x \in [0.2, 0.8]$.

N	L_1 error	L_1 convergence rate	L_∞ error	L_∞ convergence rate
100	1.19798738E-02	3.55465429E+00	1.41394830E-01	3.59353048E+00
200	1.01951748E-03	3.59442267E+00	1.17131327E-02	3.57490098E+00
400	8.44045064E-05	3.53476139E+00	9.82925583E-04	3.53877887E+00
800	7.28276723E-06	3.46757181E+00	8.45750086E-05	3.48865135E+00
1600	6.58344638E-07	3.33312974E+00	7.53448116E-06	3.42031001E+00

Table 2.4: Convergence rate of $u(x, 2)$ using unshifted quadratic upwind stencil for $x \in [0.2, 0.8]$

2.5 OWWWE as absorbing boundary condition for the 1-D WWE

The initial motivation for deriving the OWWWE was to implement it as an absorbing boundary layer for the 1-D WWE. We do this by solving the following system of equations

$$\begin{aligned} u_{tt} + |D|u &= 0 \\ u(x,0) &= u_0(x) \\ u_t(x,0) &= 0 \\ v_t + \frac{\partial}{\partial x}|D|^{-1/2}v &= 0 \\ v(x,0) &= \frac{1}{2}u_0(x) \\ w_t - \frac{\partial}{\partial x}|D|^{-1/2}w &= 0 \\ w(x,0) &= \frac{1}{2}u_0(x) \end{aligned}$$

and combine the solutions to find

$$\widetilde{u}(x,t) = \begin{cases} w(x,t) & : 0 \leq x \leq 0.2 \\ u(x,t) & : 0.2 \leq x \leq 0.8 \\ v(x,t) & : 0.8 \leq x \leq 1 \end{cases}$$

and plot $\widetilde{u}(x,t)$ as a solution of the WWE with OWWWE implemented in an absorbing boundary conditions.

Figure 2.13 shows the solution to the 1-D WWE in the interior of the domain and the solution to the OWWWE in the absorbing boundary layers near the boundary. We see that the reflections in Figure 2.2 are no longer present in our calculation, demonstrating that the OWWWE is an excellent method to suppress outgoing waves

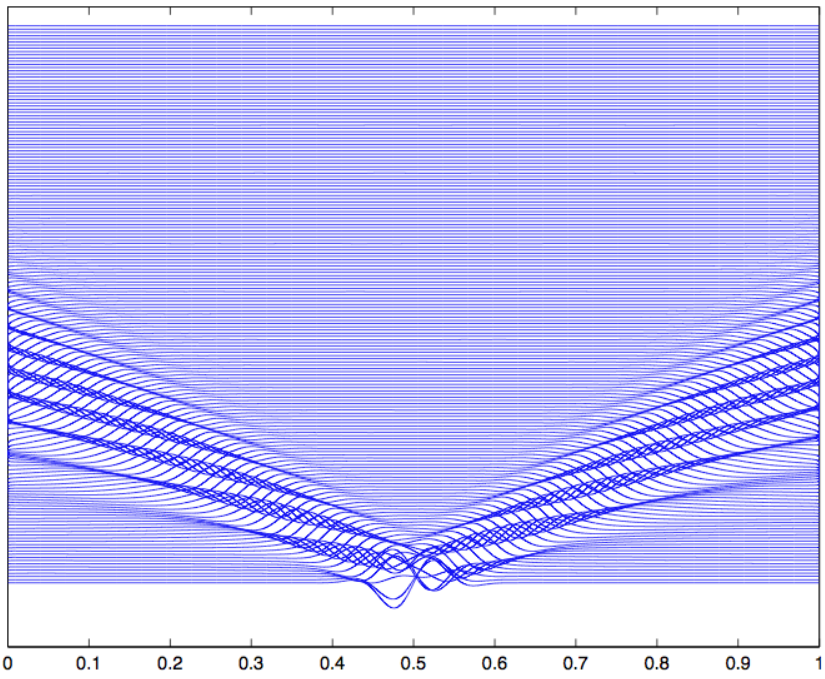

Figure 2.13: Solution to 1-D WWE for $x \in [0, 1]$ with the OWWWE as an absorbing boundary condition. Time is on the vertical axis for $t \in [0, 20]$.

from reflecting off the boundary. In [11, 12], it is shown that (2.27) is actually a perfectly absorbing boundary condition for the 1-D WWE. However, because implementing (2.27) requires truncating infinite integrals, it is not a perfectly absorbing boundary condition in practice.

We note that there are ways to increase the absorption in the layer by adding damping effects to the OWWWE, through additive or exponential damping. Though the OWWWE is a perfectly absorbing boundary condition, one could still add more damping effects to reduce the size of the outgoing wave before it reaches the boundary.

2.6 Implementing the OWWWE on nonuniform grids

In cases where we want to model the OWWWE on a larger domain, one can consider expanding the grid near the boundary with nonuniform grid spacing. This requires an extension of implementing the OWWWE to nonuniform grids. Most

importantly, the local polynomial reconstruction must be extended to nonuniform grids.

Consider the spatial domain, $x \in [0,1]$, divided into N intervals $I_j = [x_{j-1/2}, x_{j+1/2}]$, with $|I_j| = \Delta x_j$, for $j = 1, \ldots, N$. Consider a linear upwind polynomial reconstruction. We choose $P_j(y)$ so that it has the following properties:

- $\dfrac{1}{\Delta x_{j-1}} \displaystyle\int_{I_{j-1}} P_j(y)\, dy = \overline{u}_{j-1}$
- $\dfrac{1}{\Delta x_j} \displaystyle\int_{I_j} P_j(y)\, dy = \overline{u}_j$

The basis functions for a linear upwind polynomial reconstruction are

$$\varphi_j^{-1}(y) = \frac{-2}{\Delta x_j + \Delta x_{j-1}} (y - x_j)$$
$$\varphi_j^{0}(y) = \frac{2}{\Delta x_j + \Delta x_{j-1}} (y - x_{j-1})$$

Finding basis functions for a degree p polynomial are found similarly. Once the basis functions are found, (2.36) will need to be calculated for any given nonuniform grid, as each entry in (2.36) is dependent on the grid spacing. Once we have found (2.36) for each nonuniform grid, we can test the numerical accuracy of using a local polynomial reconstruction in the flux integral (2.33) to see if the nonuniform grid spacing will affect the convergence rate.

For the results in Table 2.5, we consider the following nonuniform grid: We start our grid at $x = 0$. For $N = 20$, let $\Delta x_1 = 1/20$ and $\Delta x_j = 1.05\Delta x_{j-1}$, for $j = 2, \ldots, 20$. For each subsequent iteration, we refine our mesh by dividing each Δx by 2. From Table 2.5, we see the numerical accuracy of the flux does not suffer from a reduced convergence rate.

For the results in Figure 2.14, we consider the following nonuniform grid: We start our grid at $x = 0$. For $n = 100$, let $\Delta x_i = 1/100$ for $i = 1, \ldots, 90$. Let $\Delta x_i = 1.35\Delta x_{i-1}$ for $i = 91, \ldots, 100$. For each subsequent iteration, we refine our

N	L_1 error	L_1 convergence rate	L_∞ error	L_∞ convergence rate
20	7.55013715E-02	0.75594116E+00	4.84061363E-03	1.23151991E+00
40	4.47088887E-02	2.51334992E+00	2.06146509E-03	2.82636138E+00
80	7.83069228E-03	2.96212386E+00	2.90640699E-04	2.94056358E+00
160	1.00487496E-03	2.86236497E+00	3.78580801E-05	2.86554056E+00
320	1.38182901E-04	2.74645623E+00	5.19451401E-06	2.73707975E+00
640	2.05915291E-05	2.64438328E+00	7.79115445E-07	2.63213940E+00
1280	3.29344116E-06	2.57732102E+00	1.25675002E-07	2.56693481E+00
2560	5.51821950E-07	2.53979296E+00	2.12092115E-08	2.53276650E+00
5120	9.48953880E-08	2.52077463E+00	3.66510006E-09	2.51584363E+00
10240	1.65354617E-08	2.51112727E+00	6.40827938E-10	2.50767226E+00

Table 2.5: Convergence rate of $f(u, x)$ when using a linear upwind polynomial reconstruction on a nonuniform grid

mesh by dividing each Δx by 2. In Figure 2.14, we see the wave propagating towards the right boundary, similar to Figure 2.9.

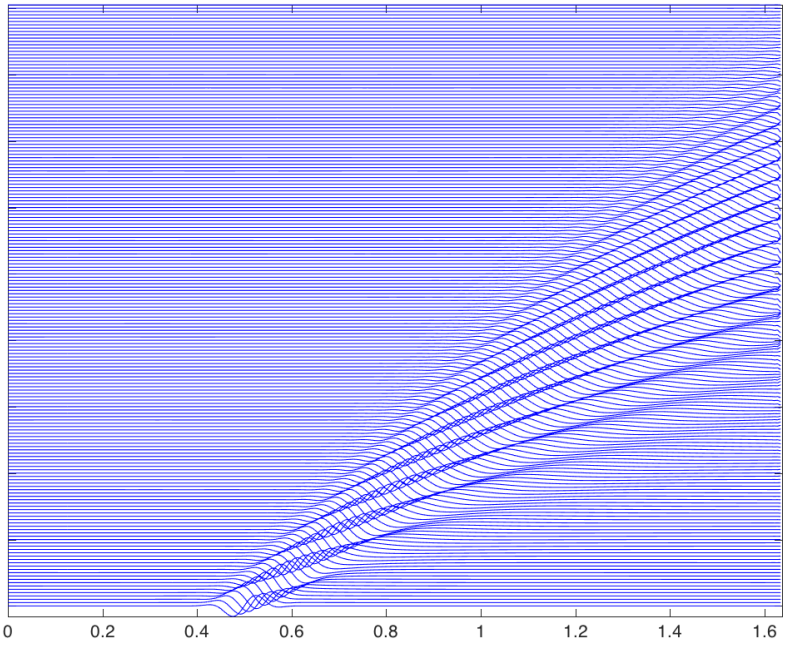

Figure 2.14: Solution to OWWWE for $x \in [0, 1.64]$ with nonuniform grid spacing. Time is on the vertical axis for $t \in [0, 25]$.

For more convergence tables on nonuniform grids, see [4].

CHAPTER III

The 2-D Linearized Water Wave Equations

In Chapter II, we investigated the 1-D WWE and its absorbing boundary conditions. In this chapter, we will introduce the 2-D WWE, as well as discuss some of its properties. We will develop absorbing boundary conditions for the 2-D WWE in Chapter IV.

In lieu of providing the entire derivation of the equations, we will only provide the initial physical setup of the water waves and the resulting quasilinear system and its linearization. For a more detailed discussion on the derivation of the 2-D WWE, see [19, 17].

3.1 The 2-D WWE

Consider the motion of the interface separating an inviscid, incompressible, irrotational fluid from a region of zero density in three-dimensional space. Assume the surface tension on the interface is zero, the density of the fluid is one, and the gravitational field is $(0, 0, -1)$. For time $t \geq 0$, denote the free interface as $\Gamma(t)$ and the region the fluid occupies, below the interface, as $\Omega(t)$. The motion of the fluid is

described by the 3-D incompressible, irrotational, inviscid Euler equations

$$\vec{v}_t + (\vec{v} \cdot \nabla)\vec{v} = (0, -1) - \nabla p \qquad x \in \Omega(t), t \geq 0 \tag{3.1}$$

$$\nabla \cdot \vec{v} = 0 \qquad x \in \Omega(t), t \geq 0 \tag{3.2}$$

$$\nabla \times \vec{v} = 0 \qquad x \in \Omega(t), t \geq 0 \tag{3.3}$$

where $\vec{v} = (v_1, v_2, v_3)$ is the fluid velocity and p is the fluid pressure. Because surface tension is ignored, the boundary conditions are

$$\begin{aligned} &p = 0 \text{ on } \Gamma(t) \\ &(1, \vec{v}) \text{ is tangent to the free surface } (t, \Gamma(t)) \end{aligned} \tag{3.4}$$

We want to find solutions of the system (3.1)–(3.4), for given initial conditions, such that for every fixed $t \geq 0$, $\Gamma(t)$ approaches the xy-plane as $|(x, y)| \to \infty$, and $|\vec{v}(x, y, z; t)| \to 0$ and $|\vec{v}_t(x, y, z; t)| \to 0$ as $|(x, y, z)| \to \infty$.

After writing (3.1)–(3.4) as a quasilinear system and linearizing (see [17, 19] for details), the 2-D WWE is the following system of equations

$$u_{tt}(x, y, t) + |D|u(x, y, t) = 0 \tag{3.5}$$

$$u(x, y, 0) = u_0(x, y) \tag{3.6}$$

$$u_t(x, y, 0) = u_1(x, y) \tag{3.7}$$

where $|D|$ is defined as

$$|D| = \sqrt{-\Delta} = \sqrt{-\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2}} \tag{3.8}$$

and has the Fourier symbol

$$\widehat{|D|u}(\xi_1, \xi_2) = \left(\sqrt{\xi_1^2 + \xi_2^2}\right) \widehat{u}(\xi_1, \xi_2). \tag{3.9}$$

To write this in physical space, we consider plane wave solutions to (3.5) of the form $\xi(x,y,t) = e^{i(\xi_1 x+\xi_2 y-\omega t)}$. These plane waves satisfy the dispersion relation

$$-\omega^2 + \sqrt{\xi_1^2+\xi_2^2} = 0. \tag{3.10}$$

We re-write (3.10) as

$$\begin{aligned} -\omega^2 + \sqrt{\xi_1^2+\xi_2^2} &= 0 \\ -\omega^2 + \frac{\xi_1^2+\xi_2^2}{\sqrt{\xi_1^2+\xi_2^2}} &= 0 \\ -\omega^2 + (i\xi_1)\frac{(-i\xi_1)}{\sqrt{\xi_1^2+\xi_2^2}} + (i\xi_2)\frac{(-i\xi_2)}{\sqrt{\xi_1^2+\xi_2^2}} &= 0. \end{aligned}$$

The Fourier symbol $\dfrac{-i\xi_1}{\sqrt{\xi_1^2+\xi_2^2}}$ corresponds to convolution with the Riesz kernel:

$$R_1 f(x,y) = c_d P.V. \int_{\mathbb{R}^d} \frac{(x-z_1)f(z)\,dz}{|x-z|^{d+1}}$$

where

$$c_d = \frac{\Gamma((d+1)/2)}{\pi^{(d+1)/2}}.$$

The Fourier symbol $\dfrac{-i\xi_2}{\sqrt{\xi_1^2+\xi_2^2}}$ is handled analogously. Because $d=2$, the 2-D WWE can be written in physical space as

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} &+ \frac{1}{2\pi}\frac{\partial}{\partial x} P.V. \int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(x-z_1)u(z_1,z_2,t)\,dz_1 dz_2}{((x-z_1)^2+(y-z_2)^2)^{3/2}} \\ &+ \frac{1}{2\pi}\frac{\partial}{\partial y} P.V. \int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(y-z_2)u(z_1,z_2,t)\,dz_1 dz_2}{((x-z_1)^2+(y-z_2)^2)^{3/2}} = 0 \end{aligned} \tag{3.11}$$

For details on the numerical method and implementation of (3.11), see Appendix C.

3.2 Dispersion relation for 2-D WWE

Recall the dispersion relation (3.10) for (3.5):

$$(-i\omega)^2 + \sqrt{\xi_1^2+\xi_2^2} = 0 \to \omega^2 = \sqrt{\xi_1^2+\xi_2^2}.$$

This has group velocity

$$\left(\frac{\partial\omega}{\partial\xi_1}, \frac{\partial\omega}{\partial\xi_2}\right) = \left(\frac{\xi_1}{2\omega\sqrt{\xi_1^2+\xi_2^2}}, \frac{\xi_2}{2\omega\sqrt{\xi_1^2+\xi_2^2}}\right). \tag{3.12}$$

Similar to the 1-D WWE, we see that the group velocity in the x and y direction support propagation in both positive and negative directions. When creating absorbing boundary equations, we want to find a dispersion relation that will support wave propagation in only one direction. However, we know that waves in 2-D will propagate at an angle of incidence with the boundary.

3.3 Numerical Solution of 2-D WWE

We are interested in solutions to (3.5)–(3.7) on unbounded domains. To compute solutions, we truncate our domain to a finite size and seek boundary conditions that make the artificial boundary transparent to the outgoing waves. Because we do not store any information about the solution $u(x, y, t)$ to (3.5) outside of the truncated domain, Dirichlet boundary conditions are used, with $u(x, y, t) = 0$.

To see the behavior of (3.5)–(3.7) as the solution reaches the artificial boundary, we numerically compute a solution. For details on the numerical method for computing a solution to the 2-D WWE, see Appendix C. Figure 3.1 shows the solution of (3.5)–(3.7) given the following initial data

$$u(x, y, 0) = \exp(-100(x - 0.5)^2 - 100(y - 0.5)^2) \tag{3.13}$$

$$u_t(x, y, 0) = 0 \tag{3.14}$$

As expected, without absorbing boundary conditions, the outgoing waves will partially reflect off the boundary and propagate back towards the interior of the domain. To model the 2-D WWE on an unbounded domain, we need boundary conditions that absorb outgoing waves as they reach the boundary. This is a difficult

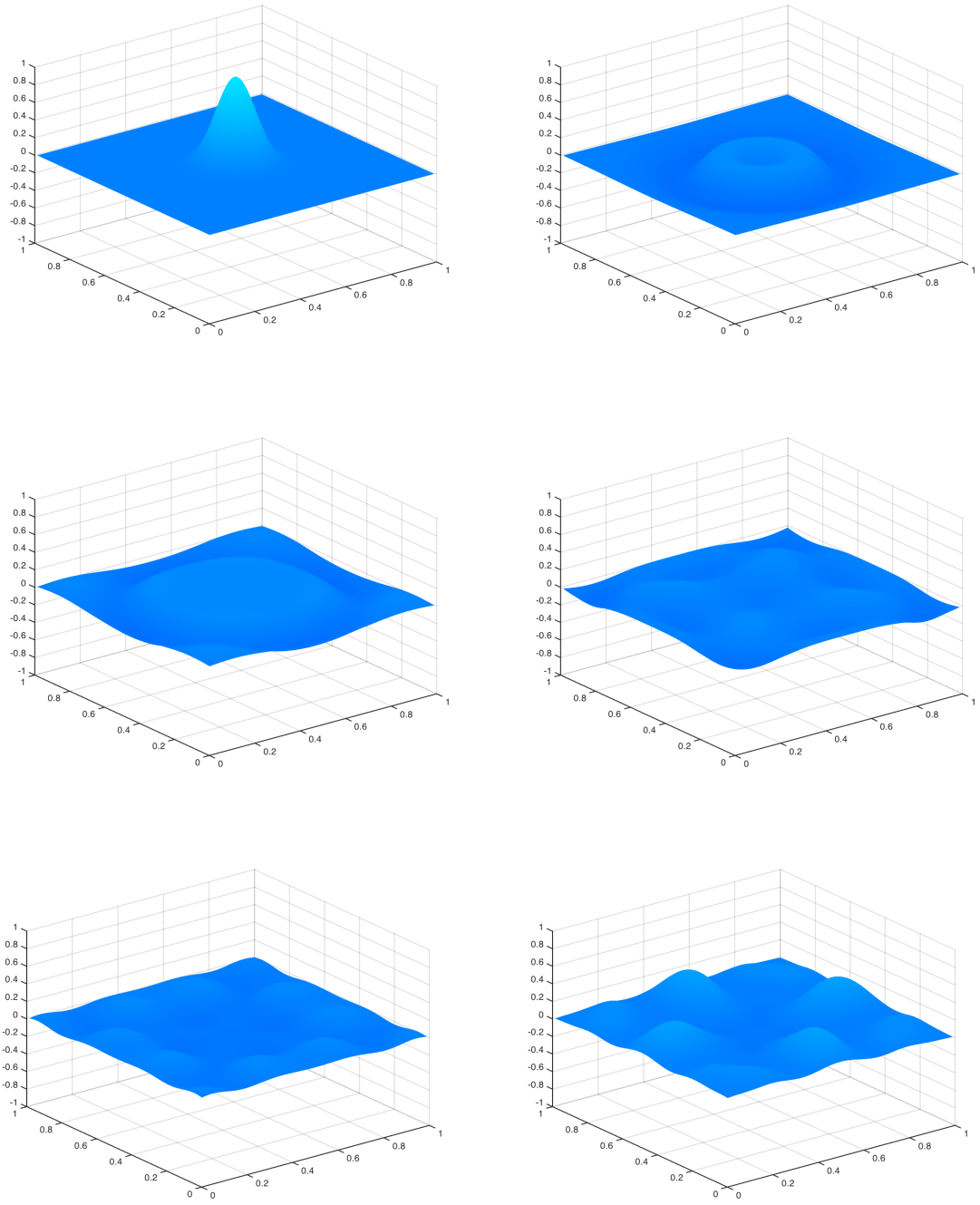

Figure 3.1: Solution $u(x, y, T)$ for the 2-D WWE without boundary conditions for $T = 0$ (top left), $T \approx 2$ (top right), $T \approx 4$ (middle left), $T \approx 6$ (middle right), $T \approx 8$ (bottom left), $T = 10$ (bottom right)

task, as there is an angle of incidence as the waves reach the boundary that will make it hard to develop absorbing boundary conditions.

We note that the 2-D WWE reduces to the 1-D WWE on an infinite domain, given initial condition that depends only on one spatial variable. For example, if given initial condition $u(x, y, 0) = u(x, 0)$, the solution to the 2-D WWE will be identical to the solution of the 1-D WWE, for any constant y-value. However, because we truncate the computational domain in both x and y when we numerically solve the 2-D WWE, the numerical solution of the 2-D WWE no longer reduces to the 1-D WWE numerically. This causes reflections to occur in both x and y directions, mostly seen on the cross boundary, even with initial condition $u(x, y, 0) = u(x, 0)$.

Figure 3.2 shows the solution of (3.5)–(3.7) given the following initial data

(3.15)

$$u(x, y, 0) = u(x, 0) = \begin{cases} \cos^6\left(\frac{20\pi}{6}\left(x - \frac{1}{2}\right)\right)\sin\left(\frac{100\pi}{6}\left(x - \frac{1}{2}\right)\right) & , \quad \frac{7}{20} \leq x \leq \frac{13}{20} \\ 0 & , \quad \text{elsewhere} \end{cases}$$

(3.16)

$$u_t(x, y, 0) = 0$$

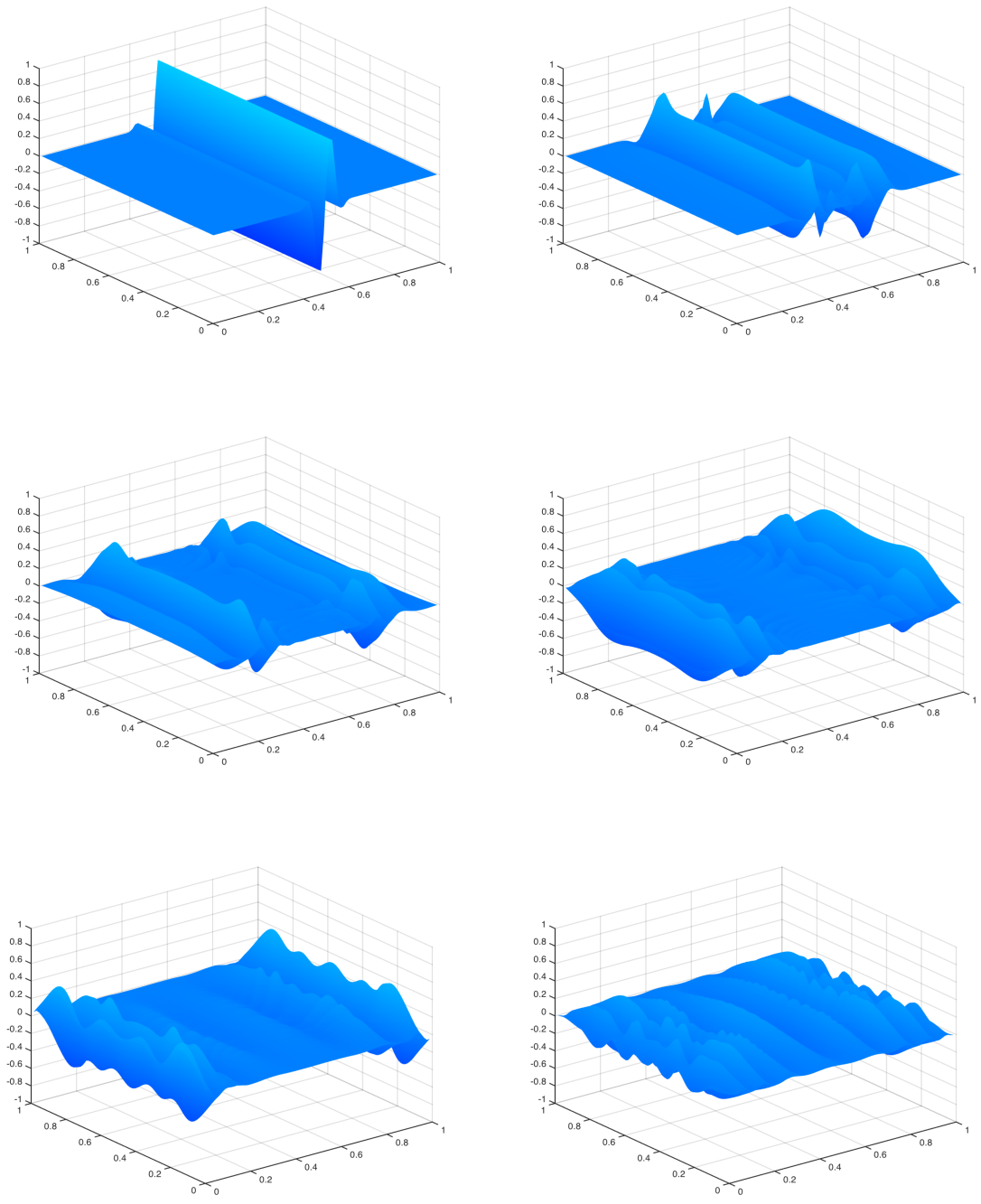

Figure 3.2: Solution $u(x, y, T)$ for the 2-D WWE, with 1-D initial conditions, without boundary conditions for $T = 0$ (top left), $T \approx 2$ (top right), $T \approx 4$ (middle left), $T \approx 6$ (middle right), $T \approx 8$ (bottom left), $T = 10$ (bottom right)

CHAPTER IV

Absorbing Boundaries for the 2-D Linearized Water Wave Equations

In Chapter III, we presented the 2-D WWE and observed reflections occurring when outgoing waves reach the artificial boundary of the truncated domain if we do not implement absorbing boundary conditions. The goal of this chapter is to derive absorbing boundary conditions for the 2-D WWE in order to reduce the amount of reflections of the outgoing waves off the boundary.

4.1 The 2-D Wave Equation

As motivation to derive absorbing boundary conditions for the 2-D WWE, we first illustrate ideas on the more familiar 2-D wave equation

$$u_{tt} = u_{xx} + u_{yy}. \tag{4.1}$$

The goal is to gain insight as to how to proceed in deriving absorbing boundary conditions for the 2-D WWE.

Plane wave solutions of the form $\psi(x, y, t) = e^{i(\xi_1 x + \xi_2 y - \omega t)}$ satisfy the dispersion relation

$$(-i\omega)^2 = (i\xi_1)^2 + (i\xi_2)^2 \Longrightarrow \omega^2 = \xi_1^2 + \xi_2^2. \tag{4.2}$$

To identify one way moving waves, we write

$$\omega = \pm \mathrm{sgn}(\xi_1)|\xi_1| \left(1 + \frac{\xi_2^2}{\xi_1^2}\right)^{1/2} = \pm \xi_1 \left(1 + \frac{\xi_2^2}{\xi_1^2}\right)^{1/2}. \tag{4.3}$$

To develop an absorbing boundary layer, we approximate (4.3) when right and left moving waves are at normal incidence to the boundary, $\xi_2 \ll \xi_1$.

For the purpose of this chapter, the computational domain is $[0,1] \times [0,1]$. Let N be the number of cells in the x-direction and in the y-direction, so $\Delta x = \Delta y = 1/N$.

4.1.1 Stability of first approximation of the 2-D wave equation's Fourier symbol

For the right moving waves, a first order Taylor approximation of (4.3) for $\dfrac{\xi_2}{\xi_1}$ near 0 is

$$\omega = \xi_1 \Longrightarrow -i\omega + i\xi_1 = 0 \tag{4.4}$$

which corresponds to the familiar 1-D advection equation

$$\left(\frac{\partial}{\partial t} + \frac{\partial}{\partial x}\right) u = 0. \tag{4.5}$$

Approximating $\dfrac{\partial}{\partial x}$ by backward differences, we get the $N \times N$ matrix

$$\frac{\partial}{\partial x} u \approx \frac{1}{\Delta x} \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -1 & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix} U = \frac{1}{\Delta x} A U$$

where $U = (u_1, u_2, \ldots, u_N)$ is the vector of values of $u(\cdot, y, t)$.

The stability of this approximation may be studied by considering the eigenvalues of A. The eigenvalues of A are $\lambda = 1$, which we loosely denote by $A = O(1)$. The eigenvalues of the first derivative are $O(\Delta x^{-1})$.

Discretizing (4.5), we get

$$\frac{U^{n+1}-U^n}{\Delta t}+\frac{1}{\Delta x}AU=0 \Longrightarrow U^{n+1}=U^n+\frac{\Delta t}{\Delta x}AU=0. \tag{4.6}$$

We note that (4.6) has a stability scaling of $\Delta t = O(\Delta x)$.

4.1.2 Stability of second approximation of the 2-D wave equation's Fourier symbol

A second order Taylor approximation of (4.3) for $\frac{\xi_2}{\xi_1}$ near 0 is

$$\omega=\xi_1\left(1+\frac{\xi_2^2}{2\xi_1^2}\right)=\xi_1+\frac{1}{2}\frac{\xi_2^2}{\xi_1} \Longrightarrow -i\omega+i\xi_1+\frac{(i\xi_2)^2}{4}\frac{2}{i\xi_1}=0. \tag{4.7}$$

One approach to implementing (4.7) as an equation in an absorbing boundary layer is to clear the denominator in (4.7) and implement the resulting equation in the absorbing boundary layer. Clearing the denominator in (4.7),

$$(-i\omega)(i\xi_1)+(i\xi_1)^2+\frac{(i\xi_2)^2}{2}=0. \tag{4.8}$$

In physical space, (4.8) corresponds to

$$\frac{\partial^2 u}{\partial t\partial x}+\frac{\partial^2 u}{\partial x^2}+\frac{1}{2}\frac{\partial^2 u}{\partial y^2}=0. \tag{4.9}$$

Another approach to implementing (4.7) as an equation in an absorbing boundary layer is to implement the equation without clearing the denominator. The following analysis is done by not clearing the denominator and interpreting (4.7).

The inverse Fourier transform of $\frac{2}{i\xi_1}$ is

$$\frac{1}{2\pi}P.V.\int_{-\infty}^{\infty}\frac{2}{i\xi_1}e^{ix\xi_1}\,d\xi_1=\operatorname{sgn}(x).$$

In physical space, (4.7) corresponds to

$$\left(\frac{\partial}{\partial t}+\frac{\partial}{\partial x}+\frac{1}{4}\frac{\partial^2}{\partial y^2}\Phi\right)u(x,y,t)=0 \tag{4.10}$$

where Φ is convolution with $\mathrm{sgn}(x)$

$$\Phi u(x,y,t) = \int_{-\infty}^{\infty} \mathrm{sgn}(x-z)u(z,y,t)\,dz. \tag{4.11}$$

We want to discretize (4.10) and compute the corresponding eigenvalues in order to analyze the stability of the scheme. Discretizing (4.11) gives

$$\Phi u(x,y,t) \approx \Delta x \sum_{j=1}^{N} \mathrm{sgn}(x-z_j)u(z_j,y,t) = \Delta x \begin{pmatrix} 0 & -1 & \cdots & -1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} U = \Delta x B U$$

where U is the vector of values of $u(\cdot, y, t)$.

Eigenvalues of numerical approximation of convolution operator Φ

Note that the eigenvalues are purely imaginary because $\Delta x B$ is skew-Hermitian. Figure 4.1 shows the computed eigenvalues for $\Delta x B$, for $N = 20, 30, \ldots, 60$. We observe that the eigenvalues of $\Delta x B$ do not exhibit dependence on Δx.

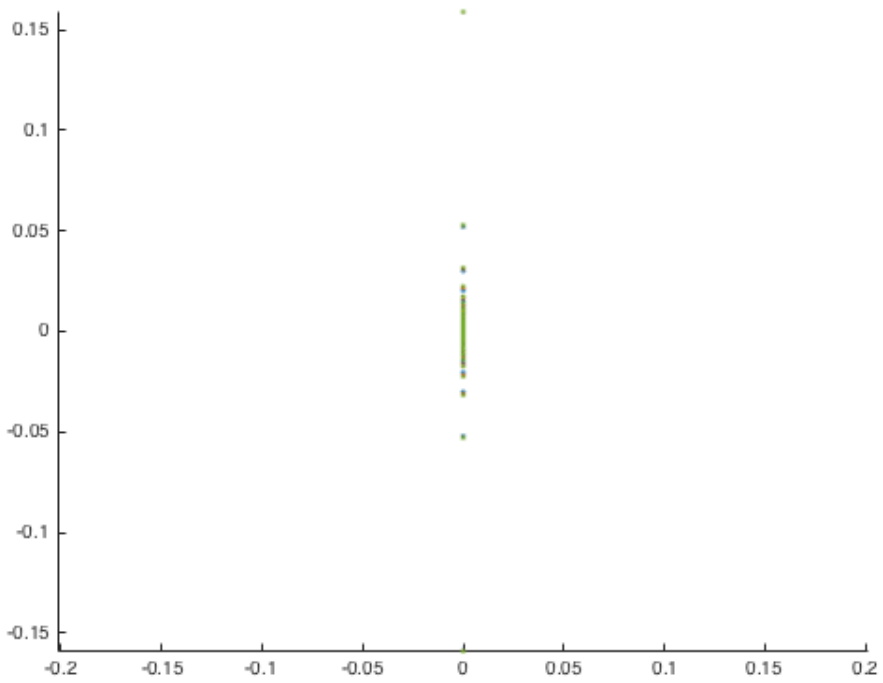

Figure 4.1: Eigenvalues of $\Delta x B$, for $N = 20, 30, \ldots, 60$

Let λ_j^N be the j^{th} eigenvalue of $\Delta x B$. In Table 4.1, we calculate the maximum and minimum eigenvalues of $\Delta x B$ for different values of N. We also calculate $\log_2(\min(|\lambda_j^N|)/\min(|\lambda_j^{2N}|))$ to gain insight into the behavior of eigenvalues as the mesh is refined.

N	Δx	$\max(\lvert\lambda_j^N\rvert)$	$\min(\lvert\lambda_j^N\rvert)$	$\log_2\left(\frac{\min(\lvert\lambda_j^N\rvert)}{\min(\lvert\lambda_j^{2N}\rvert)}\right)$
2^2	2.500000000e-01	6.035533906e-01	1.035533906e-01	2.058241813e+00
2^3	1.250000000e-01	6.284174365e-01	2.486404592e-02	2.014063266e+00
2^4	6.250000000e-02	6.345731492e-01	6.155712710e-03	2.003486077e+00
2^5	3.125000000e-02	6.361083633e-01	1.535214055e-03	2.000869680e+00
2^6	1.562500000e-02	6.364919355e-01	3.835722205e-04	2.000217305e+00
2^7	7.812500000e-03	6.365878141e-01	9.587861234e-05	2.000054319e+00
2^8	3.906250000e-03	6.366117829e-01	2.396875062e-05	2.000013579e+00
2^9	1.953125000e-03	6.366177750e-01	5.992131253e-06	2.000003395e+00
2^{10}	9.765625000e-04	6.366192730e-01	1.498029288e-06	

Table 4.1: Maximum and minimum values of the eigenvalues of $\Delta x B$

From Table 4.1 and Figure 4.1, we can see that $\max(|\lambda_j^N|)$ is $O(1)$ and $\min(|\lambda_j^N|)$ is $O(\Delta x^2)$. Figure 4.1 also shows that the overall shape of the curve of eigenvalues is not changing as N increases, which we loosely denote by $\Delta x B = O(1)$.

Discretizing the $\partial_y^2\Phi$ operator

Discretizing $\dfrac{\partial^2}{\partial y^2}$ by using centered-differencing produces the following $N^2 \times N^2$ matrix

$$\frac{1}{\Delta y^2}\begin{pmatrix} -2\Delta xB & \Delta xB & 0 & \cdots & 0 \\ \Delta xB & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \Delta xB \\ 0 & \cdots & 0 & \Delta xB & -2\Delta xB \end{pmatrix} \widehat{U} = \frac{1}{\Delta y^2} G\widehat{U} \tag{4.12}$$

where $\widehat{U}$ is the vector of solutions at the cell centers in our grid.

Scalar case for eigenvalues of the second derivative using centered-differencing

To gain some intuition for the numerical stability of (4.12), we look at the scalar case of the second derivative.

Consider a domain $[0, 1]$ with N evenly spaced subintervals, so $\Delta x = 1/N$. Consider a function $f(x)$ and its second derivative, $f''(x)$. A centered-difference approximation of $f''(x)$ at $x = x_j$ is

$$f''(x_j) \approx \frac{f(x_{j+1}) - 2f(x_j) + f(x_{j-1})}{\Delta x^2}.$$

In matrix form,

$$\begin{pmatrix} f''(x_1) \\ f''(x_2) \\ \vdots \\ \vdots \\ f''(x_N) \end{pmatrix} \approx \frac{1}{\Delta x^2} \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ \vdots \\ f(x_N) \end{pmatrix} = \frac{1}{\Delta x^2} C \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ \vdots \\ f(x_N) \end{pmatrix}$$

where C is a $N \times N$ matrix. The eigenvalues of C are $\lambda_j = 2(\cos(\pi j \Delta x) - 1)$, for $j = 1, \ldots, N$. The largest, in magnitude, eigenvalue of C, $\lambda_N \approx 2(\cos(\pi N \Delta x) - 1) = -4$ is constant, so the largest eigenvalue of C is $O(1)$. The smallest, in magnitude, eigenvalue of C, $\lambda_1 = 2(\cos(\pi \Delta x) - 1)$ is $O(\Delta x^2)$. The overall shape of the curve of eigenvalues is not changing as N increases, loosely meaning $C = O(1)$.

Returning to the eigenvalues of the discretized $\partial_y^2 \Phi$ operator

Figure 4.2 is a plot of the eigenvalues for the matrix G for $N = 20, 30, \ldots, 60$. As the mesh is refined, the eigenvalues appear to be filling in a curve. This indicates that the stability scaling for (4.12) is $\Delta t = O(\Delta x^2)$, implying $\frac{1}{\Delta y^2} G = O(\Delta y^{-2})$.

Discretizing the entire second approximation

To discretize (4.10), we must discretize and combine $\frac{\partial}{\partial x}$ and $\frac{1}{4} \frac{\partial^2}{\partial y^2} \Phi$:

$$\left(\frac{\partial}{\partial x} + \frac{1}{4} \frac{\partial^2}{\partial y^2} \Phi \right) u(x, y, t) \approx \left[\frac{1}{\Delta x} \begin{pmatrix} A & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A \end{pmatrix} + \frac{1}{\Delta y^2} G \right] \widehat{U}$$

Because $A = O(1)$ and $G = O(1)$, $\frac{1}{\Delta x} A = O(\Delta x^{-1})$ and $\frac{1}{\Delta y^2} G = O(\Delta x^{-2})$. For stable time integration, $\Delta t \lambda_k$ must be inside the stability region of the time

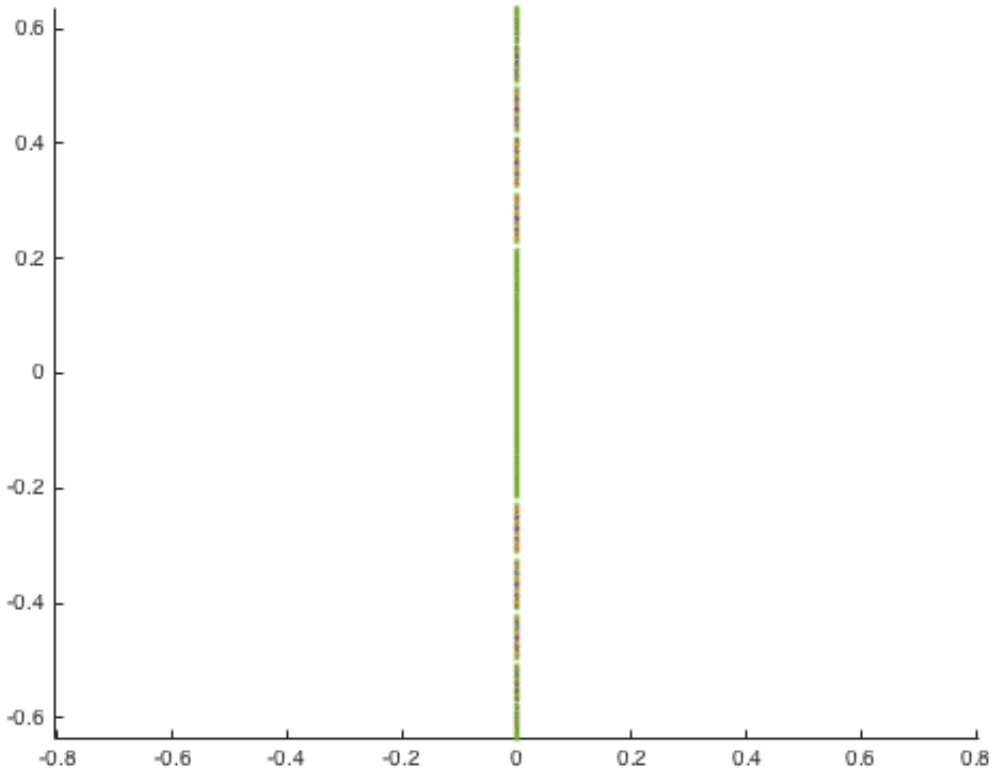

Figure 4.2: Eigenvalues of the matrix G, for $N = 20, 30, \ldots, 60$

integration method for all eigenvalues. This will mean that the discretization of $\frac{\partial}{\partial x} + \frac{1}{4}\frac{\partial^2}{\partial y^2}\Phi$ is $O(\Delta x^{-2})$. After discretizing the $\frac{\partial}{\partial t}$ term and multiplying by Δt, we have $\Delta t\left(\frac{\partial}{\partial x} + \frac{1}{4}\frac{\partial^2}{\partial y^2}\Phi\right) = O(\Delta t\Delta x^{-1} + \Delta t\Delta x^{-2}) = O(\Delta t\Delta x^{-2})$, so the stability condition that is required is $\Delta t = O(\Delta x^2)$.

For $\Delta t = O(\Delta x^2) = c_1\Delta x^2$, we want to find the eigenvalues of the discretization

of $\dfrac{\partial}{\partial x} + \dfrac{1}{4}\dfrac{\partial^2}{\partial y^2}\Phi$.

$$\Delta t \left[\frac{1}{\Delta x} \begin{pmatrix} A & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A \end{pmatrix} + \frac{1}{\Delta y^2} G \right] = c_1 \left[\Delta x \begin{pmatrix} A & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A \end{pmatrix} + G \right] = c_1 M$$

Note that M is a $N^2 \times N^2$ matrix.

Figure 4.3 is a plot of the eigenvalues of the matrix M when $N = 20, 30, \ldots, 60$.

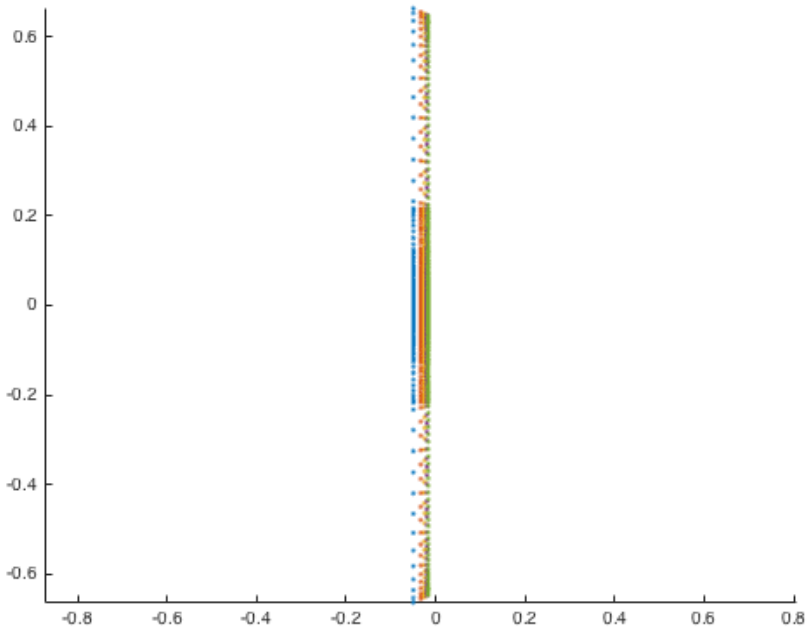

Figure 4.3: Eigenvalues of the matrix M for $N = 20, 30, \ldots, 60$

The first approximation of (4.2) had a stability condition of $\Delta t = O(\Delta x)$ and the second approximation had a stability condition of $\Delta t = O(\Delta x^2)$, which is a substantial increase in computational runtime.

When discretizing (4.7), it may be better to clear the denominators and implement the resulting Fourier symbol instead of implementing a convolution operator in the

approximation.

$$\omega = \xi_1 \left(1 + \frac{\xi_2^2}{2\xi_1^2}\right) = \xi_1 + \frac{1}{2}\frac{\xi_2^2}{\xi_1} \Longrightarrow 0 = (-i\omega)(i\xi_1) + (i\xi_1)^2 + \frac{(i\xi_2)^2}{2}.$$

In physical space, this can be written as

$$\left(\frac{\partial^2}{\partial t \partial x} + \frac{\partial^2}{\partial x^2} + \frac{1}{2}\frac{\partial^2}{\partial y^2}\right) u = 0.$$

In [6], this is shown to have a stability condition of $\Delta t = O(\Delta x)$. In this case, the second approximation does not suffer from $\Delta t = O(\Delta x^2)$, making this a more attractive option to use as an absorbing boundary condition.

4.2 The 2-D Linearized Water Wave Equation

We want to repeat the above analysis for the 2-D WWE. We know the 2-D WWE has the following Fourier symbol

$$-\omega^2 + \sqrt{\xi_1^2 + \xi_2^2} = 0. \tag{4.13}$$

We want to isolate and solve for ω in (4.13). Mirroring (2.21) in the 1-D WWE case to solve for ω, we have

$$\omega = \pm \mathrm{sgn}(\xi_1)|\xi_1|^{1/2} \left(1 + \frac{\xi_2^2}{\xi_1^2}\right)^{1/4}. \tag{4.14}$$

To develop a one-way equation to use in a boundary layer for the 2-D WWE, we consider expanding (4.14) about normal incidence, when $\xi_2 \ll \xi_1$.

4.2.1 Stability of first approximation for the 2-D WWE

For the right moving waves, a first approximation of (4.14) for $\dfrac{\xi_2}{\xi_1}$ near 0 is

$$\omega = \mathrm{sgn}(\xi_1)|\xi_1|^{1/2} \Longrightarrow -i\omega + \frac{i\xi_1}{|\xi_1|^{1/2}} = 0$$

which corresponds to the 1-D OWWWE

$$\left(\frac{\partial}{\partial t} + \frac{\partial}{\partial x}|D|^{-1/2}\right) u(x,y,t) = 0 \tag{4.15}$$

where

$$|D|^{-1/2}u(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{u(z)\,dz}{\sqrt{|x-z|}}.$$

Note that (4.15) is the OWWWE from Chapter II.

Consider using a linear upwind reconstruction to approximate $u(x)$ in the integral transform for (4.15). Integrating the singularity exactly would yield the dense $N \times N$ matrix

$$\frac{\partial}{\partial x}|D|^{-1/2}u \approx \frac{1}{\Delta x}(\sqrt{\Delta x}\widetilde{A})\overline{U} = \frac{1}{\sqrt{\Delta x}}\widetilde{A}\overline{U}$$

where $\overline{U}$ is the vector of cell averages in the x-direction of $u(\cdot, y, t)$.

For stable time integration, $\Delta t \lambda_k$ must be inside the stability region of the time integration method for all eigenvalues. For example, if we use Forward Euler to approximate (4.15), we find

$$\overline{U}^{n+1} = \overline{U}^n + \Delta t \frac{1}{\sqrt{\Delta x}}\widetilde{A}\overline{U}.$$

In particular, this implies a stability scaling $O(\Delta t \Delta x^{-1/2}) = O(\Delta t \Delta x^{-1/2})$ needs to remain bounded, thus $\Delta t = O(\sqrt{\Delta x})$.

Figure 4.4 shows the eigenvalues of $\widetilde{A}$, for $N = 20, 30, \ldots, 100$.

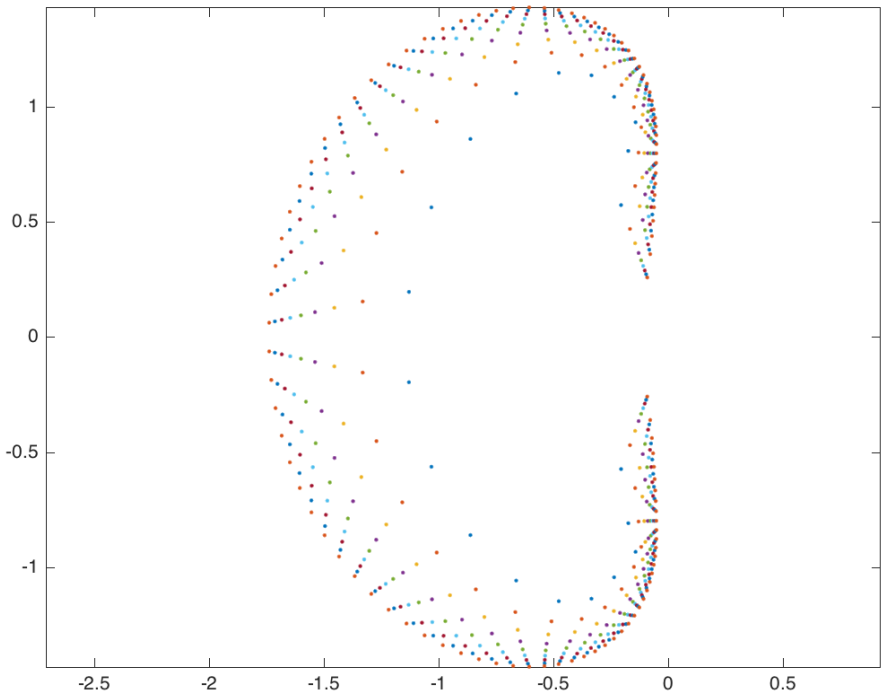

Figure 4.4: Eigenvalues of the matrix $\widetilde{A}$ for $N = 20, 30, \ldots, 100$

As the mesh is refined, the eigenvalues of $\widetilde{A}$ seem to fill in a curve, while the shape of the curve remains unchanged, which we loosely denote $\widetilde{A} = O(1)$.

4.2.2 Stability of second approximation for the 2-D WWE

A second order Taylor approximation in (4.14) yields

$$\omega = \operatorname{sgn}(\xi_1)|\xi_1|^{1/2}\left(1 + \frac{\xi_2^2}{4\xi_1^2}\right). \tag{4.16}$$

There are several ways to view this approximation of the Fourier symbol.

Convolution with kernel $|\xi_1|^{-5/2}$

Multiplying by i and rearranging terms, (4.16) becomes

$$\begin{aligned}
i\omega &= i\text{sgn}(\xi_1)|\xi_1|^{1/2} + i\text{sgn}(\xi_1)|\xi_1|^{1/2}\frac{\xi_2^2}{4\xi_1^2} \\
i\omega &= \frac{i\xi_1}{|\xi_1|^{1/2}} + \frac{i\xi_1}{|\xi_1|^{1/2}}\frac{\xi_2^2}{4\xi_1^2} \\
i\omega &= \frac{i\xi_1}{|\xi_1|^{1/2}} - \frac{i\xi_1}{4|\xi_1|^{5/2}}(i\xi_2)^2 \\
0 &= -i\omega + \frac{i\xi_1}{|\xi_1|^{1/2}} - \frac{i\xi_1}{4|\xi_1|^{5/2}}(i\xi_2)^2 \\
0 &- i\omega + \frac{i\xi_1}{|\xi_1|^{1/2}} + \frac{1}{3\sqrt{2\pi}}(i\xi_2)^2(i\xi_1)\left(-\frac{3\sqrt{2\pi}}{4|\xi_1|^{5/2}}\right)
\end{aligned}$$

The inverse Fourier transform of $-\dfrac{3\sqrt{2\pi}}{4|\xi_1|^{5/2}}$ is

$$\frac{1}{2\pi}P.V.\int_{-\infty}^{\infty}\left(-\frac{3\sqrt{2\pi}}{4|\xi_1|^{5/2}}\right)e^{ix\xi_1}\,d\xi_1 = |x|^{3/2}.$$

Thus, in physical space, (4.16) corresponds to

(4.17)
$$\frac{\partial}{\partial t}u(x,y,t) + \frac{\partial}{\partial x}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{u(z,y,t)\,dz}{\sqrt{|x-z|}} + \frac{\partial^2}{\partial y^2}\frac{\partial}{\partial x}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}u(z,y,t)\,dz = 0$$

where

$$|D|^{-5/2}u(x) = \frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}u(z)\,dz.$$

Eigenvalues of numerical approximation of $\partial_x|D|^{-5/2}$

Using a linear upwind reconstruction of the function $u(x,y,t)$ in the convolution integral, we obtain

$$\frac{\partial}{\partial x}\int_{-\infty}^{\infty}|x-z|^{3/2}u(z,y,t)\,dz \approx \Delta x^{3/2}\widetilde{B}\overline{U}.$$

Let $\overline{U}$ be the vector of cell averages in the x-direction of $u(\cdot,y,t)$.

First, we want to look at the eigenvalues of $\Delta x^{3/2}\widetilde{B}$, shown in Figure 4.5 for $N = 20, 30, \ldots, 100$.

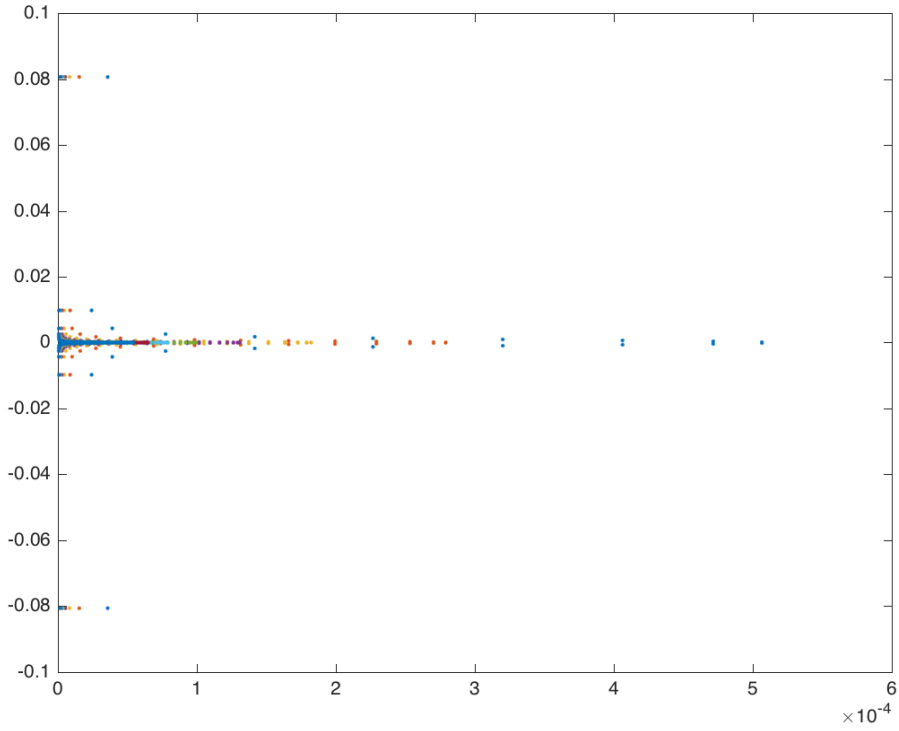

Figure 4.5: Eigenvalues of the matrix $\Delta x^{3/2}\widetilde{B}$ for $N = 20, 30, \ldots, 100$

As the mesh is refined, the eigenvalues of $\Delta x^{3/2}\widetilde{B}$ seem to fill in a curve, while the shape of the curve remains unchanged, which we loosely denote $\Delta x^{3/2}\widetilde{B} = O(1)$.

Eigenvalues of the discretized $\partial_y^2 \partial_x |D|^{-5/2}$ operator

By using centered-differencing, we apply the $\dfrac{\partial^2}{\partial y^2}$ term to the discretization of $\partial_x |D|^{-5/2}$ and obtain the following $N^2 \times N^2$ matrix

$$\frac{1}{\Delta y^2}\begin{pmatrix} -2\Delta x^{3/2}\widetilde{B} & \Delta x^{3/2}\widetilde{B} & 0 & \cdots & 0 \\ \Delta x^{3/2}\widetilde{B} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \Delta x^{3/2}\widetilde{B} \\ 0 & \cdots & 0 & \Delta x^{3/2}\widetilde{B} & -2\Delta x^{3/2}\widetilde{B} \end{pmatrix} \overline{\overline{U}} = \frac{1}{\Delta y^2}\widetilde{G}\overline{\overline{U}}$$

Figure 4.6 is a plot of the eigenvalues for the matrix $\widetilde{G}$ for $N = 20, 30, \ldots, 100$.

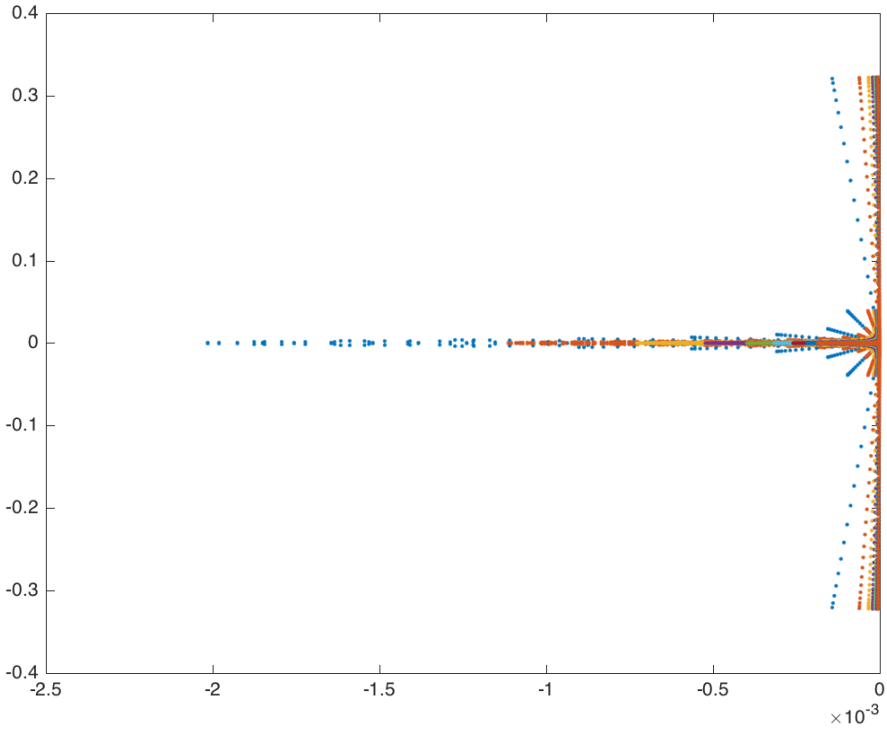

Figure 4.6: Eigenvalues of the matrix $\widetilde{G}$, for $N = 20, 30, \ldots, 100$

As the mesh is refined, the eigenvalues of $\widetilde{G}$ seem to fill in a curve, while the shape of the curve remains unchanged, which we loosely denote $\widetilde{G} = O(1)$ and

$$\frac{1}{\Delta y^2}\widetilde{G} = O(\Delta y^{-2}).$$

Discretizing the entire second approximation

To find a discretization for (4.17), we must discretize and combine $\frac{\partial}{\partial x}|D|^{-1/2}$ and $\frac{\partial^2}{\partial y^2}\frac{\partial}{\partial x}|D|^{-5/2}$:

$$\left(\frac{\partial}{\partial x}|D|^{-1/2} + \frac{\partial^2}{\partial y^2}\frac{\partial}{\partial x}|D|^{-5/2}\right) u(x,y,t) \approx \left[\frac{1}{\sqrt{\Delta x}}\begin{pmatrix} \widetilde{A} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \widetilde{A} \end{pmatrix} + \frac{1}{\Delta y^2}\widetilde{G}\right]\overline{\overline{U}}$$

If $\widetilde{A} = O(1)$ and $\widetilde{G} = O(1)$, then $\frac{1}{\sqrt{\Delta x}}\widetilde{A} = O(\Delta x^{-1/2})$ and $\frac{1}{\Delta y^2}\widetilde{G} = O(\Delta x^{-2})$. This will mean that the discretization of $\frac{\partial}{\partial x}|D|^{-1/2} + \frac{\partial^2}{\partial y^2}\frac{\partial}{\partial x}|D|^{-5/2}$ is $O(\Delta x^{-2})$. For stable time integration, $\Delta t\lambda_k$ must be inside the stability region of the time integration method for all eigenvalues. For example, if we use Forward Euler to approximate (4.17), we find

$$\overline{\overline{U}}^{n+1} = \overline{\overline{U}}^{n} + \Delta t\left[\frac{1}{\sqrt{\Delta x}}\begin{pmatrix} \widetilde{A} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \widetilde{A} \end{pmatrix} + \frac{1}{\Delta y^2}\widetilde{G}\right]\overline{\overline{U}}$$

In particular, this implies a stability scaling $O(\Delta t\Delta x^{-1/2} + \Delta t\Delta x^{-2})$ needs to remain bounded, thus $\Delta t = O(\Delta x^2)$. This is a substantial increase in scaling from (4.15), where Δt had a scaling of $O(\sqrt{\Delta x})$, which makes this implementation computationally unreasonable to use.

For $\Delta t = O(\Delta x^2) = c_1 \Delta x^2$, we want to find the eigenvalues of the $N^2 \times N^2$ matrix

$$\frac{\Delta t}{\sqrt{\Delta x}} \begin{pmatrix} \widetilde{A} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \widetilde{A} \end{pmatrix} + \frac{\Delta t}{\Delta y^2} \widetilde{G} = c_1 \left[\Delta x^{3/2} \begin{pmatrix} \widetilde{A} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \widetilde{A} \end{pmatrix} + \widetilde{G} \right] = c_1 \widetilde{M}$$

Figure 4.7 shows the eigenvalues of $\widetilde{M}$ when $N = 20, 30, \ldots, 100$.

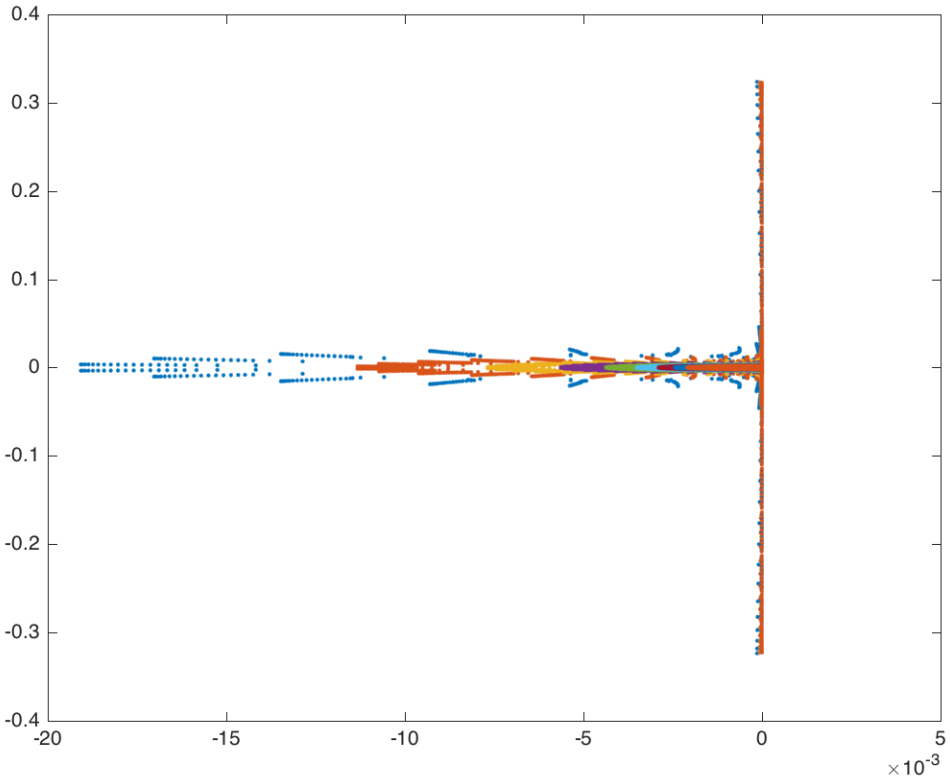

Figure 4.7: Eigenvalues of the matrix $\widetilde{M}$ for $N = 20, 30, \ldots, 100$

The first approximation of (4.14) had a stability condition of $\Delta t = O(\sqrt{\Delta x})$, while the second approximation, (4.17) had a stability condition of $\Delta t = O(\Delta x^2)$, which is a substantial increase in computational runtime. This scaling becomes prohibitively expensive to run numerically as N increases.

Convolution with two other kernels: $|\xi|^{-3/4}$ and $\mathbf{sgn}(\xi_1)|\xi_1|^{-3/4}$

Alternatively, we could view (4.14) as repeated convolution with two different kernels.

$$\omega = \frac{\xi_1}{|\xi_1|^{1/2}} - \frac{(i\xi_2)^2}{4}\left(\frac{\mathrm{sgn}(\xi_1)}{|\xi_1|^{3/4}}\right)\left(\frac{1}{|\xi_1|^{3/4}}\right). \tag{4.18}$$

Note that the inverse Fourier transform of $\dfrac{\mathrm{sgn}(\xi_1)}{i|\xi_1|^{3/4}}$ is

$$\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{\mathrm{sgn}(\xi)}{i|\xi|^{3/4}}e^{ix\xi}\,d\xi = \frac{\Gamma(\frac{1}{4})\sin(\frac{\pi}{8})}{\pi}\frac{\mathrm{sgn}(x)}{|x|^{1/4}}$$

and the inverse Fourier transform of $\dfrac{1}{|\xi_1|^{3/4}}$ is

$$\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{1}{|\xi|^{3/4}}e^{ix\xi}\,d\xi = \frac{\Gamma(\frac{1}{4})\cos(\frac{\pi}{8})}{\pi}\frac{1}{|x|^{1/4}}$$

where $\Gamma(z) = \displaystyle\int_0^{\infty} t^{z-1}e^{-t}\,dt$ is the Gamma function.

Multiplying (4.18) by i and rearranging terms, the second approximation can be written as

$$-i\omega + \frac{i\xi_1}{|\xi_1|^{1/2}} + \frac{(i\xi_2)^2}{4}\left(\frac{\mathrm{sgn}(\xi_1)}{i|\xi_1|^{3/4}}\right)\left(\frac{1}{|\xi_1|^{3/4}}\right) = 0.$$

In physical space, this corresponds to

$$\frac{\partial}{\partial t}u(x,y,t) + \frac{\partial}{\partial x}|D|^{-1/2}u(x,y,t) + \frac{1}{4}\frac{\partial^2}{\partial y^2}\Phi_1|D|^{-3/4}u(x,y,t) = 0 \tag{4.19}$$

where

$$\Phi_1 u(x) = \frac{\Gamma(\frac{1}{4})\sin(\frac{\pi}{8})}{\pi}\int_{-\infty}^{\infty}\frac{\mathrm{sgn}(x-z)u(z)\,dz}{|x-z|^{1/4}}$$

and

$$|D|^{-3/4}u(x) = \frac{\Gamma(\frac{1}{4})\cos(\frac{\pi}{8})}{\pi}\int_{-\infty}^{\infty}\frac{u(z)\,dz}{|x-z|^{1/4}}.$$

Eigenvalues of approximation of product of convolution operators

We look at the eigenvalues of a discretization of $\frac{1}{4}\frac{\partial^2}{\partial y^2}\Phi_1|D|^{-3/4}u(x,y,t)$. Using a linear upwind reconstruction of the function $u(x,y,t)$ in the convolution integral, we obtain the $N \times N$ matrix

$$|D|^{-3/4}u(x) = \frac{\Gamma(\frac{1}{4})\cos(\frac{\pi}{8})}{\pi}\int_{-\infty}^{\infty}\frac{u(z)\,dz}{|x-z|^{1/4}} \approx \Delta x^{3/4}\widetilde{B_1}\overline{U}$$

where $\overline{U}$ is the vector of cell averages in the x-direction of $u(\cdot,y,t)$.

Separately, we look at an approximation of $\Phi_1 u(x)$ using a linear reconstruction to approximate the convolution integral.

$$\Phi_1 u(x) = \frac{\Gamma(\frac{1}{4})\sin(\frac{\pi}{8})}{\pi}\int_{-\infty}^{\infty}\frac{\mathrm{sgn}(x-z)u(z)\,dz}{|x-z|^{1/4}} \approx \Delta x^{3/4}\widetilde{B_2}\overline{U}.$$

Combining these two approximations together and applying repeated convolutions, the operator $\Phi|D|^{-3/4}u(x,y,t)$ is approximated by the $N \times N$ matrix.

$$\Phi_1|D|^{-3/4}u(x,y,t) \approx \Delta x^{3/4}\widetilde{B_2}\Delta x^{3/4}\widetilde{B_1}\overline{U} = \Delta^{3/2}\widetilde{C}\overline{U}.$$

First, we want to look at the eigenvalues of $\Delta x^{3/2}\widetilde{C}$. Figure 4.8 shows the eigenvalues for $\Delta x^{3/2}\widetilde{C}$, for $N = 20, 30, \ldots, 100$.

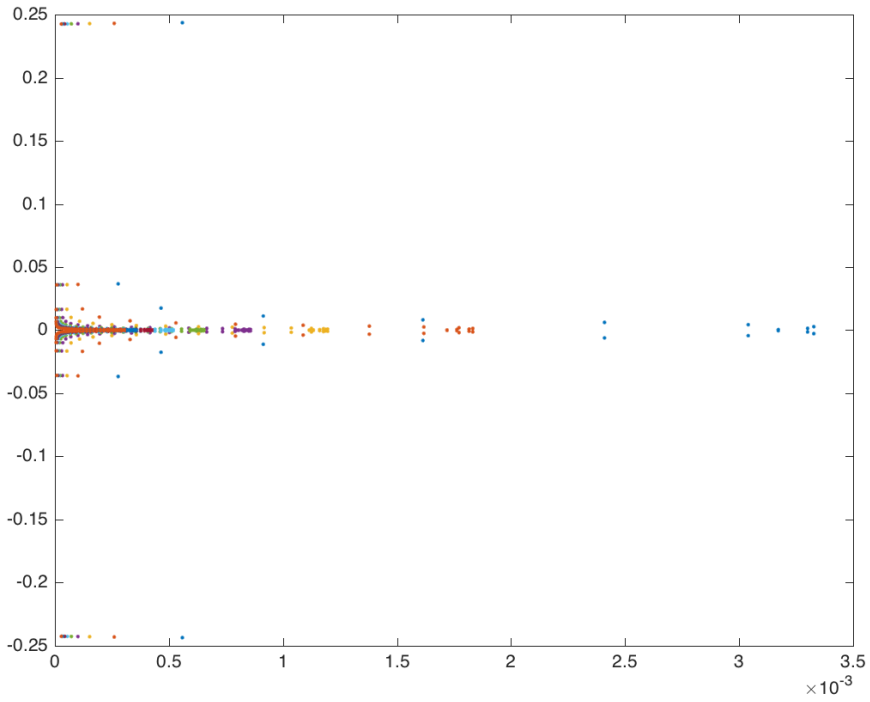

Figure 4.8: Eigenvalues of the matrix $\Delta x^{3/2}\widetilde{C}$, for $N = 20, 30, \ldots, 100$

As the mesh is refined, the eigenvalues of $\Delta x^{3/2}\widetilde{C}$ seem to fill in a curve, while the shape of the curve remains unchanged, which we loosely denote $\Delta x^{3/2}\widetilde{C} = O(1)$.

Discretizing the $\partial_y^2 \Phi |D|^{-3/4}$ operator

By using centered-differencing, we apply the $\dfrac{\partial^2}{\partial y^2}$ term to the discretization of $\Phi|D|^{-3/4}$ to create the following matrix

$$\frac{1}{\Delta y^2}\begin{pmatrix} -2\Delta x^{3/2}\widetilde{C} & \Delta x^{3/2}\widetilde{C} & 0 & \cdots & 0 \\ \Delta x^{3/2}\widetilde{C} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \Delta x^{3/2}\widetilde{C} \\ 0 & \cdots & 0 & \Delta x^{3/2}\widetilde{C} & -2\Delta x^{3/2}\widetilde{C} \end{pmatrix} \overline{\overline{U}} = \frac{1}{\Delta y^2}\widetilde{J}\overline{\overline{U}}$$

where $\overline{\overline{U}}$ is the vector of solutions on our grid.

Figure 4.9 is a plot of the eigenvalues for the matrix $\widetilde{J}$ for $N = 20, 30, \ldots, 100$.

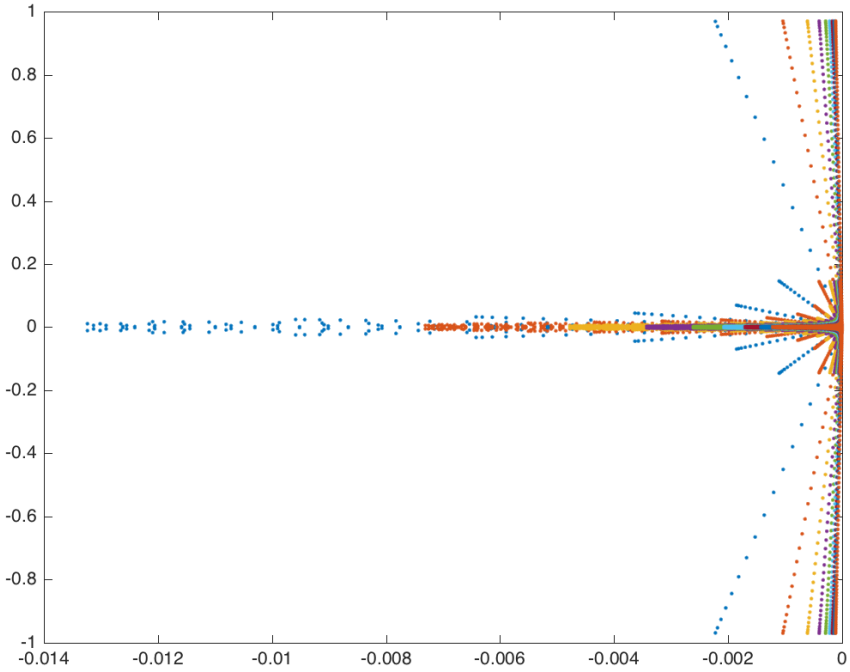

Figure 4.9: Eigenvalues of the matrix $\widetilde{J}$, for $N = 20, 30, \ldots, 100$

As the mesh is refined, the eigenvalues of $\widetilde{J}$ seem to fill in a curve, while the shape of the curve remains unchanged, which we loosely denote $\widetilde{J} = O(1)$ and $\dfrac{1}{\Delta y^2}\widetilde{J} = O(\Delta y^{-2})$.

Discretizing the entire second approximation

To find a discretization for (4.19), we must discretize and combine $\dfrac{\partial}{\partial x}|D|^{-1/2}$ and $\dfrac{1}{4}\dfrac{\partial^2}{\partial y^2}\Phi_1|D|^{-3/4}$.

$$\left(\frac{\partial}{\partial x}|D|^{-1/2} + \frac{1}{4}\frac{\partial^2}{\partial y^2}\Phi_1|D|^{-3/4}\right)u(x,y,t) \approx \left[\frac{1}{\sqrt{\Delta x}}\begin{pmatrix} \widetilde{A} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \widetilde{A} \end{pmatrix} + \frac{1}{\Delta y^2}\widetilde{J}\right]\overline{\overline{U}}$$

If $\widetilde{A} = O(1)$ and $\widetilde{J} = O(1)$, then $\frac{1}{\sqrt{\Delta x}}\widetilde{A} = O(\Delta x^{-1/2})$ and $\frac{1}{\Delta y^2}\widetilde{J} = O(\Delta x^{-2})$. This will mean that the discretization of $\frac{\partial}{\partial x}|D|^{-1/2} + \frac{1}{4}\frac{\partial^2}{\partial y^2}\Phi_1|D|^{-3/4}$ is $O(\Delta x^{-2})$. For stable time integration, $\Delta t\lambda_k$ must be inside the stability region of the time integration method for all eigenvalues. For example, if we use Forward Euler to approximate (4.19), we find

$$\overline{\overline{U}}^{n+1} = \overline{\overline{U}}^{n} + \Delta t \left[\frac{1}{\sqrt{\Delta x}} \begin{pmatrix} \widetilde{A} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \widetilde{A} \end{pmatrix} + \frac{1}{\Delta y^2}\widetilde{J} \right] \overline{\overline{U}}$$

In particular, this implies a stability scaling $O(\Delta t\Delta x^{-1/2} + \Delta t\Delta x^{-2})$ needs to remain bounded, thus $\Delta t = O(\Delta x^2)$. This is a substantial increase in scaling from (4.15), where Δt had a scaling of $O(\sqrt{\Delta x})$, which makes this implementation computationally unreasonable to use.

For $\Delta t = O(\Delta x^2) = c_1\Delta x^2$, we want to find the eigenvalues of the $N^2 \times N^2$ matrix

$$\Delta t \left[\frac{1}{\sqrt{\Delta x}} \begin{pmatrix} \widetilde{A} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \widetilde{A} \end{pmatrix} + \frac{1}{\Delta y^2}\widetilde{J} \right] = c_1 \left[\Delta x^{3/2} \begin{pmatrix} \widetilde{A} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \widetilde{A} \end{pmatrix} + \widetilde{J} \right] = c_1\widetilde{K}$$

Figure 4.10 shows the eigenvalues of $\widetilde{K}$ when $N = 20, 30, \ldots, 100$.

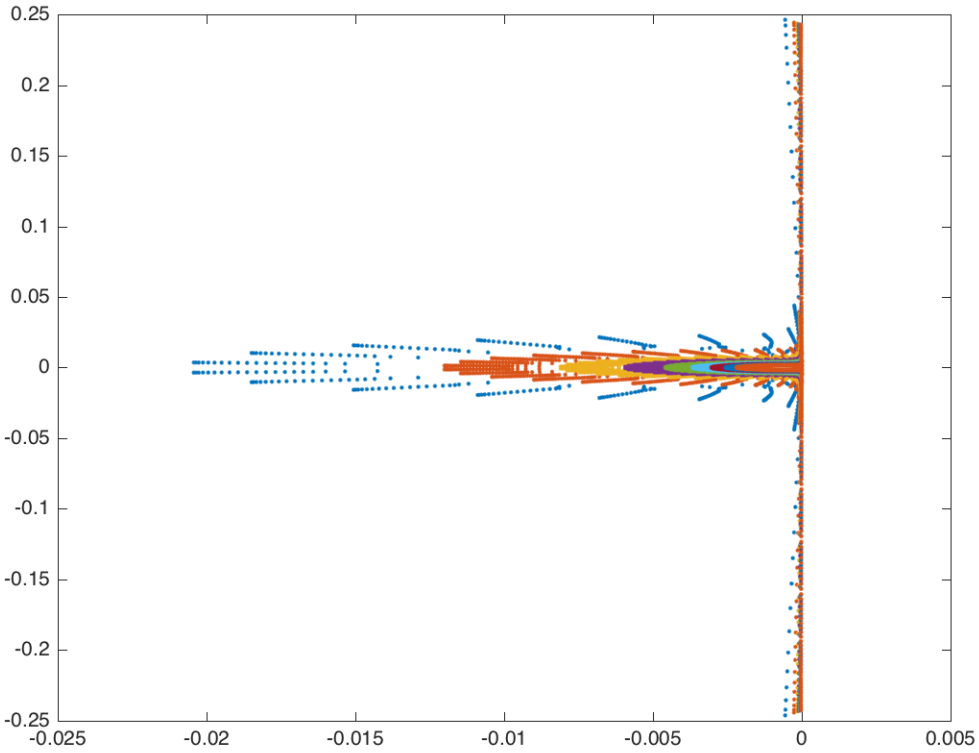

Figure 4.10: Eigenvalues of the matrix $\widetilde{K}$ for $N = 20, 30, \ldots, 100$

We again observe that the second order approximation for (4.14) has a stability condition of $\Delta t = O(\Delta x^2)$. Compared to the scaling in (4.15), $\Delta t = O(\sqrt{\Delta x})$, this is prohibitively expensive to implement and run.

When considering how to implement (4.14), it may be better to clear the denominators and implement the resulting Fourier symbol as opposed to working with these very singular kernels.

If the denominator in (4.14) is cleared, the resulting Fourier symbol is

$$\omega = \frac{\xi_1}{|\xi_1|^{1/2}} + \frac{1}{|\xi_1|^{1/2}} \frac{\xi_2^2}{4\xi_1} \Longrightarrow 0 = (-i\omega)(i\xi_1) + \frac{(i\xi_1)^2}{|\xi_1|^{1/2}} + \frac{(i\xi_2)^2}{4|\xi_1|^{1/2}}.$$

In physical space, this can be written as

$$\left(\frac{\partial^2}{\partial t \partial x} + \frac{\partial^2}{\partial x^2}|D|^{-1/2} + \frac{1}{4}\frac{\partial^2}{\partial y^2}|D|^{-1/2}\right) u = 0.$$

4.3 Using (4.15) as an absorbing boundary condition for the 2-D WWE

Similar to how the OWWWE was implemented for the 1-D WWE, we implement (4.15) in an absorbing boundary layer to see how effective it is at absorbing outgoing waves as they reach the boundary.

Figure 4.11 shows the solution of the 2-D WWE at various times T with (4.15) implemented in an absorbing boundary layer around the bulk domain. We note that the appropriate version of (4.15) is used for the appropriate direction of wave propagation. The initial conditions in Figure 4.11 are given in (3.13)–(3.14).

In comparison with Figure 3.1, Figure 4.11 does an excellent job at absorbing outgoing waves for the same initial conditions. Though there are some reflections from the outgoing waves hitting the boundary, they are significantly smaller compared to the initial size of the wave and the size of the reflections in Figure 3.1.

We also show the solution of the 2-D WWE with (4.15) given (3.15)–(3.16), an initial condition corresponding to 1-D wave propagation in the x-direction. The expectation here is that implementing (4.15) in a boundary layer will help reduce waves off both the x and y boundaries.

In comparison with Figure 3.2, Figure 4.12 does an excellent job at absorbing outgoing waves for the same initial conditions. Though there are some reflections from the outgoing waves hitting the boundary, they are significantly smaller compared to the initial size of the wave and the size of the reflections in Figure 3.2. We note that there are still reflections off the y boundary, though they seem to be slightly reduced in magnitude.

4.4 Using (4.17) as an absorbing boundary condition for the 2-D WWE

We implement (4.17) in an absorbing boundary layer to see how effective it is at absorbing outgoing waves as they reach the boundary.

Figure 4.13 shows the solution of the 2-D WWE at various times T with (4.17) implemented in an absorbing boundary layer around the bulk domain. We note that the appropriate version of (4.17) is used for the appropriate direction of wave propagation. The initial conditions in Figure 4.13 are given in (3.13)–(3.14).

We note that based on Figure 4.13, it does not appear as though (4.17) is properly allowing waves to enter the boundary layer that it is implemented in and is causing numerical reflections at the interface between the interior domain and the absorbing boundary layer, making this method ineffective.

We also show the solution of the 2-D WWE with (4.17) for 1-D data (3.15)–(3.16). The expectation here is that implementing (4.17) in a boundary layer will do a better job absorbing outgoing waves approaching the boundary than (4.15) does.

We again see that based on Figure 4.14, it does not appear as though (4.17) is properly allowing waves to enter the boundary layer that it is implemented in and is causing numerical reflections at the interface between the interior domain and the absorbing boundary layer, making this method ineffective.

4.5 Using (4.19) as an absorbing boundary condition for the 2-D WWE

We implement (4.19) in an absorbing boundary layer to see how effective it is at absorbing outgoing waves as they reach the boundary.

Figure 4.15 shows the solution of the 2-D WWE at various times T with (4.19) implemented in an absorbing boundary layer around the bulk domain. We note that the appropriate version of (4.19) is used for the appropriate direction of wave

propagation. The initial conditions in Figure 4.15 are given in (3.13)–(3.14).

We note that based on Figure 4.15, it does not appear as though (4.19) is properly allowing waves to enter the boundary layer that it is implemented in and is causing numerical reflections at the interface between the interior domain and the absorbing boundary layer, making this method ineffective.

We also show the solution of the 2-D WWE with (4.19) for 1-D data (3.15)–(3.16). The expectation here is that implementing (4.19) in a boundary layer will do a better job absorbing outgoing waves approaching the boundary than (4.15) does.

We again see that based on Figure 4.16, it does not appear as though (4.19) is properly allowing waves to enter the boundary layer that it is implemented in and is causing numerical reflections at the interface between the interior domain and the absorbing boundary layer, making this method ineffective.

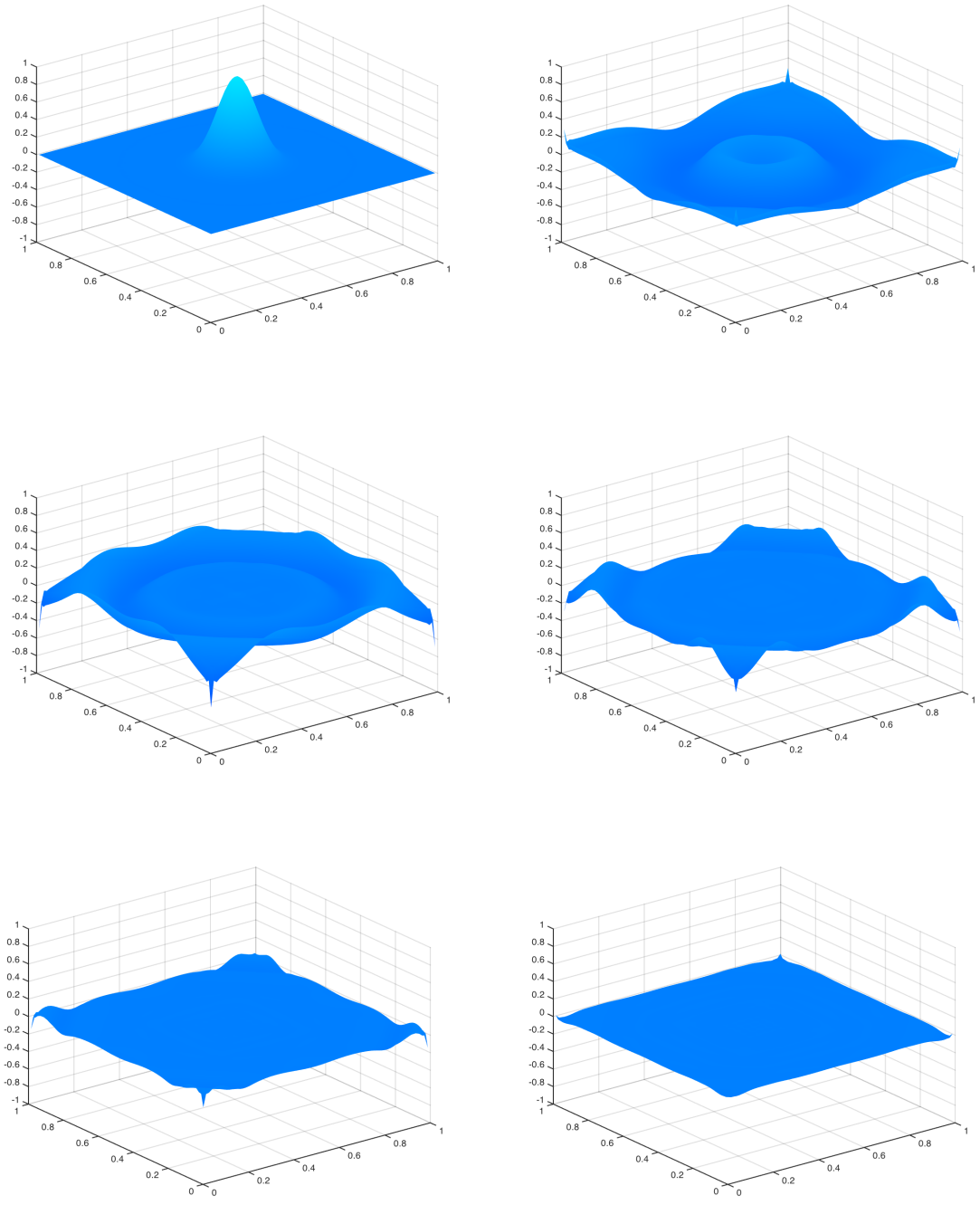

Figure 4.11: Solution $u(x, y, T)$ for the 2-D WWE with (4.15) implemented in a boundary layer for $T = 0$ (top left), $T \approx 2$ (top right), $T \approx 4$ (middle left), $T \approx 6$ (middle right), $T \approx 8$ (bottom left), $T = 10$ (bottom right)

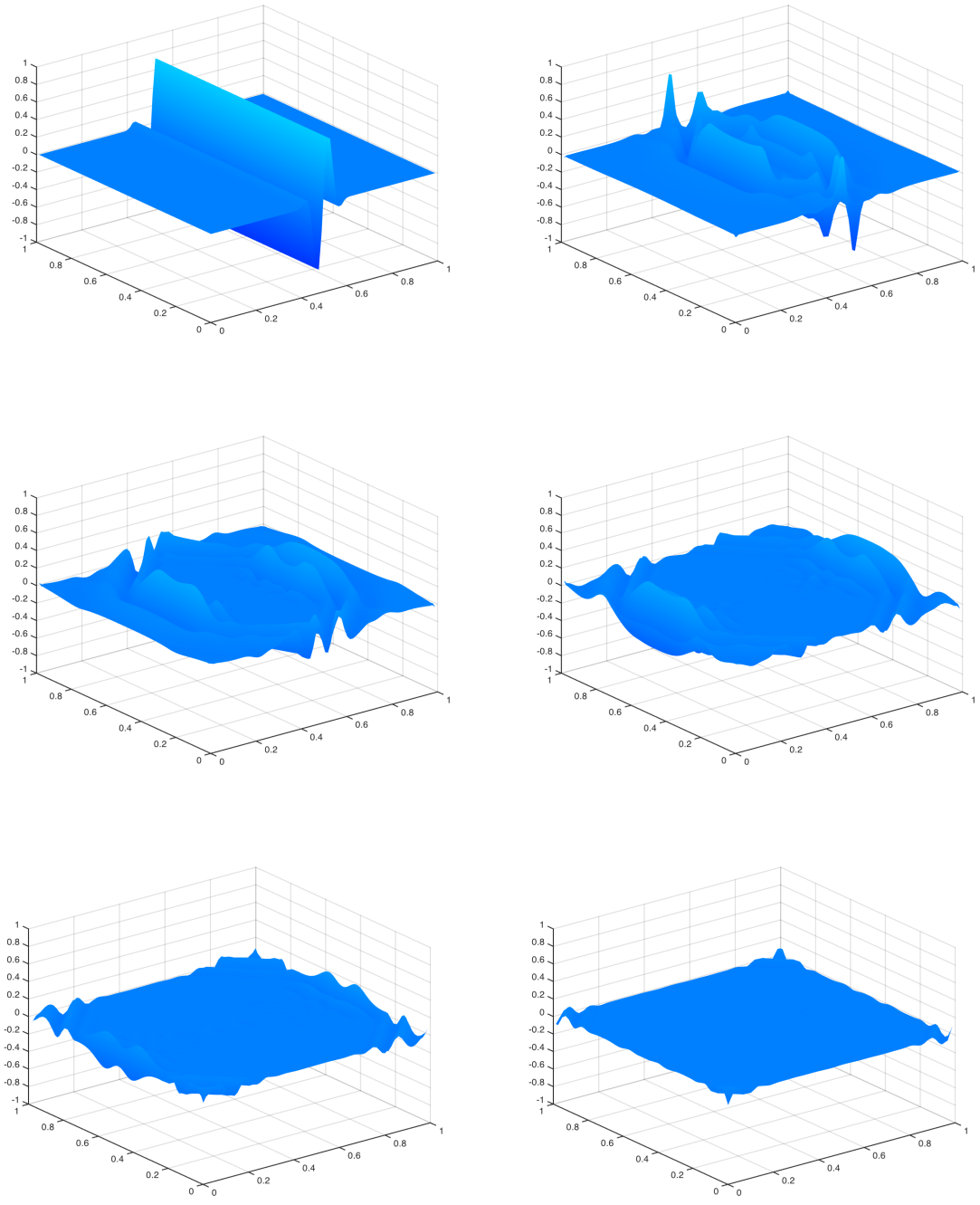

Figure 4.12: Solution $u(x, y, T)$ for the 2-D WWE with (4.15) implemented in a boundary layer for $T = 0$ (top left), $T \approx 2$ (top right), $T \approx 4$ (middle left), $T \approx 6$ (middle right), $T \approx 8$ (bottom left), $T = 10$ (bottom right)

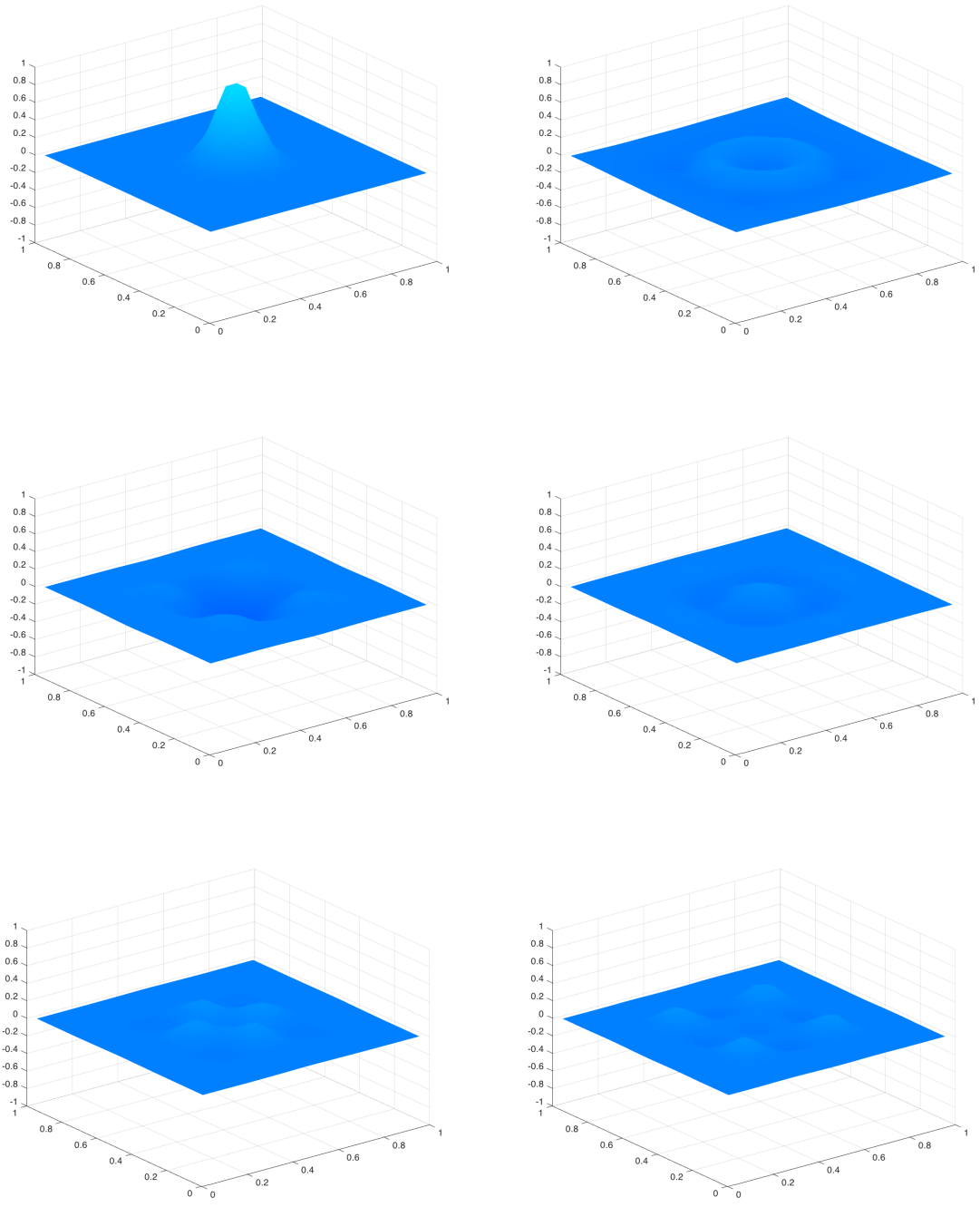

Figure 4.13: Solution $u(x, y, T)$ for the 2-D WWE with (4.17) implemented in a boundary layer for $T = 0$ (top left), $T \approx 2$ (top right), $T \approx 4$ (middle left), $T \approx 6$ (middle right), $T \approx 8$ (bottom left), $T = 10$ (bottom right)

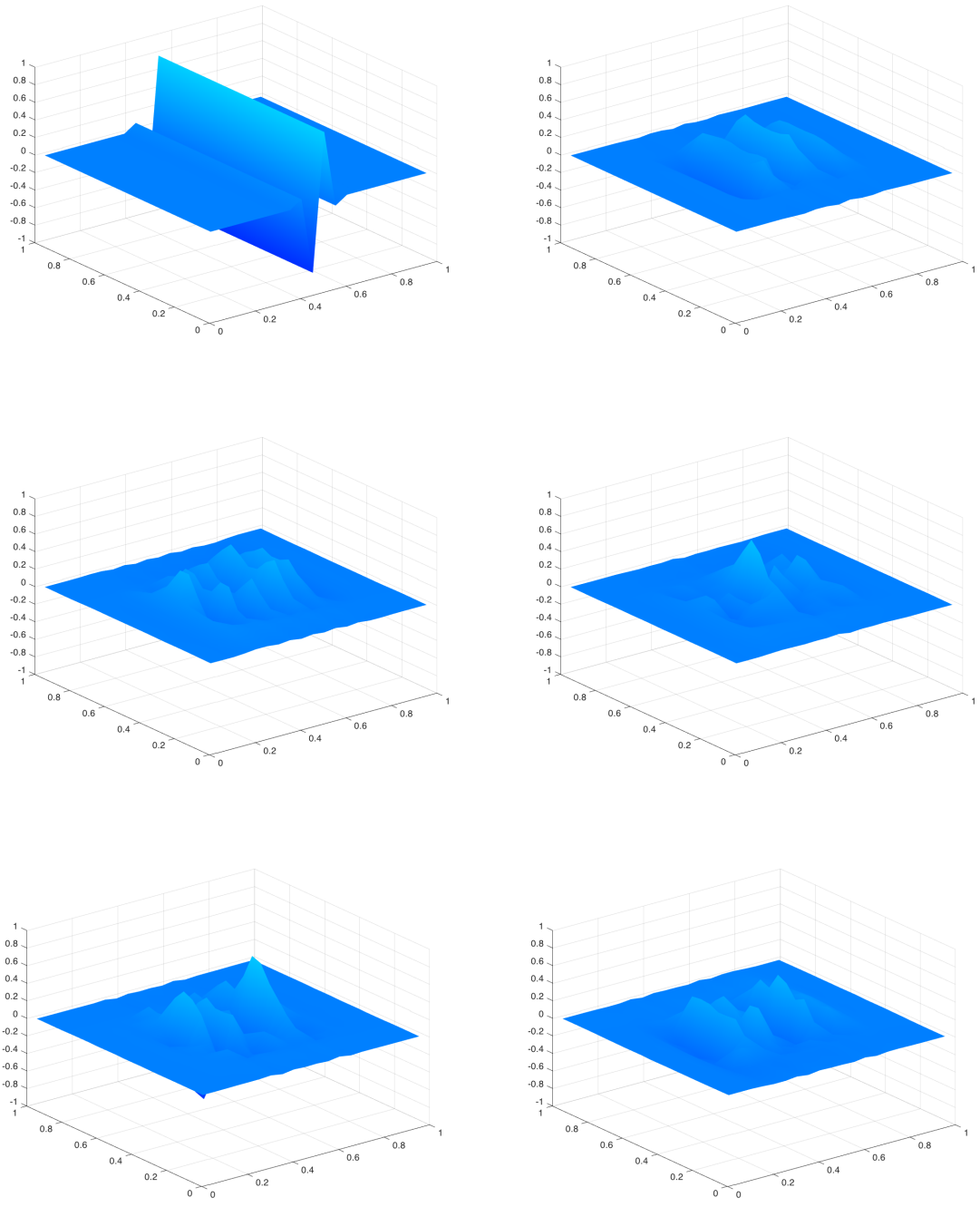

Figure 4.14: Solution $u(x, y, T)$ for the 2-D WWE with (4.17) implemented in a boundary layer for $T = 0$ (top left), $T \approx 2$ (top right), $T \approx 4$ (middle left), $T \approx 6$ (middle right), $T \approx 8$ (bottom left), $T = 10$ (bottom right)

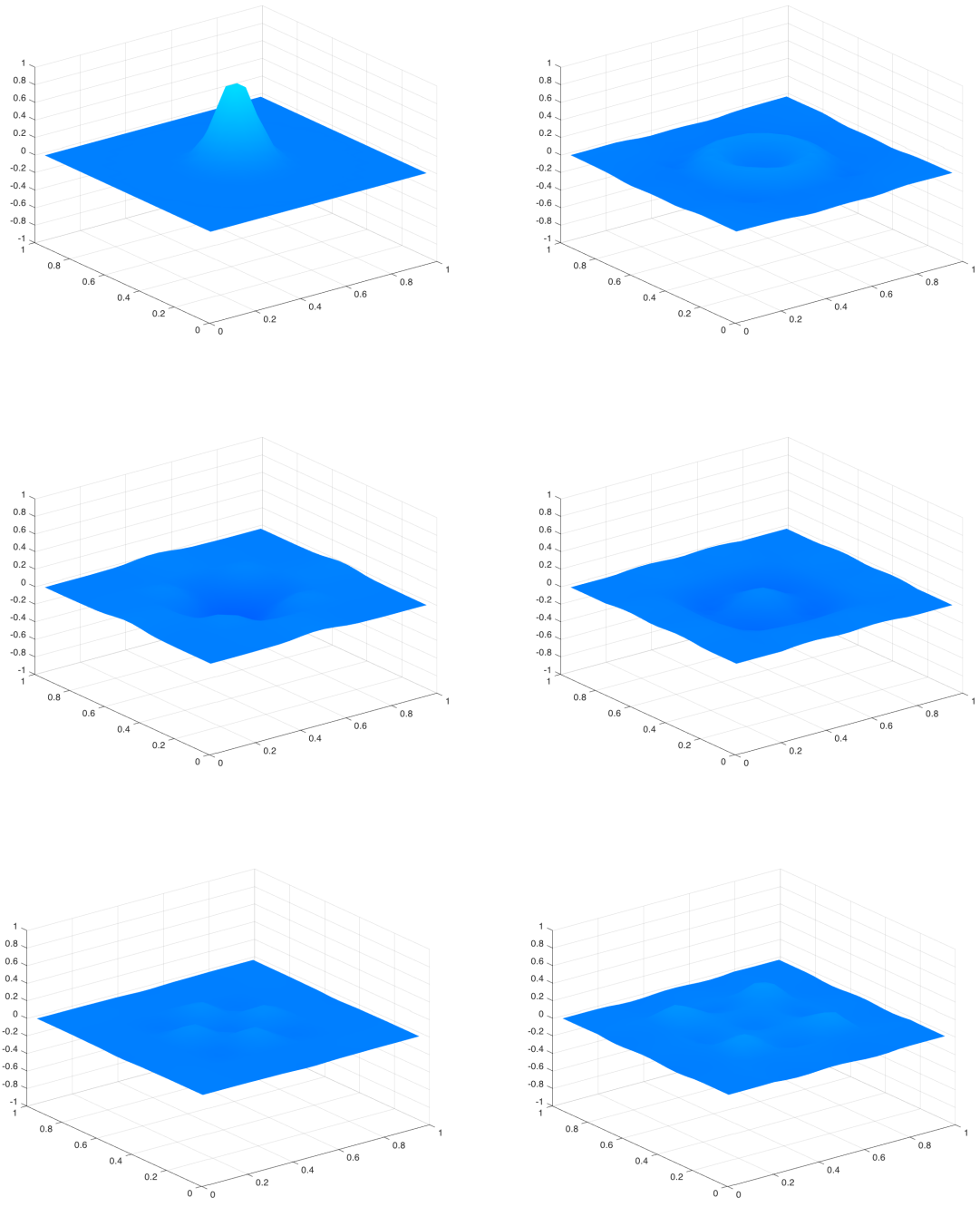

Figure 4.15: Solution $u(x, y, T)$ for the 2-D WWE with (4.19) implemented in a boundary layer for $T = 0$ (top left), $T \approx 2$ (top right), $T \approx 4$ (middle left), $T \approx 6$ (middle right), $T \approx 8$ (bottom left), $T = 10$ (bottom right)

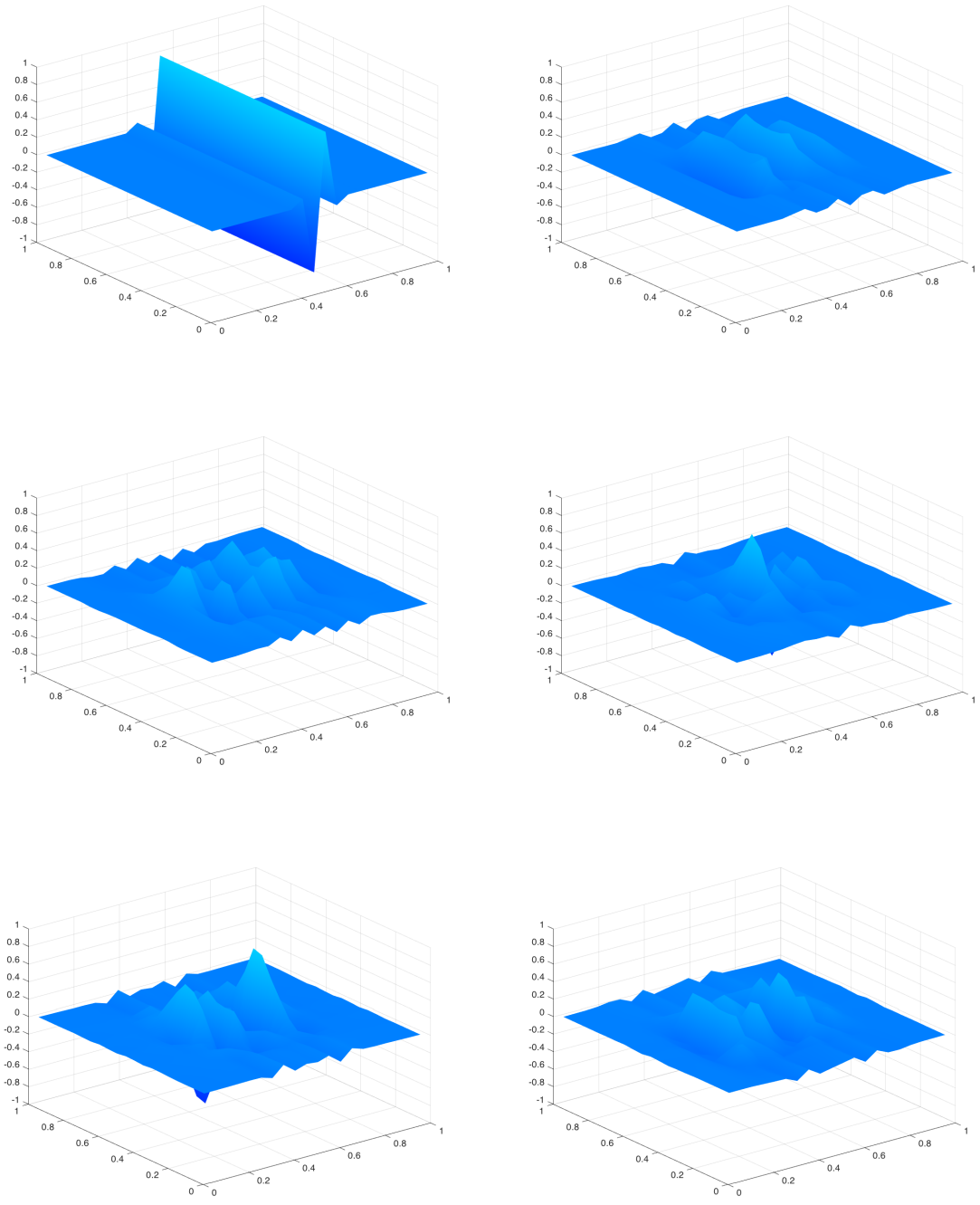

Figure 4.16: Solution $u(x, y, T)$ for the 2-D WWE with (4.19) implemented in a boundary layer for $T = 0$ (top left), $T \approx 2$ (top right), $T \approx 4$ (middle left), $T \approx 6$ (middle right), $T \approx 8$ (bottom left), $T = 10$ (bottom right)

CHAPTER V

Conclusions and Future Work

In this dissertation, we reviewed how absorbing boundary conditions were implemented for the 1-D WWE. We used these results to help us derive absorbing boundary conditions for the 2-D WWE. Although they are not perfectly absorbing boundary conditions like the OWWWE is for the 1-D WWE, we have found boundary conditions that significantly reduce the amount of numerical reflections off the boundary from outgoing waves. Some of the approximations can use improvement, particularly when dealing with the interface between domain and layer, but at least one approximation significantly reduces wave reflection.

There are still many directions for future work, some of which are listed here.

5.1 Higher order approximations

The Fourier symbol of the 2-D WWE, given in (4.13), can be approximated to a higher order to create more equations for use in an absorbing boundary condition. In theory, the higher order approximation we take, the more absorbing the boundary layer becomes. However, in Chapter IV, we saw that if the absorbing boundary conditions are not implemented properly, the interface between the layer and the domain may cause numerical reflections to occur. We also note that different approximation methods (e.g. Taylor or Padé, etc.) may give different methods. Because of this,

even if the approximation is not strictly "higher" order than one detailed here, a different approximation of (4.13) may lead to vastly different implementations.

One resulting question is if (4.17) and (4.19) can be rectified to produce a reasonable numerical method. Not only is the scaling $\Delta t = O(\Delta x^2)$ unreasonable to use, the equations do not properly behave at the interface between boundary layer and domain.

5.2 Higher accuracy approximations

Throughout the dissertation, many of the infinite integrals were approximated using a midpoint quadrature rule. In order to gain higher accuracy for the numerical method, we want to approximate these integrals using a higher order quadrature. In addition to gaining accuracy from better approximation of infinite integrals, we can also consider using a higher order approximation of the function $u(x, y, t)$ in the convolution integrals, similar to the method used in Chapter II with the OWWWE and the 1-D WWE. Using a higher order approximation in several variables for $u(x, y, t)$ may lead to increased accuracy.

5.3 Computational runtime

One thing to consider is to implement the discretization of convolution integrals using a Fast Fourier Transform (FFT). Recall that for functions $f(x), g(x) \in L^1(\mathbb{R})$, we know that

$$f * g(x) = \mathcal{F}^{-1}\left(\widehat{f}(\xi)\widehat{g}(\xi)\right)(x).$$

Thus, if we want to implement a convolution integral, we can consider using a FFT, which will decrease the computational complexity and also offer a numerical speedup from using the FFT for multiplication.

We also can improve the numerical method for the 2-D WWE (see Appendix C) by using a FFT to speed up the matrix multiplication. Although we have a nice structure for the matrix used in the 2-D WWE (it is a block Toeplitz matrix), the multiplication will still be $O(N^4)$. If an FFT can improve this matrix multiplication to $O(N^2 \log(N))$, this can be a tremendous speedup. Another way to implement this is to use a treecode or fast multipole method to approximate the convolution integrals.

5.4 Different geometries

When solving the 2-D WWE, only uniform grids in 2-D were considered. If one extended solving the 2-D WWE to nonuniform grids that stretch near the boundary, the 2-D WWE could be solved on a larger domain. Additionally, one could consider using an unstructured mesh to solve the 2-D WWE.

5.4.1 Radially Symmetric Water Wave Equations in 2-D

Suppose $u(x, y, t)$ is a solution to (3.5)–(3.7) that depends only on the distance from the origin, $r = \sqrt{x^2 + y^2}$. The goal is to rewrite (3.5)–(3.7) in a form that can be applied to radially symmetric functions. However, the operator $|D|$ is not obviously converted to a radially symmetric form.

Note the following identity in two spatial dimensions

$$|D|u = \frac{-\Delta}{|D|}u = -\text{div grad}\left(\frac{1}{2\pi|x|} * u\right) = -\text{div}\left(\frac{1}{2\pi|x|}*\right)\text{grad } u.$$

Recall that for a function $u(x, y, t) = u(r, t)$, grad $u = u_r\hat{r}$. Given an initial position and velocity that is dependent only on r, the radially symmetric formulation of the

2-D WWE can be written as

$$u_{tt}(r,t) - \frac{1}{2\pi}\frac{1}{r}\frac{\partial}{\partial r}\left(r\left(\frac{1}{|x|} * u_r\right)\right) = 0 \tag{5.1}$$

$$u(r,0) = u_0(r) \tag{5.2}$$

$$u_t(r,0) = u_1(r) \tag{5.3}$$

We know

$$\frac{1}{|x|} * u_r\hat{r} = \iint_{\mathbb{R}^2} \frac{1}{|x-y|} u_r(y)\frac{y}{|y|}dy. \tag{5.4}$$

The contribution from $|y| = \rho$ in (5.4) is

$$u_r(\rho)\int_{|y|=\rho} \frac{1}{|x-y|}\frac{y}{|y|}d\sigma(y) = u_r(\rho)\int_{|y|=1} \frac{1}{|x/\rho - y|}\frac{y}{|y|}d\sigma(y). \tag{5.5}$$

Define $b : \mathbb{R}^+ \to \mathbb{R}$

$$b(|x|)\frac{x}{|x|} = \int_{|y|=1} \frac{1}{|x-y|}\frac{y}{|y|}d\sigma(y). \tag{5.6}$$

Summing (5.5) on ρ turns (5.4) into

$$\int_0^\infty \rho u_r(\rho) b(|x|/\rho)\frac{d\rho}{\rho}. \tag{5.7}$$

Using the change of variables $\rho = e^s$ and $s = \ln(p)$, (5.4) can be written as

$$\int_{-\infty}^\infty e^{r-s}u_r(e^{r-s})b(e^s)\,ds. \tag{5.8}$$

Putting (5.8) into (5.1), we can write (5.1) explicitly as

$$u_{tt}(r,t) - \frac{1}{2\pi r}\frac{\partial}{\partial r}\left(r\int_{-\infty}^\infty e^{r-s}u_r(e^{r-s},t)b(e^s)\,ds\right) = 0 \tag{5.9}$$

where $b(s)$ is defined as

$$b(s) = \int_0^{2\pi} \frac{\cos\theta\,d\theta}{((s-\cos\theta)^2 + \sin^2\theta)^{1/2}}. \tag{5.10}$$

Note that (5.10) can be rewritten as a combination of complete elliptic integrals of the first and second kind:

$$b(s) = \frac{(s^2+1)K(-4s) + (s^2+1)K(4s) - E(-4s) - E(4s)}{s}$$

where

$$K(m) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - m\sin^2(\theta)}}$$
$$E(m) = \int_0^{\pi/2} \sqrt{1 - m\sin^2(\theta)}d\theta$$

Implementing a numerical method for (5.1) could lead to better insight as to how waves reflect off the boundaries. This could also be implemented in the 2-D WWE for use in an absorbing layer. One question that will arise from creating a numerical method for the radially symmetric 2-D WWE is whether or not there will be a need to derive absorbing boundary layers. If there is a need, deriving and implementing them would be another future endeavor. If not, analyzing why there isn't a need for an absorbing boundary layer might lead to whether this equation is an absorbing boundary condition for the 2-D WWE.

5.5 2-D Fourier Transform

In Chapter IV, we found an equation to use in an absorbing boundary layer by treating ξ_1 and ξ_2 separately. We could instead consider ξ_1 and ξ_2 to be part of a 2-D Fourier symbol. Recall the approximation for right moving waves is

$$i\omega = i\text{sgn}(\xi_1)\sqrt{|\xi_1|}\left(1 + \frac{\xi_2^2}{4\xi_1^2}\right) = \frac{i\xi_1}{\sqrt{|\xi_1|}} + \left(\frac{i\xi_1}{\sqrt{|\xi_1|}}\right)\left(\frac{\xi_2^2}{4\xi_1^2}\right).$$

Instead of breaking up $\frac{\xi_2^2}{4\xi_1^2}$ into a product, we view this as the Fourier symbol of a 2-D function $m(x,y)$. We use the result in [15] to see the function $m(x,y)$ is

$$m(x,y) = \int_{S^1} \left[\frac{\pi i}{2}\text{sgn}(x\xi_1 + y\xi_2) - \log|x\xi_1 + y\xi_2|\right] \frac{\xi_2^2}{4\xi_1^2} d\sigma(\vec{\xi}). \tag{5.11}$$

From here, we would want to implement (5.11) into our boundary layer equation.

APPENDICES

APPENDIX A

List of Runge-Kutta Methods Used

The following are the tableau for the Runge-Kutta methods used.

0	0	0	0	0
1/2	1/2	0	0	0
1/2	0	1/2	0	0
1	0	0	1	0
	1/6	1/3	1/3	1/6

Table A.1: Explicit RK4 tableau

0	0	0	0	0	0	0	0
1/2	1/2	0	0	0	0	0	0
2/3	2/9	4/9	0	0	0	0	0
1/3	7/36	2/9	-1/12	0	0	0	0
5/6	-35/144	-55/36	35/48	15/8	0	0	0
1/6	-1/360	-11/36	-1/8	1/2	1/10	0	0
1	-41/260	22/13	43/156	-118/39	32/195	80/30	0
	13/200	0	11/40	11/40	4/25	4/25	13/200

Table A.2: Explicit Butcher 6 tableau

For the Cooper-Verner 11-step explicit RK8 tableau, see [5].

APPENDIX B

Local polynomial reconstruction

There are many choices that can be made to locally reconstruct a polynomial. For example, choose a linear (polynomial of degree/order 1) reconstruction. There are two different stencils that can be used. Figure B.1 shows the visualization of the linear upwind reconstruction.

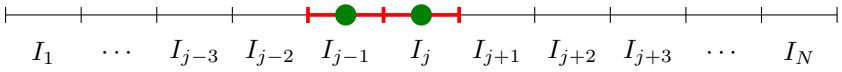

Figure B.1: Visualization of linear upwind stencil

The other choice of stencil, the linear downwind stencil, is visualized in Figure B.2.

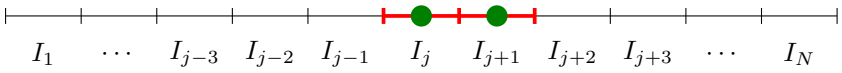

Figure B.2: Visualization of linear downwind stencil

The linear upwind polynomial reconstruction will be a polynomial $P_j(y)$ of degree 1 with the following properties:

$$\text{(B.1)} \qquad \frac{1}{\Delta x}\int_{I_{j-1}} P_j(y)\,dy = \overline{u}_{j-1} \qquad\qquad \frac{1}{\Delta x}\int_{I_j} P_j(y)\,dy = \overline{u}_j$$

We use basis functions to express $P_j(y)$, i.e. $P_j(y) = \overline{u}_{j-1}\varphi_j^{-1}(y) + \overline{u}_j\varphi_j^0(y)$, where

$\varphi_j^{-1}(y), \varphi_j^0(y)$ satisfy the following conditions

$$\begin{aligned} &\frac{1}{\Delta x}\int_{I_{j-1}} \varphi_j^{-1}(y)\,dy = 1 && \frac{1}{\Delta x}\int_{I_j} \varphi_j^{-1}(y)\,dy = 0 \\ &\frac{1}{\Delta x}\int_{I_{j-1}} \varphi_j^{0}(y)\,dy = 0 && \frac{1}{\Delta x}\int_{I_j} \varphi_j^{0}(y)\,dy = 1 \end{aligned} \tag{B.2}$$

A coordinate transformation is used to shift and rescale the intervals to be unit length. Let $z = (y - x_j)/\Delta x$. This change of variables turn (B.2) to

$$\begin{aligned} &\int_{-3/2}^{-1/2} \phi_j^{-1}(z)\,dz = 1 && \int_{-1/2}^{1/2} \phi_j^{-1}(z)\,dz = 0 \\ &\int_{-3/2}^{-1/2} \phi_j^{0}(z)\,dz = 0 && \int_{-1/2}^{1/2} \phi_j^{0}(z)\,dz = 1 \end{aligned} \tag{B.3}$$

where $\phi_j^{-1}(z)$ and $\phi_j^0(z)$ are scaled basis functions. Solving (B.3), the scaled basis functions are

$$\phi_j^{-1}(z) = 1 - (z+1) \qquad \phi_j^0(z) = 1 + z$$

To find other reconstructions, this is done by setting up similar equations to (B.1) for higher degree polynomials. These conditions are easy to formulate and are not reproduced here.

For a quadratic upwind reconstruction, the scaled basis functions are

$$\begin{aligned} \phi_j^{-2}(z) &= \frac{23}{24} - \frac{3}{2}(z+2) + \frac{1}{2}(z+2)^2 \\ \phi_j^{-1}(z) &= \frac{13}{12} - (z+1)^2 \\ \phi_j^{0}(z) &= \frac{23}{24} + \frac{3}{2}z + \frac{1}{2}z^2 \end{aligned}$$

The quadratic upwind reconstruction is visualized in Figure B.3.

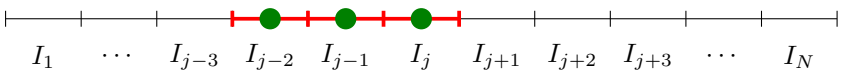

Figure B.3: Visualization of quadratic upwind stencil

For a cubic very upwind reconstruction, the scaled basis functions are

$$\begin{aligned}
\phi_j^{-3}(z) &= \frac{11}{12} - \frac{43}{24}(z+3) + (z+3)^2 - \frac{1}{6}(z+3)^3 \\
\phi_j^{-2}(z) &= \frac{13}{12} - \frac{5}{8}(z+2) - (z+2)^2 + \frac{1}{2}(z+2)^3 \\
\phi_j^{-1}(z) &= \frac{13}{12} + \frac{5}{8}(z+1) - (z+1)^2 - \frac{1}{2}(z+1)^3 \\
\phi_j^{0}(z) &= \frac{11}{12} + \frac{43}{24}z + z^2 + \frac{1}{6}z^3
\end{aligned}$$

To find other reconstructions of a given degree/order, a shift of the basis functions for another reconstruction of the same degree/order suffices. For example, consider polynomial reconstructions of degree 2. The quadratic upwind stencil has scaled basis functions

$$\begin{aligned}
\phi_{j,upwind}^{-2}(z) &= \frac{23}{24} - \frac{3}{2}(z+2) + \frac{1}{2}(z+2)^2 \\
\phi_{j,upwind}^{-1}(z) &= \frac{13}{12} - (z+1)^2 \\
\phi_{j,upwind}^{0}(z) &= \frac{23}{24} + \frac{3}{2}z + \frac{1}{2}z^2
\end{aligned}$$

The quadratic centered stencil has scaled basis functions

$$\begin{aligned}
\phi_{j,centered}^{-1}(z) = \phi_{j,upwind}^{-2}(z-1) &= \frac{23}{24} - \frac{3}{2}(z+1) + \frac{1}{2}(z+1)^2 \\
\phi_{j,centered}^{0}(z) = \phi_{j,upwind}^{-1}(z-1) &= \frac{13}{12} - z^2 \\
\phi_{j,centered}^{1}(z) = \phi_{j,upwind}^{0}(z-1) &= \frac{23}{24} + \frac{3}{2}(z-1) + \frac{1}{2}(z-1)^2
\end{aligned}$$

The quadratic centered reconstruction is visualized in Figure B.4.

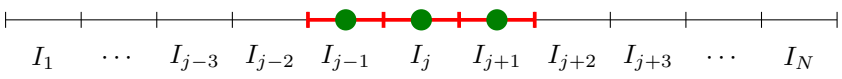

Figure B.4: Visualization of quadratic centered stencil

The quadratic downwind stencil has scaled basis functions

$$\phi^0_{j,downwind}(z) = \phi^{-2}_{j,upwind}(z-2) = \frac{23}{24} - \frac{3}{2}z + \frac{1}{2}z^2$$
$$\phi^1_{j,downwind}(z) = \phi^{-2}_{j,upwind}(z-2) = \frac{13}{12} - (z-1)^2$$
$$\phi^2_{j,downwind}(z) = \phi^{-2}_{j,upwind}(z-2) = \frac{23}{24} + \frac{3}{2}(z-2) + \frac{1}{2}(z-2)^2$$

The quadratic downwind reconstruction is visualized in Figure B.5.

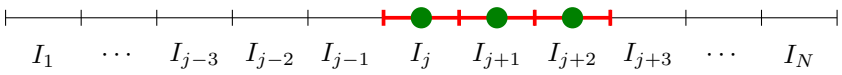

Figure B.5: Visualization of quadratic downwind stencil

APPENDIX C

Cell-average discretization for 2-D WWE

Consider the 2-D WWE

$$\begin{aligned}\frac{\partial^2 u}{\partial t^2} &+ \frac{1}{2\pi}\frac{\partial}{\partial x} P.V. \int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(x-z_1)u(z_1,z_2,t)\,dz_1dz_2}{((x-z_1)^2+(y-z_2)^2)^{3/2}} \\ &+ \frac{1}{2\pi}\frac{\partial}{\partial y} P.V. \int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(y-z_2)u(z_1,z_2,t)\,dz_1dz_2}{((x-z_1)^2+(y-z_2)^2)^{3/2}} = 0.\end{aligned} \tag{C.1}$$

Let the domain of our problem be $[0,L]\times[0,L]$. Let the x-domain be divided into N subintervals and the y-domain be divided into N subintervals. Then $\Delta x = \Delta y = L/N$ and $x_{j+1/2} = j\Delta x$ and $y_{k+1/2} = k\Delta y$. Denote the cell average of $u(x,y,t)$ on $I_j \times I_k = [x_{j-1/2}, x_{j+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ as $\overline{u}_{j,k}(t)$. We denote the vector of cell-averages as $\overline{U}(t) = (\overline{u}_{1,1}(t), \ldots, \overline{u}_{N,1}(t), \overline{u}_{2,1}(t), \ldots, \overline{u}_{2,N}(t), \ldots, \overline{u}_{N,N}(t))^T$.

To develop an evolution equation for $\overline{u}_{j,k}(t)$, we take the average of (C.1) on $I_j \times I_k$. Define

$$\begin{aligned}F(x,y;u) &= \frac{1}{2\pi} P.V. \int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(x-z_1)u(z_1,z_2,t)\,dz_1dz_2}{((x-z_1)^2+(y-z_2)^2)^{3/2}} \\ G(x,y;u) &= \frac{1}{2\pi} P.V. \int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(y-z_2)u(z_1,z_2,t)\,dz_1dz_2}{((x-z_1)^2+(y-z_2)^2)^{3/2}} \\ I &= \frac{\partial^2 u}{\partial t^2} \\ II &= \frac{1}{2\pi}\frac{\partial}{\partial x} P.V. \int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(x-z_1)u(z_1,z_2,t)\,dz_1dz_2}{((x-z_1)^2+(y-z_2)^2)^{3/2}} = \frac{\partial}{\partial x}F(x,y;u) \\ III &= \frac{1}{2\pi}\frac{\partial}{\partial y} P.V. \int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(y-z_2)u(z_1,z_2,t)\,dz_1dz_2}{((x-z_1)^2+(y-z_2)^2)^{3/2}} = \frac{\partial}{\partial y}G(x,y;u).\end{aligned}$$

First, take the cell-average of I on $I_j \times I_k$.

$$\frac{1}{\Delta x \Delta y} \int_{I_j} \int_{I_k} \frac{\partial^2 u}{\partial t^2} \, dy \, dx = \overline{u}''_{j,k}(t).$$

Next, take the cell-average of II on $I_j \times I_k$.

$$\begin{aligned} \frac{1}{\Delta x \Delta y} \int_{I_j} \int_{I_k} \left(\frac{\partial}{\partial x} F(x, y; u) \right) dy \, dx &= \frac{1}{\Delta x \Delta y} \int_{I_k} \left(F(x_{j+1/2}, y; u) - F(x_{j-1/2}, y; u) \right) dy \\ &\approx \frac{1}{\Delta x} \left(F(x_{j+1/2}, y_k; u) - F(x_{j-1/2}, y_k; u) \right). \end{aligned}$$

We approximate $F(x, y; u)$ by a midpoint rule to evaluate the infinite integrals. Because the integrals are evaluated at (x, y) values on the boundary of the subintervals, $F(x, y; u)$ will not be evaluated at a point where the integrand will diverge.

$$\begin{aligned} F(x, y; u) &= \frac{1}{2\pi} P.V. \int_{\mathbb{R}} \int_{\mathbb{R}} \frac{(x - z_1) u(z_1, z_2, t) \, dz_1 dz_2}{((x - z_1)^2 + (y - z_2)^2)^{3/2}} \\ &= \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{1}{2\pi} P.V. \int_{I_n} \int_{I_m} \frac{(x - z_1) u(z_1, z_2, t) \, dz_1 dz_2}{((x - z_1)^2 + (y - z_2)^2)^{3/2}} \\ &\approx \sum_{m=1}^{N} \sum_{n=1}^{N} \frac{1}{2\pi} P.V. \int_{I_n} \int_{I_m} \frac{(x - z_1) u(z_1, z_2, t) \, dz_1 dz_2}{((x - z_1)^2 + (y - z_2)^2)^{3/2}} \\ &\approx \sum_{m=1}^{N} \sum_{n=1}^{N} \frac{1}{2\pi} \left(\frac{(x - x_m) \overline{u}_{m,n}(t)}{((x - x_m)^2 + (y - y_n)^2)^{3/2}} \right) \Delta y \Delta x. \end{aligned}$$

So the cell-average of II on $I_j \times I_k$ is approximately

$$\begin{aligned} &\frac{1}{2\pi} \Delta y \sum_{m=1}^{N} \sum_{n=1}^{N} \left(\frac{(x_{j+1/2} - x_m) \overline{u}_{m,n}(t)}{((x_{j+1/2} - x_m)^2 + (y_k - y_n)^2)^{3/2}} - \frac{(x_{j-1/2} - x_m) \overline{u}_{m,n}(t)}{((x_{j-1/2} - x_m)^2 + (y_k - y_n)^2)^{3/2}} \right) \\ =&\frac{1}{2\pi \Delta x} \sum_{m=1}^{N} \sum_{n=1}^{N} \left(\frac{(j + 1/2 - m) \overline{u}_{m,n}(t)}{((j + 1/2 - m)^2 + (k - n)^2)^{3/2}} - \frac{(j - 1/2 - m) \overline{u}_{m,n}(t)}{((j - 1/2 - m)^2 + (k - n)^2)^{3/2}} \right) \\ =&X_{j,k} \overline{U}. \end{aligned}$$

Last, take the cell-average of III on $I_j \times I_k$.

$$\begin{aligned} \frac{1}{\Delta x \Delta y} \int_{I_j} \int_{I_k} \left(\frac{\partial}{\partial y} G(x, y; u) \right) dy \, dx &= \frac{1}{\Delta x \Delta y} \int_{I_j} \left(G(x, y_{k+1/2}; u) - G(x, y_{k-1/2}; u) \right) dx \\ &\approx \frac{1}{\Delta y} \left(G(x_j, y_{k+1/2}; u) - G(x_j, y_{k-1/2}; u) \right). \end{aligned}$$

We approximate $G(x,y;u)$ by a midpoint rule to evaluate the infinite integrals. Because the integrals are evaluated at (x,y) values on the boundary of the subintervals, $G(x,y;u)$ will not be evaluated at a point where the integrand will diverge.

$$\begin{aligned}
G(x,y;u) &= \frac{1}{2\pi} P.V. \int_{\mathbb{R}}\int_{\mathbb{R}} \frac{(y-z_2)u(z_1,z_2,t)\,dz_1dz_2}{((x-z_1)^2+(y-z_2)^2)^{3/2}} \\
&= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \frac{1}{2\pi} P.V. \int_{I_n}\int_{I_m} \frac{(y-z_2)u(z_1,z_2,t)\,dz_1dz_2}{((x-z_1)^2+(y-z_2)^2)^{3/2}} \\
&\approx \sum_{m=1}^{N}\sum_{n=1}^{N} \frac{1}{2\pi} P.V. \int_{I_n}\int_{I_m} \frac{(y-z_2)u(z_1,z_2,t)\,dz_1dz_2}{((x-z_1)^2+(y-z_2)^2)^{3/2}} \\
&\approx \sum_{m=1}^{N}\sum_{n=1}^{N} \frac{1}{2\pi} \left(\frac{(y-y_n)\overline{u}_{m,n}(t)}{((x-x_m)^2+(y-y_n)^2)^{3/2}} \right) \Delta y \Delta x.
\end{aligned}$$

So the cell-average of III on $I_j \times I_k$ is approximately

$$\begin{aligned}
&\frac{1}{2\pi}\Delta x \sum_{m=1}^{N}\sum_{n=1}^{N} \left(\frac{(y_{k+1/2}-y_n)\overline{u}_{m,n}(t)}{((x_j-x_m)^2+(y_{k+1/2}-y_n)^2)^{3/2}} - \frac{(y_{k-1/2}-y_n)\overline{u}_{m,n}(t)}{((x_j-x_m)^2+(y_{k-1/2}-y_n)^2)^{3/2}} \right) \\
=&\frac{1}{2\pi\Delta x} \sum_{m=1}^{N}\sum_{n=1}^{N} \left(\frac{(k+1/2-n)\overline{u}_{m,n}(t)}{((j-m)^2+(k+1/2-n)^2)^{3/2}} - \frac{(k-1/2-n)\overline{u}_{m,n}(t)}{((j-m)^2+(k-1/2-n)^2)^{3/2}} \right) \\
=&Y_{j,k}\overline{U}.
\end{aligned}$$

Thus, an evolution equation for $\overline{u}_{j,k}(t)$ is

$$\overline{u}''_{j,k}(t) + X_{j,k}\overline{U} + Y_{j,k}\overline{U} = 0. \tag{C.2}$$

Writing this as a system of equations for the vector of solutions $\overline{U}$, we have

$$\overline{U}''(t) + M\overline{U} = 0. \tag{C.3}$$

APPENDIX D

Cell-average discretization for boundary layer equation (first approximation, (4.15))

For this appendix, we focus on the right-moving waves. The left-, upward-, and downward-moving waves, have similar analyses. Recall equation (4.15):

$$\frac{\partial}{\partial t}u(x,y,t) + \frac{\partial}{\partial x}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{u(z,y,t)\,dz}{\sqrt{|x-z|}} = 0. \tag{D.1}$$

Define

$$\begin{aligned} F(x,y;u) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{u(z,y,t)\,dz}{\sqrt{|x-z|}} \\ I &= \frac{\partial}{\partial t}u(x,y,t) \\ II &= \frac{\partial}{\partial x}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{u(z,y,t)\,dz}{\sqrt{|x-z|}} = \frac{\partial}{\partial x}F(x,y;u). \end{aligned}$$

First, take the cell-average of I on $I_j \times I_k$.

$$\frac{1}{\Delta x\Delta y}\int_{I_j}\int_{I_k}\frac{\partial}{\partial t}u(x,y,t)\,dy\,dx = \overline{u}'_{j,k}(t).$$

Next, take the cell-average of II on $I_j \times I_k$.

$$\frac{1}{\Delta x\Delta y}\int_{I_j}\int_{I_k}\left(\frac{\partial}{\partial x}F(x,y;u)\right)\,dy\,dx = \frac{1}{\Delta x\Delta y}\int_{I_k}\left(F(x_{j+1/2},y;u) - F(x_{j-1/2},y;u)\right)\,dy.$$

We now look at $\dfrac{1}{\Delta x\Delta y}\displaystyle\int_{I_j}F(x_{j+1/2},y;u)dy$. Using a linear upwind polynomial re-

construction in x for $u(x,y,t)$ in the integral, we have

$$\begin{aligned}
\frac{1}{\Delta x \Delta y}\int_{I_k} F(x_{j+1/2},y;u)\,dy &= \frac{1}{\Delta x\Delta y}\int_{I_k}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{u(z,y,t)\,dz}{\sqrt{|x_{j+1/2}-z|}}\,dy \\
&\approx \frac{1}{\Delta x\Delta y}\int_{I_k}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{P_j(z,y,t)\,dz}{\sqrt{|x_{j+1/2}-z|}}\,dy \\
&= \frac{1}{\Delta x}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{\left(\overline{u}_{j,k}(t)\phi_j^0(z)+\overline{u}_{j-1,k}(t)\phi_j^{-1}(z)\right)\,dz}{\sqrt{|x_{j+1/2}-z|}} \\
&= \frac{1}{\Delta x}\left(\sqrt{\Delta x}\widetilde{A_1}\right)\overline{U}_{:,k}
\end{aligned}$$

where $\widetilde{A_1}$ is a $N\times N$ matrix and $\overline{U}_{:,k}$ is a N vector of solution cell averages that depend on the solution cell averages for all cells in the x-direction and I_k in the y-direction.

Combining all of this together, we find that (D.1) can be implemented by solving

$$\overline{U}'(t)+\frac{1}{\sqrt{\Delta x}}\widetilde{B_1}\overline{U}=0$$

for some $N^2\times N^2$ matrix, $\widetilde{B_1}$, that depends on approximating $F(x,y;u)$.

APPENDIX E

Cell-average discretization for boundary layer equation (second approximation, (4.17))

For this appendix, we focus on the right-moving waves. The left-, upward-, and downward-moving waves, have similar analyses. Recall equation (4.17):

$$\begin{aligned}\frac{\partial}{\partial t}u(x,y,t) + \frac{\partial}{\partial x}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{u(z,y,t)\,dz}{\sqrt{|x-z|}} \\ + \frac{\partial^2}{\partial y^2}\frac{\partial}{\partial x}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}u(z,y,t)\,dz = 0.\end{aligned} \tag{E.1}$$

Define

$$\begin{aligned}F(x,y;u) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{u(z,y,t)\,dz}{\sqrt{|x-z|}} \\ G(x,y;u) &= \frac{\partial}{\partial x}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}u(z,y,t)\,dz \\ I &= \frac{\partial}{\partial t}u(x,y,t) \\ II &= \frac{\partial}{\partial x}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\frac{u(z,y,t)\,dz}{\sqrt{|x-z|}} = \frac{\partial}{\partial x}F(x,y;u) \\ III &= \frac{\partial^2}{\partial y^2}\frac{\partial}{\partial x}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}u(z,y,t)\,dz = \frac{\partial}{\partial x}G(x,y;u).\end{aligned}$$

Note that we show how to cell-average terms I and II in Appendix D.

Take the cell-average of III on $I_j \times I_k$.

$$\begin{aligned}\frac{1}{\Delta x\Delta y}\int_{I_j}\int_{I_k}\left(\frac{\partial^2}{\partial y^2}\frac{\partial}{\partial x}G(x,y;u)\right)dy\,dx =& \frac{1}{\Delta x\Delta y}\int_{I_k}\frac{\partial^2}{\partial y^2}G(x_{j+1/2},y;u)],dy \\ &- \frac{1}{\Delta x\Delta y}\int_{I_k}\frac{\partial^2}{\partial y^2}G(x_{j-1/2},y;u)\,dy.\end{aligned}$$

We now look at $\frac{1}{\Delta x \Delta y}\int_{I_k}\frac{\partial^2}{\partial y^2}G(x_{j+1/2},y;u)\,dy$. Interchanging the $\frac{\partial^2}{\partial y^2}$ term into the integral and applying it to $u(z,y,t)$, we have

$$\begin{aligned}
\frac{1}{\Delta x\Delta y}\int_{I_k}\frac{\partial^2}{\partial y^2}G(x_{j+1/2},y;u)\,dy =& \frac{1}{\Delta x\Delta y}\int_{I_k}\frac{\partial^2}{\partial y^2}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}u(z,y,t)\,dz\,dy\\
=& \frac{1}{\Delta x\Delta y}\int_{I_k}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}u_{yy}(z,y,t)\,dz\,dy\\
\approx& \frac{1}{\Delta x}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}u_{yy}(z,y_k,t)\,dz\\
\approx& \frac{1}{\Delta x\Delta y^2}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}u(z,y_{k+1},t)\,dy\\
&-\frac{1}{\Delta x\Delta y^2}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}2u(z,y_k,t)\,dz\\
&+\frac{1}{\Delta x\Delta y^2}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}u(z,y_{k-1},t)\,dz.
\end{aligned}$$

Using a linear upwind polynomial reconstruction in x for $u(x,y,t)$ in the integral, we have

$$\begin{aligned}
\approx& \frac{1}{\Delta x\Delta y^2}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}\left(\overline{u}_{j,k+1}(z)\phi_j^0(z)+\overline{u}_{j-1,k+1}(z)\phi_j^{-1}(z)\right)\,dy\\
&-\frac{1}{\Delta x\Delta y^2}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}2\left(\overline{u}_{j,k}(z)\phi_j^0(z)+\overline{u}_{j-1,k}(z)\phi_j^{-1}(z)\right)\,dy\\
&+\frac{1}{\Delta x\Delta y^2}\frac{1}{3\sqrt{2\pi}}\int_{-\infty}^{\infty}|x-z|^{3/2}\left(\overline{u}_{j,k-1}(z)\phi_j^0(z)+\overline{u}_{j-1,k-1}(z)\phi_j^{-1}(z)\right)\,dy\\
=& \frac{1}{\Delta x\Delta y^2}\widetilde{A_2}\left(\overline{U}_{:,k+1}-2\overline{U}_{:,k}+\overline{U}_{:,k-1}\right)
\end{aligned}$$

where $\widetilde{A_2}$ is a $N\times N$ matrix and $\overline{U}_{:,k}$ is a N vector of solution cell averages that depend on the solution cell averages for all cells in the x-direction and I_k in the y-direction.

Combining this with the cell averages of I and II from appendix D, we find that (E.1) can be implemented by solving

$$\overline{U}'(t)+\frac{1}{\sqrt{\Delta x}}\widetilde{B_2}\overline{U}=0$$

for some $N^2\times N^2$ matrix, $\widetilde{B_2}$, that depends on approximating $F(x,y;u)$ and $G(x,y;u)$.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] A. Bayliss and E. Turkel. Radiation boundary conditions for wave-like equations. *Communications on Pure and Applied Mathematics*, 33(6):707–725, 1980.

[2] J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185–200, 1994.

[3] F. Betancourt, R. Bürger, K. H. Karlsen, and E. M. Tory. On nonlocal conservation laws modelling sedimentation. *Nonlinearity*, 24(3):855–885, 2011.

[4] S. Carney. Numerical methods for the One Way Water Wave Equation. REU Report, 2013.

[5] G. J. Cooper and J. H. Verner. Some explicit Runge-Kutta methods of high order. *SIAM Journal on Numerical Analysis*, 9(3):389–405, 1972.

[6] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Mathematics of Computation*, 31(139):629–651, 1977.

[7] M. B. Giles. Nonreflecting boundary conditions for Euler equation calculations. *AIAA Journal*, 28(12):2050–2058, 1990.

[8] D. Givoli and D. Cohen. Nonreflecting boundary conditions based on Kirchhoff-type formulae. *Journal of Computational Physics*, 117(1):102–113, 1995.

[9] A. Iserles and G. Strang. The optimal accuracy of difference schemes. *Transactions of the American Mathematical Society*, 277(2):779–803, 1983.

[10] M. Israeli and S. A. Orszag. Approximation of radiation boundary conditions. *Journal of Computational Physics*, 41(1):115–135, 1981.

[11] G. I. Jennings. *Efficient Numerical Methods for Water Wave Propagation in Unbounded Domains*. PhD thesis, University of Michigan, 2012.

[12] G. I. Jennings, S. Karni, and J. B. Rauch. Water wave propagation in unbounded domains. Part I: nonreflecting boundaries. *Journal of Computational Physics*, 276:729–739, 2014.

[13] G. I. Jennings, D. Prigge, S. Carney, S. Karni, J. B. Rauch, and R. Abgrall. Water wave propagation in unbounded domains. Part II: Numerical methods for fractional PDEs. *Journal of Computational Physics*, 275:443–458, 2014.

[14] I. Orlanski. A simple boundary condition for unbounded hyperbolic flows. *Journal of Computational Physics*, 21(3):251 – 269, 1976.

[15] E. M. Stein. *Singular Integrals and Differentiability Properties of Functions*. Princeton University Press, 1970.

[16] S. Wu. Well-posedness in Sobolev spaces of the full water wave problem in 2-D. *Inventiones Mathematicae*, 130(1):39–72, 1997.

[17] S. Wu. Well-posedness in Sobolev spaces of the full water wave problem in 3-D. *Journal of the American Mathematical Society*, 12(2):445–495, 1999.

[18] S. Wu. Almost global wellposedness of the 2-D full water wave problem. *Inventiones Mathematicae*, 177(1):45–135, 2009.

[19] S. Wu. Global wellposedness of the 3-D full water wave problem. *Inventiones Mathematicae*, 184(1):125–220, 2011.

[20] Q. Yang, F. Liu, and I. Turner. Numerical methods for fractional partial differential equations with Riesz space fractional derivatives. *Applied Mathematical Modelling. Simulation and Computation for Engineering and Environmental Systems*, 34(1):200–218, 2010.