COMPUTATION OF STOKES WAVES USING CONFORMAL MAPPING

A Thesis by

Justin Laurence Mears

Bachelor of Business Administration, Wichita State University, 2009

Submitted to the Department of Mathematics, Statistics, and Physics
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

May 2016

COMPUTATION OF STOKES WAVES USING CONFORMAL MAPPING

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Mathematics.

Thomas DeLillo, Committee Chair

Kenneth Miller, Committee Member

Zheng Chen, Committee Member

ABSTRACT

In this paper we study the time dependent water wave problem for Stokes waves on water of infinite and finite depth using numerical conformal mapping techniques. We begin by briefly reviewing Fornberg's numerical conformal mapping method and then briefly develop the theory for periodic cubic splines which are necessary in our later applications. Since this study involves the dynamics of water waves, we derive the hydrodynamics equations describing such motion. It is here that we find that the velocity potential of an incompressible, irrotational fluid flow satisfies Laplace's equation throughout the fluid and that all motion on the surface of a fluid depends on what happens on the interior. We then develop a numerical representation for Stokes waves and use this representation to study the time dependent wave problem on water with infinite and finite depth. It is here that conformal mapping techniques become important because solutions to Laplace's equation are invariant under conformal mapping, allowing us to solve the problem for the velocity potential in more ideal domains, namely the unit disk and the annulus with $\rho \leq |z| \leq 1$, rather than the physical plane. We then develop a numerical approach to both problems utilizing Euler's method to solve the differential equations describing the free surface conditions of the wave.

ACKNOWLEDGEMENTS

I would like to thank Bengt Fornberg and Jonathan Birge for making their original research, methods and codes utilized for their research of Stokes waves on water of infinite depth available for my study. Their work was crucial in my understanding of the infinite depth case allowing me to successfully extend their approach and methods to handle the finite depth problem. Their generosity was pivotal to my success.

TABLE OF CONTENTS

Chapter **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Problems involving fluid motion, especially in the case of water, have been studied throughout history. As Craik comments in [Craik04], many notable mathematicians and physicists such has Newton, Euler, Laplace, Lagrange and Cauchy have made significant contributions to the studies of fluid mechanics and fluid dynamics. It is in the field of fluid dynamics that George Gabriel Stokes made his most important contributions regarding the study of ocillatory waves. [Craik05] He first addressed small amplitude periodic water waves and their velocity in 1847 with [Stok47] and then later revised his approach to simplify calculations in his 1880 paper. In his 1847 paper, Stokes motivates his study by considering a ship at sea with no astronomical means of determining its position other than estimation by its previous direction and velocity. "But," he states, "the estimated velocity and direction of motion of the ship are her velocity and direction of motion relatively to the water." [Stok47] "If then the (entirety) of the water near the surface is moving in the direction of the waves," then the ship's estimation will be wrong. [Stok47] This then gives a reason for the ship to be able to calculate the velocity of the water with data easily gathered from the deck of the ship, in this case approximate wave height and length. [Stok47] It is this type of velocity problem, and associated wave propagation, which will be covered in this study utilizing Stokes' updated calculations as covered in [Sch74].

Luckily because of the governing equations of fluid mechanics these types of problems are natural applications for conformal mapping techniques. Bengt Fornberg first addressed the problem of Stokes waves on water with infinite depth around 1980 in [Forwave] utilizing his Newton-like conformal mapping technique in FORTRAN. In 2000 Jonathan Birge readdresses this same problem in [Birge] using MATLAB.

In this paper we extend Birge's approach to utilize DeLillo's Fornberg-like conformal mapping method to solve the velocity problem for Stokes waves on water of infinite depth. We

then follow a similar approach to solving the velocity problem of Stokes waves on water of finite depth.

CHAPTER 2

CONFORMAL MAPPING

We say that a function mapping between two regions in the complex plane is conformal if the map preserves angles. An obvious consequence of this definition is that conformal mapping plays an influential role in the geometric theory of analytic functions. [Ahl] As a result of some existence and uniqueness theorems we are able to determine analytic functions without expressing them explicitly and also gain insight on analytic properties about the mapping function by considering the geometric properties of the regions that are being mapped. [Ahl] As one of the afformentioned theorems, the Riemann Mapping Theorem describes existence and uniqueness of a biholomorphic function between the open unit disk and a simply connected region in the complex plane. This theorem is important in any discussion involving conformal mapping and thus will be important in this study of numerical conformal mapping. The theorem as stated in [Ahl] follows here without proof.

Theorem 2.0.1 (The Riemann Mapping Theorem)**.** *Given any simply connected region Ω which is not the whole plane, and a point $z_0 \in \Omega$, there exists a unique analytic function $f(z)$ in Ω, normalized by the conditions $f(z_0) = 0$, $f'(z_0) > 0$, such that $f(z)$ defines a one-to-one mapping of Ω onto the disk $|w| < 1$.*

Because we are also considering doubly connected regions we will need an analogue of the Riemann Mapping Theorem involving doubly connected regions and the unit annulus stated from [Hen86] without proof.

Theorem 2.0.2. *Let Ω be a nondegenerate doubly connected region. Then there exists a unique real number ρ, $0 < \rho < 1$, such that there exists a one-to-one analytic function g that maps Ω onto the annulus $\rho < |z| < 1$. If the outer boundaries correspond to each other, then g is determined up to a rotation of the annulus.*

We will also need a basic result from the theory of complex integrals. Using Cauchy's integral theorem, we can represent any analytic function in terms of an integral formula. The proof for this theorem, and more information on complex integral there is available in [Ahl].

Theorem 2.0.3 (Cauchy Integral Formula)**.** *Let $f(z)$ be analytic in a simply connected domain Ω with interior D. Also let C be a simple, positively oriented, closed curve in Ω and $z \in D$. Then*

$$f(z) = \frac{1}{2\pi i} \oint_C \frac{f(\zeta)}{\zeta - z}\, d\zeta$$

The last result for our study of numerical conformal mapping methods involves the boundary behavior and analytic extension of conformal maps. This theorem is stated from [Hen86] without proof.

Theorem 2.0.4 (Osgood-Carathéodory Theorem)**.** *Let Ω be a Jordan region and let g be a conformal map of Ω onto the open unit disk D. Then f can be extended to a topological map of the closure of Ω onto the closure of D.*

With these basic results we can now make a brief introduction to Fornberg's conformal mapping method.

2.1 Fornberg's Conformal Mapping Method

We follow a brief analysis of material from [De14], [De15] and [DP98] where these topics are covered in greater detail. Suppose that the we have a region Ω bounded by a Jordan curve Γ parameterized by S such that $\Gamma : \gamma(S)$, with $0 \leq S \leq L$ and $\gamma(0) = \gamma(L)$. The Osgood-Carathéodory Theorem allows us to define the *boundary correspondence* function, $\theta(S) = \arg g(\gamma(S))$ and its inverse, called the boundary correspondence, $S = S(\theta)$. So if we know the boundary correspondence then by the Cauchy Integral formula we know the conformal map for $z \in D$,

$$f(z) = \frac{1}{2\pi i} \int_{|\zeta|} \frac{\gamma(S(\theta))}{\zeta - z}\, d\zeta(\theta).$$

Fornberg's method finds the conformal map $f(z)$ by solving this nonlinear integral equation for $S(\theta)$ by a quadratically convergent Newton-like method. This method allows for the use

of fast Fourier transforms resulting with operation counts of $O(N \log N)$. Fornberg's methods in [For80] and [For84] describe numerical approachs for constructing conformal maps from the interior of the unit disk to the interior of a simply connected region or from the interior of the unit annulus to the interior of a doubly connected region, respectively. We begin by considering the method for simply connected regions.

2.1.1 Simply Connected Fornberg

For a simply connected region Ω bound by a smooth Jordan curve Γ, Fornberg's method finds a conformal map from the interior of the unit disk to the interior of Ω by solving a nonlinear integral equation for the boundary correspondence $S(\theta)$ using a Newton-like scheme. So, suppose we have such a region Ω and curve Γ parameterized by S such that $\Gamma : \gamma(S)$ with the conditions that $f(0) = 0$ and either $f'(0) > 0$ or $f(1)$ is fixed. The problem is then to find the boundary correspondence $S(\theta)$ so that the conformal map f is analytic in the unit disk. Given the k-th Netwon iterate $S = S^k(\theta)$, we want to find a real valued correction $U^k(\theta)$ so that the linearization

$$f(e^{i\theta}) = \gamma\left(S^k(\theta) + U^k(\theta)\right) \approx \gamma\left(S^k(\theta)\right) + \gamma'\left(S^k(\theta)\right) U^k(\theta) \tag{2.1}$$

extends analytically into the interior of the unit disk with $f(0) = 0$. The conditions under which this will occur are established by the following analytic extension theorem.

Theorem 2.1.1. *A function $f \in Lip(C)$ on the boundary C of the unit disk, with Fourier expansion $f(e^{i\theta}) = \sum_{k=-\infty}^{\infty} a_k e^{ik\theta}$, extends to an analytic function in the interior of the disk with $f(0) = 0$ if and only if*

$$\sum_{k=-\infty}^{0} a_k e^{ik\theta} = 0, \tag{2.2}$$

that is, the negative indexed Fourier coefficients must be 0.

Now, if the linearization given by equation(2.1) satisfies Theorem 2.1.1 then we can find a linear system for the correction $U^k(\theta)$ that can be discretized by N-point trigonometric interpolation and solved superlinerarly with the conjugate gradient. The Newton update is

then given by

$$\underline{S}^{k+1} = \underline{S}^{(k)} + \underline{U}^{(k)}$$

with $U_0 = 0$ set to fix a boundary point. Next we consider Fornberg's method for doubly connected regions.

2.1.2 Doubly Connected Fornberg

Now suppose we have a region $\widehat{\Omega}$ bounded by two smooth Jordan curves Γ_1 and Γ_2 parameterized by S_1 and S_2 respectively, so that $\Gamma_1 : \gamma_1(S_1)$ and $\Gamma_2 : \gamma_1(S_2)$. Fornberg's method for the doubly connected case finds a conformal map f from the annulus $\rho \leq |z| \leq 1$ to $\widehat{\Omega}$. This is more complicated than the simply connected case because we have to determine the conformal modulus ρ and establish some type of uniqueness condition since f is only determined up to rotations. The problem in this case then is to find the boundary correspondences $S_1(\theta)$ and $S_2(\theta)$ and the conformal modulus ρ such that the conformal mapping function f is analytic in the annulus $\rho < |z| < 1$ and $f(e^{i\theta}) = \gamma_1\left(S_1(\theta)\right)$ and $f(\rho e^{i\theta}) = \gamma_2\left(S_2(\theta)\right)$. Each step uses a Newton-like scheme to compute corrections $U_1(\theta)$, $U_2(\theta)$ and $\delta\rho$ to $S_1(\theta)$, $S_2(\theta)$ and ρ, respectively, so that the following linearizations about S_1, S_2 and ρ

$$\begin{aligned} f(e^{i\theta}) &\approx \gamma_1\left(S_1(\theta)\right) + \gamma_1'\left(S_1(\theta)\right) U_1(\theta) \\ f(\rho e^{i\theta}) &\approx \gamma_2\left(S_2(\theta)\right) + \gamma_2'\left(S_2(\theta)\right) U_2(\theta) + i\gamma_2'\left(S_2(\theta)\right) \frac{dS_2(\theta)/d\theta}{\rho}\delta\rho. \end{aligned}$$

extend analytically into the annulus. The conditions for analytic extension is given by the following theorem.

Theorem 2.1.2. *Let C_1 and C_2 respectively denote the outer and inner boundaries of the annulus $\rho < |z| < 1$ and set $C = C_1 - C_2$. Then the function $f \in Lip(C)$ extends analytically to the annulus $\rho < |z| < 1$ if and only if*

$$\int_{C_1} f(z) z^k \, dz = \int_{C_2} f(z) z^k \, dz, \quad \textit{for} \quad k = \pm 1, 2, 3, \ldots.$$

If we let

$$f(e^{i\theta}) = \sum_{k=-\infty}^{\infty} a_k e^{ik\theta} \qquad \textit{and} \qquad f(\rho e^{i\theta}) = \sum_{k=-\infty}^{\infty} b_k e^{ik\theta}$$

then the above condition becomes

$$\rho^k a_k = b_k, \quad \textit{for} \quad k = \pm 1, 2, 3, \ldots.$$

So, if $b_k = \rho^k a_k$ then we can find a linear system for our corrections, discretize using N-point trigonometric interpolation and solve using the conjugate gradient. Finally, the Newton update is given by

$$\begin{aligned} S_1^{k+1}(\theta) &= S_1^k(\theta) + U_1^k(\theta), \\ S_2^{k+1}(\theta) &= S_2^k(\theta) + U_2^k(\theta), \\ \rho_{k+1} &= \rho_k + \delta\rho_k. \end{aligned}$$

CHAPTER 3

CUBIC SPLINES

3.1 Periodic Cubic Splines

Employed in the previously discussed conformal mapping algorithms and again in later applications, the Periodic Cubic Spline is an example of a piecewise cubic polynomial *interpolating function*. A thorough analysis of the theory behind cubic splines is available in texts such as [Hen82] and will not be addressed in this paper, however we will follow the construction in [Mol04] regarding the periodic cubic spline.

Given periodic data points in the plane (x_k, y_k) for $k = 0,\ 1,\ \ldots,\ n$, with distinct interpolation points x_k and with the condition that $y_n = y_1$, we wish to find the unique piecewise cubic interpolating polynomial function P satisfying the following conditions.

1. $P(x_k) = y_k$ for $k = 0,\ 1,\ \ldots,\ n$. (This guarantees the function reproduces the given data and is continuous at all interpolating points.)

2. $P'(x_k^-) = P'(x_k^+)$ for $k = 1,\ 2,\ \ldots,\ n-1$. (This guarantees the slope of the interpolant will match on both sides of an interpolation point, i.e. no corners.)

3. $P''(x_k^-) = P''(x_k^+)$ for $k = 1,\ 2,\ \ldots,\ n-1$, that is, $P \in C^2$. (This guarantees the smoothness of the interpolant since the second derivative exists and is continuous.)

4. $P(x + \Delta) = P(x)$ for all x where $\Delta = x_n - x_1$. (Periodicity condition.)

So to find P begin by letting $[a, b]$ be a closed finite interval partitioned by the interpolating points x_k for $k = 0,\ 1,\ \ldots,\ n$. It is important to note here that we are allowed to choose how we parameterize this interval. For example the conformal mapping methods discussed in the Chapter 2 use the chordal arclength of the curve to parameterize the partition. Later we will use an finite set of equally spaced intervals. Regardless of the method chosen the

partitioned interval should resemble the follow.

$$a = x_0 < x_1 < x_2 < \cdots < x_{k-1} < x_k < x_{k+1} < \cdots < x_{n-1} < x_n = b.$$

Suppose x is any element in the subinterval $[x_k, x_{k+1}]$ and define $s(x) = s = x - x_k$, $h_k = h = x_{k+1} - x_k$ and $d_k = P'(x_k)$ as the local variable(in terms of x), interval length and slope of the interpolant at x_k respectively. Thus for each interval, $[x_k, x_{k+1}]$ we must find a cubic polynomial

$$P(s) = c_1 s^3 + c_2 s^2 + c_3 s + c_4 \tag{3.1}$$

satisfying the given conditions. One can easily see that the following are equivalent to conditions (1) and (2),

$$\begin{aligned} P(x_k)(= P(0)) &= y_k, & P'(x_k)(= P'(0)) &= d_k, \\ P(x_{k+1})(= P(h)) &= y_{k+1}, & P'(x_{k+1})(= P'(h)) &= d_{k+1}. \end{aligned}$$

Requiring these relationships to hold results in the coefficients

$$\begin{aligned} c_1 &= \frac{d_k h - 2(y_{k+1} - y_k) + d_{k+1} h}{h^3} & c_2 &= \frac{3(y_{k+1} - y_k) - 2d_k h - d_{k+1} h}{h^2} \\ c_3 &= d_k & c_4 &= y_k. \end{aligned}$$

Applying these values to (2.1) and rearranging yields,

$$P(s) = \frac{s^2(3h - 2s)}{h^3} y_{k+1} + \frac{h^3 - 3hs^2 + 2s^3}{h^3} y_k + \frac{s^2(s-h)}{h^2} d_{k+1} + \frac{s(s-h)^2}{h^2} d_k \tag{3.2}$$

which is the form that will be used for the remainder of this discussion. One can verify that this form still satisfies the given equivalent conditions. Note that (2.2) implies

$$P'(s) = \frac{6sh - 6s^2}{h^3} y_{k+1} + \frac{6s^2 - 6sh}{h^3} y_k + \frac{3s^2 - 2sh}{h^2} d_{k+1} + \frac{3s^2 - 4sh + h^2}{h^2} d_k \tag{3.3}$$

Condition (3) applies to the second derivative and so from equation (2.2) we differentiate to get

$$P''(s) = \left(\frac{6h - 12s}{h^2}\right)\left(\frac{y_{k+1} - y_k}{h}\right) + \frac{6s - 2h}{h^2} d_{k+1} + \frac{6s - 4h}{h^2} d_k \tag{3.4}$$

which upon introducing the divided difference $\delta_k := \dfrac{y_{k+1} - y_k}{h}$, simplifies to

$$P''(s) = \frac{(6h - 12s)\delta_k + (6s - 2h)d_{k+1} + (6s - 4h)d_k}{h^2}. \tag{3.5}$$

Letting $h = h_k$, substituting and simplifying we then have that

$$P''(x_k^-) = \frac{-6\delta_{k-1} + 4d_k + 2d_{k-1}}{h_{k-1}} \quad \text{and} \quad P''(x_k^+) = P''(0) = \frac{6\delta_k - 2d_{k+1} - 4d_k}{h_k}. \tag{3.6}$$

Ensuring that condition (3) is satisfied yields

$$\frac{-6\delta_{k-1} + 4d_k + 2d_{k-1}}{h_{k-1}} = \frac{6\delta_k - 2d_{k+1} - 4d_k}{h_k} \tag{3.7}$$

which simplifies to

$$h_k d_{k-1} + 2(h_{k-1} + h_k)d_k + h_{k-1}d_k = 3(h_k\delta_{k-1} + h_{k-1}\delta_k) \tag{3.8}$$

for $k = 2, 3, ..., n-1$. Since the h_k's and δ_k's are known, equation (3.8) represents a $(n-2)$ system of linear equations for the n unknown d_k's. The extra information needed to solve the system is given by the periodic endpoint data described in condition (4).

Now with $\Delta - \textit{periodic}$ data we have the restriction on the input y_k's such that

$$P(x_1) = y_1 = P(x_n) = y_n. \tag{3.9}$$

Also condition (2) still holds for the endpoints and so

$$P'(x_1) = d_1 = P'(x_n) = d_n. \tag{3.10}$$

Two immediate consequences of this periodicity is that one equation for the d_k's is eliminated and equation (3.8) holds for all $k = 1, ..., n-1$. The resulting $(n-1)$ system of linear equations is then of the form $Ad = r$ where A is a $(n-1) \times (n-1)$ tri-diagonal matrix with the exception of two off diagonal corner elements, d is a $(n-1)$ length vector of the unknown d_k's, and r is a $(n-1)$ length vector of the right-hand side of equation (3.8). Thus the elements of the system resemble the following.

$$Ad = \begin{bmatrix} 2(h_{n-1}+h_1) & h_{n-1} & 0 & \cdots & 0 & h_1 \\ h_2 & 2(h_1+h_2) & h_1 & 0 & 0 & 0 \\ 0 & h_2 & 2(h_2+h_3) & h_2 & 0 & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & h_{n-2} & 2(h_{n-3}+h_{n-2}) & h_{n-3} \\ h_{n-2} & 0 & \cdots & 0 & h_{n-1} & 2(h_{n-2}+h_{n-1}) \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{n-2} \\ d_{n-1} \end{bmatrix}$$

$$r = 3 \begin{bmatrix} h_1\delta_{n-1} + h_{n-1}\delta_1 \\ h_2\delta_1 + h_1\delta_2 \\ h_3\delta_2 + h_2\delta_3 \\ \vdots \\ \vdots \\ h_{n-1}\delta_{n-2} + h_{n-2}\delta_{n-1} \end{bmatrix}$$

This system is easily solvable using Guassian elimination with back substitution. Such an approach is developed in the following algorithms.

3.2 Algorithm

The algorithms, available in Appendix A, are modifications of MATLAB codes for cubic splines with knot-a-knot boundary conditions orginally written by Cleve Moler of Mathworks which are available on the Mathworks website associated with [Mol04]. These modifications were originally assigned as an exercise in [Mol04] where the reader is tasked with making appropriate modifications to the original code so that it utilizes periodic boundary input data and returns a piecewise periodic cubic spline interpolant for that given input data. The accuracy of such a method is of $O(n^4)$ with cost of $O(n^2)$ operations. Most of this work occurs in the processes of finding the slopes, d_k at the interpolation points.

The first algorithm in the process, algorithm (A.1) consists of the `periodic_splinetx.m` algorithm with the `periodic_splineslopes` subroutine. The inputs for the algorithm are the Δ-periodic x_k interpolation points with corresponding y_k's and the u_k values at which the interpolant will be evaluated. The first steps of `periodic_splinetx.m` involve finding the h_k and δ_k values from the inputs then passes that information to the `periodic_splineslopes` subroutine which uses this information to set up the tridiagonal system $Ad = r$. Because of the periodic nature of the data most of the modifications to algorithm (A.1) occured in the subroutine and consisted of changes to the lengths of the vectors describing the tridiagonal matrix A and some modifications to the vector r on the right-hand side of the system. The

most significant modification to the tridiagonal matrix occurred in the following two lines which changed the center diagonal.

```
%Center Row of Tridiagonal System
  b(1) = 2*(h(n)+h(1));
  b(2:n) = 2*(h(2:n)+h(1:n-1));.
```

To accommodate the periodic data, the right-hand side was modified by the following.

```
%Right-hand side
  r(1) = 3*(h(1)*delta(n) + h(n)*delta(1));
  r(2:n) = 3*(h(2:n).*delta(1:n-1)+h(1:n-1).*delta(2:n));
```

Once the system is setup by the subroutine, it is passed to `periodic_tridisolve.m` which solves the system for the d_k's.

The `periodic_tridisolve.m` algorithm, given by algorithm (A.2), uses Gaussian elimination with back substitution to solve the tridiagonal system for the d_k's, that is, the slopes at each of the interpolant . Because the system includes two off-diagonal elements the calculations for solving the system are slightly more complicated than a purely tridiagonal system. The first place this arises is in the main Guassian elimination loop where the following calculations must be added into the original code.

```
a(j) = -mu*p;
  mu2 = c(n)/b(j);
  c(n) = -mu2*c(j);
  b(n) = b(n) - mu2*p;
  x(n) = x(n) - mu2*x(j);
  p = a(j);
```

Another consequence of the off-diagonal elements is that the (n-2)th and (n-1)th Guassian elimination step must be done separately from the primary loop. These two steps are given by the following lines of code.

```
%Guassian Elimination on the n-2 step
  b(n-1) = b(n-1) - (a(n-2)/b(n-2))*c(n-2);
  c(n-1) = c(n-1) - (a(n-2)/b(n-2))*a(n-3);
  x(n-1) = x(n-1) - (a(n-2)/b(n-2))*x(n-2);
%Bottom corner element on the n-2 step
  a(n-1) = a(n-1) - (c(n)/b(n-2))*c(n-2);
  b(n) = b(n) - (c(n)/b(n-2))*a(n-3);
  x(n) = x(n) - (c(n)/b(n-2))*x(n-2);
%Guassian Elimination on the n-1 step
  b(n) = b(n) - (a(n-1)/b(n-1))*c(n-1);
  x(n) = x(n) - (a(n-1)/b(n-1))*x(n-1);
```

The third consequence is that two steps of back substitution are required prior to using a back substitution loop which works for all but the last calculation, the first row of the matrix, which must also be done outside the loop. Calculations for the initialization steps, the loop and the first row are given by the following lines of code.

```
%Initializing the backsolving algorithm
  x(n) = x(n)/b(n);
  x(n-1) = (x(n-1) - c(n-1)*x(n))/b(n-1);
 for j = n-2:-1:2
  x(j) = (x(j) - c(j)*x(j+1) - a(j-1)*x(n))/b(j);
 end
%Backsolving for first row
  x(1) = (x(1) - c(1)*x(2) - a(n)*x(n))/b(1);
```

Once the d_k's are found, the information is passed back to `periodic_splinetx.m` to find the piecewise cubic polynomial interpolant on each subinterval. It then ensures that the u_k's are within the Δ period length, corrects them if necessary, and evaluates the spline at those points.

CHAPTER 4

GOVERNING FLUID EQUATIONS

Since we will later study the propagation of gravity waves on water it is important to understand the hydrodynamic equations governing such motion. These equations are well known and our approach for deriving them follows the derivations given in [Deb94] and [Birge].

4.1 Fluid Flow Beneath the Surface

Suppose that we have fluid moving with some velocity v and let $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ be the velocity vector field describing the motion of the fluid particles. Since water wave propagation it is reasonable for us to begin by assuming that the fluid is incompressible. This assumption gives us that

$$\text{div } \mathbf{v} = \nabla \cdot \mathbf{v} = 0. \tag{4.1}$$

The second assumption is that the fluid is irrotational in the interior, yielding

$$\text{curl } \mathbf{v} = \nabla \times \mathbf{v} = \mathbf{0}. \tag{4.2}$$

Because of the zero curl we know that the vector field $\mathbf{v}$ is conservative which allows us to express it as a potential function Φ such that

$$\mathbf{v} = \nabla \Phi. \tag{4.3}$$

This allows the zero divergence condition in equation 4.1 to be written as

$$\nabla^2 \Phi = \Delta \Phi = 0. \tag{4.4}$$

Thus the velocity potential for an inviscid ideal fluid flow satisfies Laplace's equation throughout the interior of the fluid at every point. Next we want to derive two important conditions on the free surface of the fluid, the kinematic and dynamic boundary conditions.

4.2 The Kinematic Free Surface Condition

Suppose S is the boundary surface of the fluid and that S depends continuously on time t. Then we can represent the surface by

$$S = S(x, y, z, t) = 0 \tag{4.5}$$

A primary characteristic of the free surface is that the normal velocity of the surface must be equal to the velocity of the fluid normal to the surface. Noting that $\mathbf{n} = \nabla S/|\nabla S|$ is the unit normal to the surface S, then the normal velocity of the surface is given by $S_t/|\nabla S|$ and the normal velocity of the fluid is given by $\mathbf{v} \cdot \mathbf{n}$. Setting these equations equal to each other yields

$$-\frac{S_t}{|\nabla S|} = \mathbf{v} \cdot \mathbf{n}.$$

Moving all terms to one side and multiplying by $|\nabla S|$ results with

$$\frac{dS}{dt} \equiv S_t + (\mathbf{v} \cdot \nabla)S = 0.$$

Not only does this give us how the surface changes with respect to time, it also shows that any fluid particle on the boundary surface will remain there. Since we are assuming not to have breaking waves, we can rewrite the surface equation as

$$S = \eta(x, y, t) - z = 0$$

and applying equation 4.3 we get

$$\begin{aligned} 0 &= \frac{\partial}{\partial t}\left(\eta(x, y, t) - z\right) + \left(\left\langle \frac{\partial \Phi}{\partial x}, \frac{\partial \Phi}{\partial y}, \frac{\partial \Phi}{\partial z} \right\rangle \cdot \left\langle \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right\rangle\right)\left(\eta(x, y, t) - z\right) \\ &= \frac{\partial \eta}{\partial t} + \frac{\partial \Phi}{\partial x}\frac{\partial \eta}{\partial x} + \frac{\partial \Phi}{\partial y}\frac{\partial \eta}{\partial y} - \frac{\partial \Phi}{\partial z}\frac{\partial z}{\partial z} = \frac{\partial \eta}{\partial t} + \frac{\partial \Phi}{\partial x}\frac{\partial \eta}{\partial x} + \frac{\partial \Phi}{\partial y}\frac{\partial \eta}{\partial y} - \frac{\partial \Phi}{\partial z} \end{aligned}$$

Thus, solving for η_t we get

$$\eta_t = \Phi_z - (\eta_x \Phi_x + \eta_y \Phi_y) \tag{4.6}$$

which is our kinematic free surface condition.

4.3 The Dynamic Free Surface Condition

Following the Lagrangian description, we begin by defining an infinitesimal fluid volume element $dV = dxdydz$ and then consider all the forces acting upon it. The first force is an external force exerted throughout the fluid by the gravitational field. This force is proportional to the volume's mass and so we have

$$\mathbf{g}\ \rho\ dV, \tag{4.7}$$

where $\mathbf{g}$ is the gravitational acceleration vector field and ρ is the fluid density.

The second force acting on our volume element dV is a net pressure force exerted by the surrounding media. Recall that pressure is a scalar field and so is independent of orientation. Pressure exerted on our volume element depends only on the force exerted on the volume element by the surrounding fluid. Thus for some surface with surface area dA in the fluid, pressure is the proportionality constant relating a normal vector to the surface with the normal force acting on the surface, that is

$$-\mathbf{F}_n = p\ \mathbf{n}\ dA \tag{4.8}$$

where the negative sign arises from the fact that the force is considered toward the surface element. Supposing dV is an infinitesimal box and letting S be its surface, to find the total force acting on dV we have to evaluate the surface integral

$$-d\mathbf{F} = \iint_S p\ \mathbf{n}\ dA. \tag{4.9}$$

Letting p be the pressure at the center of dV we can relate the pressures across each of the faces by using the local pressure gradients $\partial p/\partial x$, $\partial p/\partial y$ and $\partial p/\partial z$. Then the pressure force can be computed by evaluating the integral as a sum over the six faces as is shown in the following calculation.

$$
\begin{aligned}
-d\mathbf{F} &= [(p\ \mathbf{n})\ dA]_1 + [(p\ \mathbf{n})\ dA]_2 + [(p\ \mathbf{n})\ dA]_3 + [(p\ \mathbf{n})\ dA]_4 + [(p\ \mathbf{n})\ dA]_5 + [(p\ \mathbf{n})\ dA]_6 \\
&= \left(p - \frac{\partial p}{\partial x}\frac{dx}{2}\right)(-\mathbf{i})\ dydz + \left(p + \frac{\partial p}{\partial x}\frac{dx}{2}\right)\mathbf{i}\ dydz + \left(p - \frac{\partial p}{\partial y}\frac{dy}{2}\right)(-\mathbf{j})\ dxdz \\
&\quad + \left(p + \frac{\partial p}{\partial y}\frac{dy}{2}\right)\mathbf{j}\ dxdz + \left(p - \frac{\partial p}{\partial z}\frac{dz}{2}\right)(-\mathbf{k})\ dxdy + \left(p + \frac{\partial p}{\partial z}\frac{dz}{2}\right)\mathbf{k}\ dxdy \\
&= \left(\frac{\partial p}{\partial x}\mathbf{i} + \frac{\partial p}{\partial y}\mathbf{j} + \frac{\partial p}{\partial z}\mathbf{k}\right) dxdydz = \nabla p\ dV.
\end{aligned}
$$

So our net force is given by $-\nabla p\ dV$.

Combining the gravitational and pressure forces and applying Newton's Second Law of motion, $F = ma$, we have

$$\mathbf{g}\ \rho\ dV - \nabla p\ dV = \rho\ dV \frac{D\mathbf{v}}{Dt} \tag{4.10}$$

where $\rho\ dV$ is the mass of the fluid volume element and the derivative $D\mathbf{v}/Dt$, known as the total derivative or material derivative, is the instantaneous acceleration. Eliminating the common terms and rearranging yields the general Euler expression for fluid flow

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla p + \mathbf{g}. \tag{4.11}$$

Recalling that the gravity field is conservative, we can express it as a potential function f and group it with the pressure field. It is also safe for us to assume that density is uniform and so equation 4.11 becomes

$$\frac{D\mathbf{v}}{Dt} = -\nabla\left(\frac{p}{\rho} + f\right).$$

Expanding the total derivative results in

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v} = -\nabla\left(\frac{p}{\rho} + f\right) \tag{4.12}$$

Substituting the vector identity $\mathbf{v}\cdot\nabla\mathbf{v} = (\nabla\times\mathbf{v})\times\mathbf{v} + \nabla(\mathbf{v}\cdot\mathbf{v})/2$ into equation 4.12 yields

$$\frac{\partial \mathbf{v}}{\partial t} + (\nabla\times\mathbf{v})\times\mathbf{v} + \frac{1}{2}\nabla(\mathbf{v}\cdot\mathbf{v}) = -\nabla\left(\frac{p}{\rho} + f\right)$$

and combining all gradient terms on one side results in Euler's general equation for fluid motion

$$\frac{\partial \mathbf{v}}{\partial t} + (\nabla \times \mathbf{v}) \times \mathbf{v} = -\nabla \left(\frac{p}{\rho} + f + \frac{1}{2}(\mathbf{v} \cdot \mathbf{v}) \right). \tag{4.13}$$

Substitution of equations 4.2 and 4.3 into equation 4.13 yields,

$$\frac{\partial \nabla \Phi}{\partial t} = -\nabla \left(\frac{p}{\rho} + f + \frac{1}{2}(\Phi_x^2 + \Phi_y^2 + \Phi_z^2) \right)$$

and exchanging order of the gradient on the left-hand side and integrating directly results in

$$\frac{\partial \Phi}{\partial t} = -\frac{p}{\rho} - f - \frac{1}{2}(\Phi_x^2 + \Phi_y^2 + \Phi_z^2) + C(t). \tag{4.14}$$

where C(t) is the constant of integration. A relevant note, however unnecessary for our derivation, is that from here we could make an appropriate transformation to recover Bernoulli's equation. Instead we assume that the applied force from gravity arises solely from a uniform field and choose to measure the potential f from the equilibrium free surface. Thus the potential at the actual instantaneous free surface of the fluid is given by $f = g\eta$ where $\eta(x, y, t)$ is the displacement of the free surface from its equilibrium value. Finally, it is safe to assume that there is uniform atmospheric pressure on the surface implying that pressure is constant and allowing us to choose $C(t) = p/\rho$. This assumption yields the equation for the dynamic free surface condition

$$\Phi_t = -g\eta - \frac{1}{2}\left(\Phi_x^2 + \Phi_y^2 + \Phi_z^2\right). \tag{4.15}$$

The dynamic free surface conditions shows how the surface of the water reacts to the effects of gravity. Comparing the kinematic and dynamic free surface conditions, one can see that the motion of the surface of a fluid is explicitly determined by the motion inside the fluid.

CHAPTER 5

APPLICATIONS OF CONFORMAL MAPPING TO THE TIME DEPENDENT WATER WAVE PROBLEM

The dynamics of irrotational incompressible fluid flows, at least in the two-dimensional case, are a natural application for conformal mapping techniques. Primarily this is a result of the well known fact that solutions to Laplace's equation are invariant under a conformal map. So we have the following theorem given and proved in [De15].

Theorem 5.0.1. *Let $w = f(z) = u(x,y) + iv(x,y)$ be a conformal map from a region D in the $z-$plane to a region Ω in the $w-$plane. Let $U = U(u,v)$ be harmonic in Ω. Then*

$$\Delta_z U = |f'(z)|^2 \Delta_w U \tag{5.1}$$

where

$$\Delta_z := \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \qquad \textit{and} \qquad \Delta_w := \frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2}$$

Therefore, since $f'(z) \neq 0$, then $U := U(u(x,y), v(x,y))$ is also harmonic in D.

This invariance allows us to transplant the boundary value problem for the Laplace equation from a complicated domain to the unit disk domain or some other model domain and then solve it using a Fourier or Laurent Series method. With the velocity potential for an inviscid ideal fluid satisfying Laplace's equation throughout the interior of the fluid, if we can find a conformal map from the wave domain to a more ideal domain, such as the unit disk or annulus, we can utilize these methods to solve the velocity potential problem and then easily map them back to the wave domain.

Once we have a solution for the velocity potential problem we can then use this information to solve the differential equations describing the dynamic and kinematic free surface conditions for the two-dimensional wave given in standard Cartesian coordinates by

$$\Phi_t = -g\eta - \frac{1}{2}\left(\Phi_x^2 + \Phi_y^2\right) \qquad \text{and} \qquad \eta_t = \Phi_y - \eta_x \Phi_x$$

which are the reduction of the three-dimensional cases discussed in Chapter 4. After solutions for these equations are found we will have all the information needed to describe the fluid flow and wave propagation.

Using the process discussed above, we will now develop the theory and numerical solution for solving velocity potential problems for Stokes waves on water of both infinite and finite depth. The problem of Stokes waves on infinitely deep water has been handled before by Fornberg in [Forwave] and Birge in [Birge], nonetheless we will again address this problem as a special case of the more general finite depth problem. With a solution to the special case we will then build off of these methods to attempt to solve the potential problem for Stokes waves on finitely deep water.

5.1 The Stokes Wave

In the field of fluid mechanics, one of the oldest problems is that of steady progressive free-surface waves. In 1847 Stokes [Stok47], [Sch74] presented a solution based on an expansion of a perturbation in a parameter assumed to vary monotonically with the *wave height/wave length* ratio. When re-examining the problem again in 1880, Stokes [Sch74] discovered that the calculations could be made simpler by choosing the independent quantities to be the velocity potential and stream functions. Following this approach, we wish to construct a mathematical representation for a Stokes wave similar to Schwartz' in [Sch74]. Suppose that we have an infinitely long train of symmetrical two-dimensional waves on top of a fluid of finite depth moving with a constant speed c. We also suppose that each wave has a period of 2π. We assume that the bottom is horizontal with a depth d and consider a reference frame centered on a single wave crest where the $y-$axis is the line of symmetry and the $x-$axis is restricted to the symmetric interval $[-\pi, \pi]$. Following the assumptions made from the derivation of the fluid equations we maintain that the fluid is inviscid, incompressible and irrotational. These assumptions imply that the solution can be represented as an analytic function of the complex variable $z = x + iy$. One cycle of the wave is shown in figure 5.1.

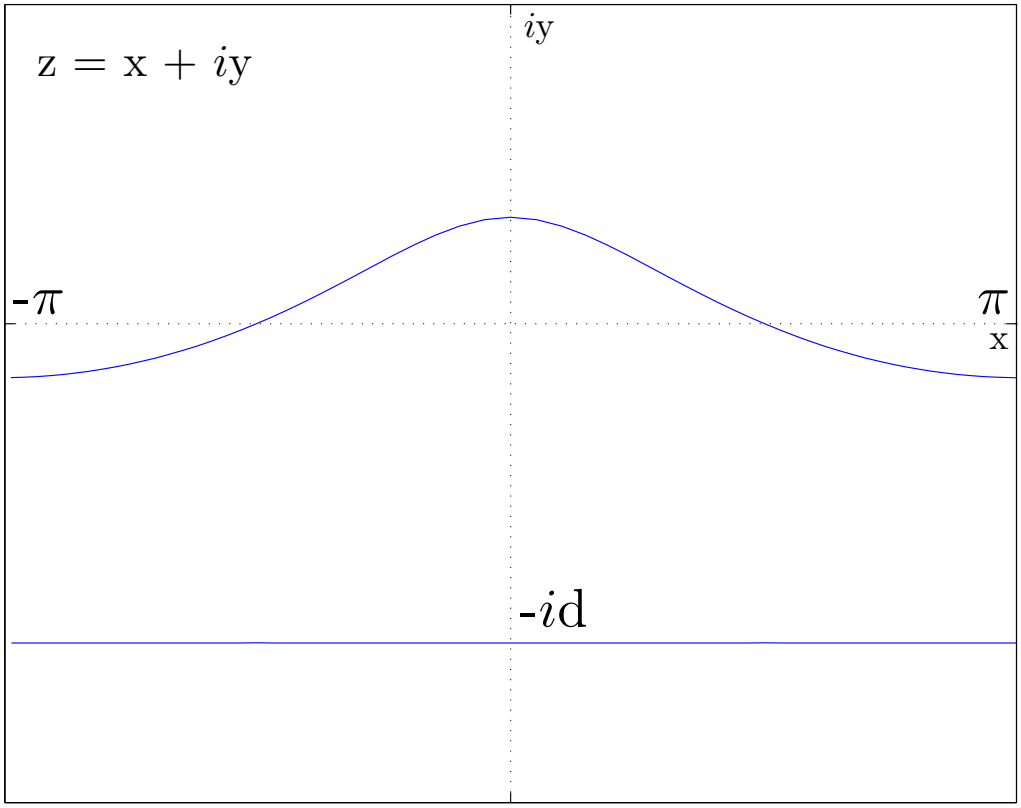

Figure 5.1: Water Wave with Finite Depth

Attempting to solve the problem in the physical domain would be challenging since the shape of the wave is unknown. Instead we will conformally map the interior of the wave in the $z-$plane into an annulus with unit outer radius in the $\zeta-$plane. To achieve this we will consider a composition of several conformal maps. First we apply the exponential map

$$w = e^{-iz}$$

mapping the interior of the fluid wave in the $z-$plane to the interior of a doubly connected region in the $w-$plane where the horizontal bottom and wave are mapped to the inner and outer boundaries respectively. The radius of the inner boundary is given by r and ϕ will be the polar angle. Since the inner boundary is circular any point p on the boundary can be written as $p = re^{i\phi}$ and note that because the horizontal bottom is given by $z = x - id$ we

have that the map of the bottom $w = e^{-iz} = e^{-ix-d} = e^{-d}e^{-ix}$ implying that $r = e^{-d}$ and $x = -\phi$.

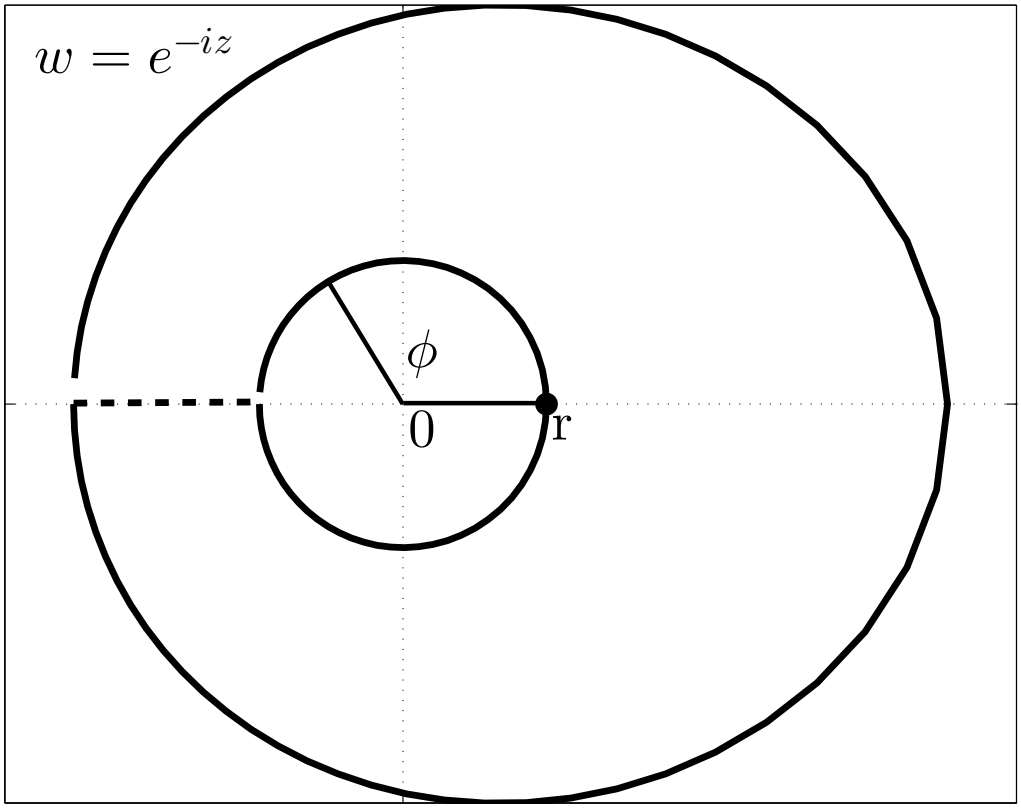

Figure 5.2: Doubly Connected Region

The second map we apply comes from an analogue to the Riemann Mapping Theorem allowing us to find a conformal mapping between our doubly connected region in the $w-$plane and the annulus with unit outer radius in the $\zeta-$plane. Let $\zeta = f^{-1}(w)$ be the map which maps the interior and boundaries of the doubly connected region into the interior and corresponding boundaries of the annulus. Figure 5.3 shows the annular region with the inner boundary radius r_0 and polar angle θ. Thus any point q on the inner boundary can be written as $q = r_0 e^{i\theta}$.

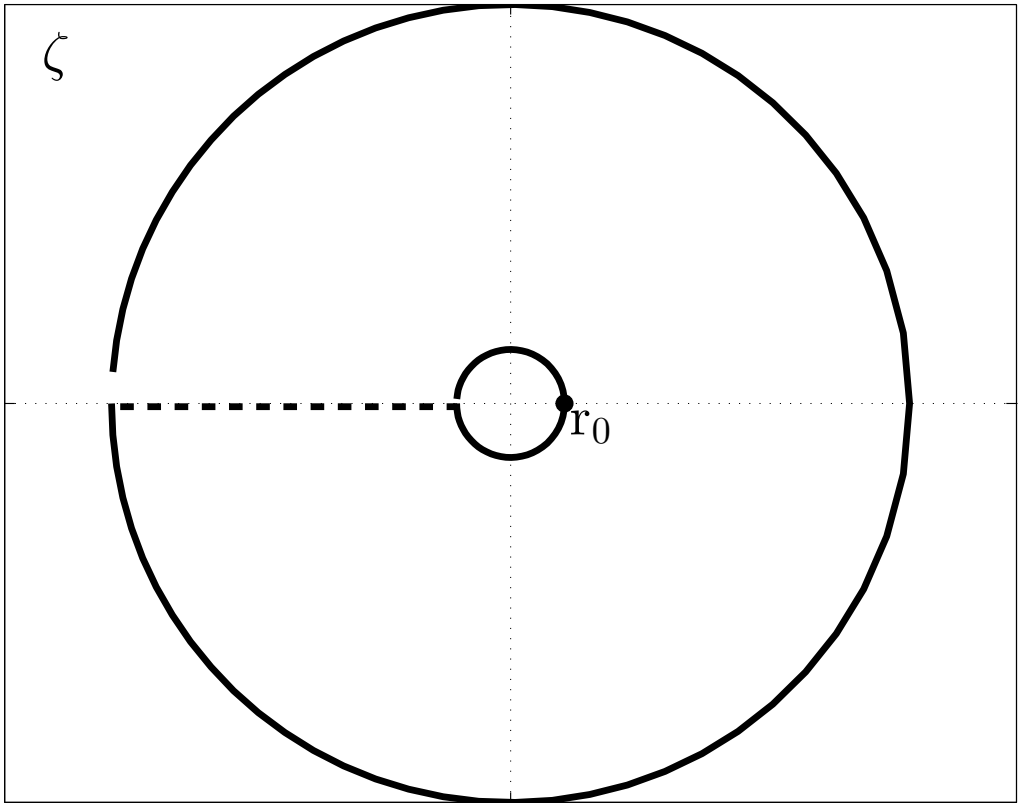

Figure 5.3: Unit Annulus

Since f^{-1} is a bijective function, we can find an inverse function $f = f(\zeta) = w$ mapping the annulus back to the doubly connected region. Using this function we consider some of the symmetries on the annular and doubly connected regions that might simplify later calculations. Since the inner boundaries of both regions are circular and the wave in the $z-$plane is symmetric about the $y-$axis, they both have a symmetry in terms of the conjugate, so

$$f(r_0 e^{i\theta}) = re^{i\phi(\theta)} \qquad \text{and} \qquad \overline{f(r_0 e^{i\theta})} = f(r_0 e^{-i\theta}) = re^{i\phi(-\theta)} = re^{-i\phi(\theta)}$$

implying that ϕ is an odd function since $\phi(-\theta) = -\phi(\theta)$ and also periodic in θ since $\phi(\pi) = \pi$ and $\phi(-\pi) = -\pi$. The annular region is also symmetric on the outer boundary $|\zeta| = 1$. Because f is analytic we can expand it as a power series in ζ, that is $f = \sum_{k=-\infty}^{\infty} c_k \zeta^k$.

Setting $\zeta = e^{i\theta}$ yields

$$f(e^{i\theta}) = \sum_{k=-\infty}^{\infty} c_k e^{ik\theta} = \overline{f(e^{i\theta})} = \sum_{k=-\infty}^{\infty} \overline{c_k}\overline{e^{-ik\theta}} = \sum_{k=-\infty}^{\infty} \overline{c_k} e^{ik\theta},$$

implying that $c_k = \overline{c_k}$ and thus the coefficients are real numbers. One would expect for this relationship to hold for the mapping of the inner boundary as well, and indeed we have

$$f(r_0 e^{i\theta}) = \sum_{k=-\infty}^{\infty} c_k r_0^k e^{ik\theta} = \overline{f(r_0 e^{i\theta})} = \sum_{k=-\infty}^{\infty} \overline{c_k r_0^k e^{-ik\theta}} = \sum_{k=-\infty}^{\infty} \overline{c_k r_0^k} e^{ik\theta}.$$

Another consequence of having the function f is that we can recover the wave in the physical $z-$plane by composing the inverses of the maps we have applied, that is a mapping from the annulus in $\zeta-$plane to the wave in the $z-$plane is given by

$$z = i \log f(\zeta). \tag{5.2}$$

Since we know that $z = x - id$ on the bottom of the physical plane, then we have

$$z = x - id = i \log f(r_0 e^{i\theta}) = i \log r e^{i\phi(\theta)} = -\phi(\theta) + i \log r$$

implying that $x\big|_{\pi}^{-\pi} = -\phi(\theta)\big|_{-\pi}^{\pi}$ and $d = -\log r$ or alternatively $r = e^{-d}$ which is what we calculated earlier. This also shows that the depth $d \to \infty$ as $r \to 0$. Next, we note that in general $z = i \log f(\zeta)$ is not periodic in θ for $\zeta = e^{i\theta}$ or $\zeta = r_0 e^{i\theta}$. However, we want any representation of the wave to be periodic and so if we rewrite equation (5.1)

$$z = i \log f(\zeta) = i \log \zeta + i \log \frac{f(\zeta)}{\zeta} \tag{5.3}$$

we see that the last term is periodic in θ for $\zeta = e^{i\theta}$ or $\zeta = r_0 e^{i\theta}$. Again setting this equal to the value we have for the bottom of the water, yields

$$\begin{aligned} x - id = i \log f(r_0 e^{i\theta}) &= i \log r_0 e^{i\theta} + i \log \frac{f(r_0 e^{i\theta})}{r_0 e^{i\theta}} \\ &= -\theta + i \log r_0 + i \log \frac{r}{r_0} e^{i\phi(\theta) - \theta} \\ &= -\theta + i \log r_0 + i \log r - i \log r_0 - [\phi(\theta) - \theta] \end{aligned}$$

where the last term on the right is clearly periodic since $\phi(\pm\pi) = \pm\pi$. Recalling that ϕ is also an odd function of θ we define that term as

$$\begin{aligned} h(\theta) := i \log \frac{f(r_0 e^{i\theta})}{r_0 e^{i\theta}} &= i \log \frac{r}{r_0} - [\phi(\theta) - \theta] \\ &= i \log \frac{r}{r_0} + \sum_{k=1}^{\infty} \widehat{h}_k \sin k\theta \\ &= i \log \frac{r}{r_0} + \sum_{k=1}^{\infty} \widehat{h}_k \frac{e^{ik\theta} - e^{-ik\theta}}{2i} \\ &= i \log \frac{r}{r_0} - i \sum_{k=1}^{\infty} \left[\frac{1}{2} \widehat{h}_k e^{ik\theta} - \frac{1}{2} \widehat{h}_k e^{-ik\theta} \right]. \end{aligned} \tag{5.4}$$

Taking the Laurent expansion of the last term in equation (5.2) yields

$$F(\zeta) := i \log \frac{f(\zeta)}{\zeta} = i \sum_{k=-\infty}^{\infty} \alpha_k \zeta^k,$$

which is analytic in the annulus $r_0 < |z| < 1$ since $f(\zeta) \neq 0$. Substituting values for ζ corresponding to the inner and outer boundaries of the annulus results with

$$H(\theta) = F(e^{i\theta}) = i \log \frac{f(e^{i\theta})}{e^{i\theta}} = i \sum_{k=-\infty}^{\infty} \alpha_k e^{ik\theta} \tag{5.5}$$

and

$$h(\theta) = F(r_0 e^{i\theta}) = i \log \frac{f(r_0 e^{i\theta})}{r_0 e^{i\theta}} = i \sum_{k=-\infty}^{\infty} \alpha_k r_0^k e^{ik\theta}. \tag{5.6}$$

Comparing equations (5.3), (5.4) and (5.5) we get

$$\alpha_0 = \log \frac{r}{r_0} \qquad \text{with} \qquad \alpha_k r_0^k = \frac{1}{2} \widehat{h}_k \qquad \text{and} \qquad \alpha_{-k} r_0^{-k} = -\frac{1}{2} \widehat{h}_k \quad \text{for} \quad k = 1, 2, 3, \ldots,$$

implying that $\alpha_{-k} = -\dfrac{1}{2} r_0^k \widehat{h}_k = -r_0^{2k} \alpha_k$.

Therefore we have found that for any ζ,

$$\begin{aligned} z = z(\zeta) &= i \log f(\zeta) \\ &= i \left[\log \zeta + \log \frac{f(\zeta)}{\zeta} \right] \\ &= i \left[\log \zeta + \sum_{k=-\infty}^{\infty} \alpha_k \zeta^k \right] \\ &= i \left[\log \zeta + \log \frac{r}{r_0} + \sum_{k=1}^{\infty} \alpha_k \zeta^k - \alpha_k r_0^{2k} \zeta^{-k} \right] \\ &= i \left[\log \zeta + \log \frac{r}{r_0} + \sum_{k=1}^{\infty} \alpha_k \left(\zeta^k - \frac{r_0^{2k}}{\zeta^k} \right) \right] \end{aligned} \tag{5.7}$$

and taking $\alpha_k = a_k/k$ we get equation (2.3) from [Sch74] representing the mapping from the $\zeta-$plane to the $z-$plane,

$$z(\zeta) = i \left[\log \zeta + \log \frac{r}{r_0} + \sum_{k=1}^{\infty} \frac{a_k}{k} \left(\zeta^k - \frac{r_0^{2k}}{\zeta^k} \right) \right]. \tag{5.8}$$

So, from equation (5.6) on the inner circle of the annulus with $|\zeta| = r_0$ we see that

$$\begin{aligned} z(r_0 e^{i\theta}) &= i \log f(r_0 e^{i\theta}) \\ &= i \left[\log r_0 e^{i\theta} + \log \frac{r}{r_0} + \sum_{k=1}^{\infty} \frac{a_k}{k} \left(r_0^k e^{ik\theta} - \frac{r_0^{2k}}{r_0^k e^{ik\theta}} \right) \right] \\ &= i \left[\log r_0 + i\theta + \log r - \log r_0 + \sum_{k=1}^{\infty} \frac{a_k}{k} r_0^k \left(e^{ik\theta} - e^{-ik\theta} \right) \right] \\ &= i \left[\log r e^{i\theta} + 2i \sum_{k=1}^{\infty} \frac{a_k}{k} r_0^k \sin k\theta \right] \\ &= -\theta - 2 \sum_{k=1}^{\infty} \frac{a_k}{k} r_0^k \sin k\theta - i(-\log r) \\ &= -\theta - 2 \sum_{k=1}^{\infty} \frac{a_k}{k} r_0^k \sin k\theta - id. \end{aligned} \tag{5.9}$$

This implies that $z(r_0 e^{\pm i\pi}) = \mp\pi - id$ which is what we have calculated previously. Now, on the outer circle of the annulus with $|\zeta| = 1$ we see that

$$\begin{aligned}
z(e^{i\theta}) &= i \log f(e^{i\theta}) \\
&= i \left[\log e^{i\theta} + \log \frac{1}{r_0} + \sum_{k=1}^{\infty} \frac{a_k}{k} \left(e^{ik\theta} - r^{2k} e^{-ik\theta} \right) \right] \\
&= -\theta - i \log r_0 + i \sum_{k=1}^{\infty} \frac{a_k}{k} \left[\left(1 - r_0^{2k} \right) \cos k\theta + i \left(1 + r_0^{2k} \right) \sin k\theta \right] \\
&= -\theta - \sum_{k=1}^{\infty} \frac{a_k}{k} \left(1 + r_0^{2k} \right) \sin k\theta + i \left[- \log r_0 + \sum_{k=1}^{\infty} \frac{a_k}{k} \left(1 - r_0^{2k} \right) \cos k\theta \right] \\
&= x + iy
\end{aligned}$$

This is gives us a representation of the Stokes wave in terms of sine and cosine series as shown by equations (3.1 a,b) in [Sch74] where

$$-x = \theta + \sum_{k=1}^{\infty} \frac{a_k}{k} \sigma_k \sin k\theta, \qquad \text{where} \qquad \sigma_k := 1 + r_0^{2k} \tag{5.10}$$

and

$$y = -\log r_0 + \sum_{k=1}^{\infty} \frac{a_k}{k} \delta_k \cos k\theta, \qquad \text{where} \qquad \delta_k := 1 - r_0^{2k}. \tag{5.11}$$

5.2 Time Dependent Water Wave with an Infinite Depth

The first case we will consider is the problem of time dependent wave propagation on deep water. Although this problem has been addressed previously by Bengt Fornberg in [Forwave] and Jonathan Birge in [Birge], we will examine it here as a type of "special case" of the finite depth case. Again suppose that we have an infinite train of 2π-periodic Stokes waves on the surface of deep water, that is, water with 'infinite depth' or depth that is much greater than the wavelength of the wave. Recalling that as the depth d of the water approaches infinity the value for r_0 approaches zero and in this limit we get the following

equations defining our Stokes wave

$$-x = \theta + \sum_{k=1}^{\infty} \frac{a_k}{k} \sin k\theta \qquad \text{and} \qquad y = \sum_{k=1}^{\infty} \frac{a_k}{k} \cos k\theta \tag{5.12}$$

with an initial velocity potential given by Φ. For specific coefficients a_k and $\theta \in [-\pi, \pi]$, these equations give us the physical wave in the $z-$plane. Wanting to study the propagation of waves defined by equation 5.12 we recall the hydrodynamic equations and note that attempting to solve these in the physical plane would be challenging. Instead, following similarly to the methods we used to derive the Stokes wave equations and taking advantage of the invariance of solutions to Laplace's equation under conformal mapping, we will map the wave into the unit disk to make calculations simpler. So, consider the image of the $z-$plane under the exponential map $w(z) = e^{-iz}$ giving us a simply connected region as shown in figure 5.4.

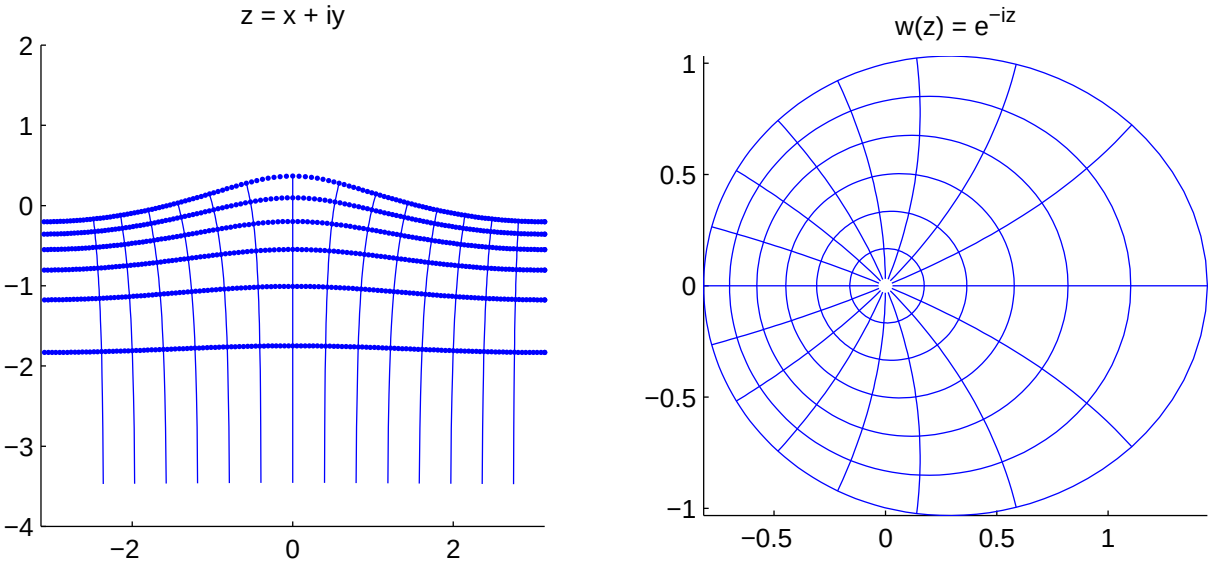

Figure 5.4: Mapping of the Physical Wave to a Simply Connected Region

Next we want to construct an ideal domain which will simplify our calculations for the velocity potential problem. Taking the still water case in the $v-$plane where $x-$values are on the interval of $[-\pi, \pi]$ and $y = 0$, we map it to the unit disk in the $\zeta-$plane by $\zeta(v) = e^{-iv}$ as shown in figure 5.5.

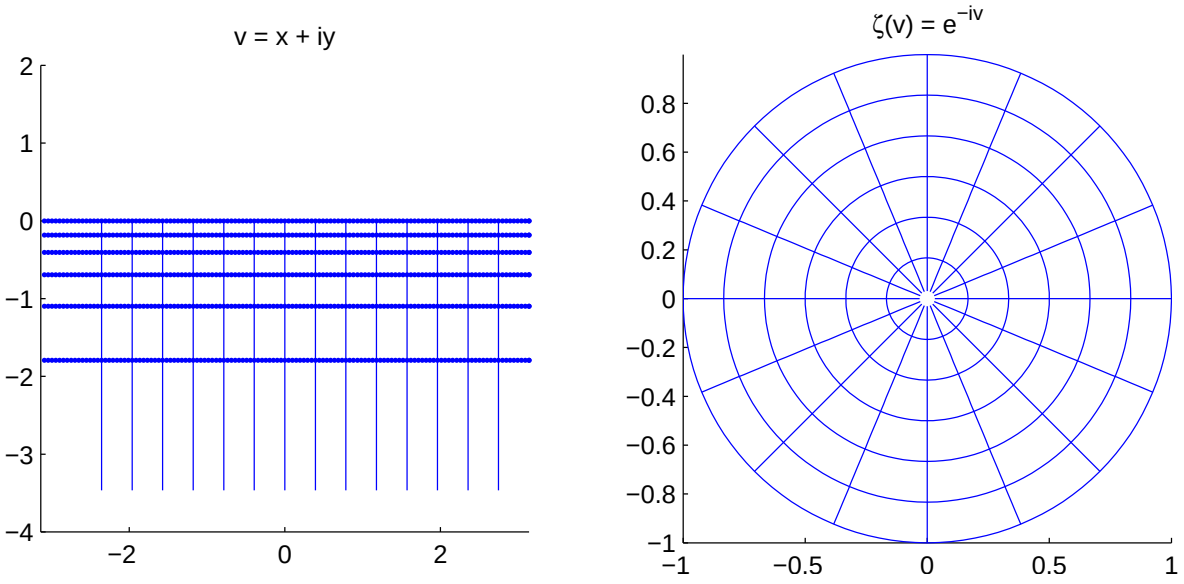

Figure 5.5: Mapping of Still Water to Unit Disk

The last step is finding a conformal map between the simpily connected region and the unit disk. Riemann's mapping theorem guarantees that there exists a biholomorphic mapping f between the unit disk in the $\zeta-$plane and the simply connected region in the $w-$plane shown in figure 5.6.

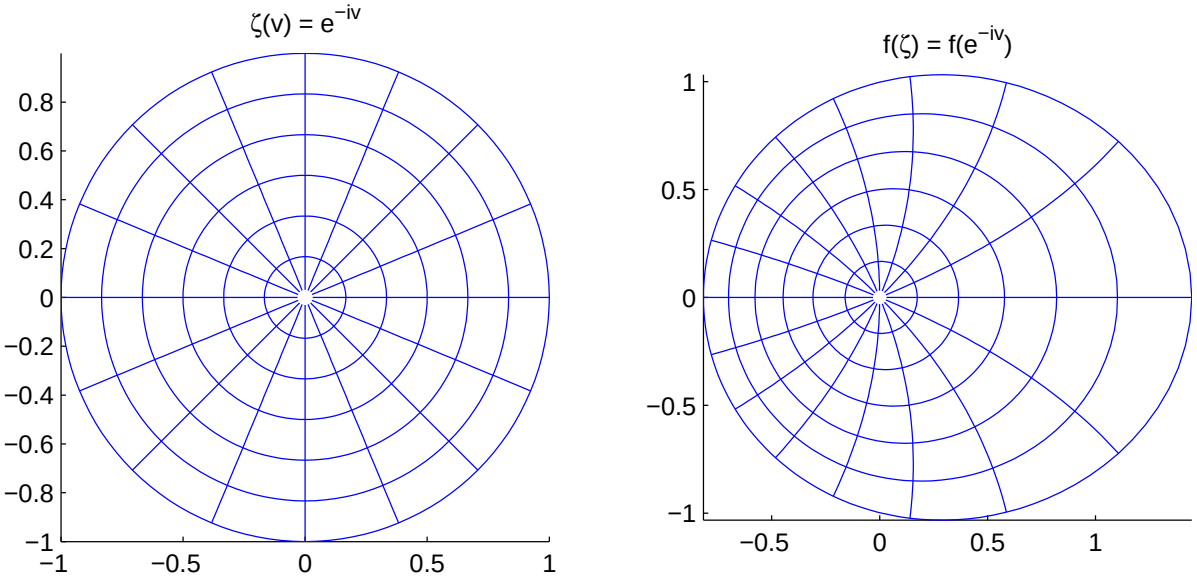

Figure 5.6: Conformal Map Between the Unit Disk and the Simply Connected Region

Thus allowing us to recover the physical wave in the $z-$plane by the familiar equation $z = z(v) = i \log f(e^{-iv})$ where $v \in [-\pi, \pi]$ and with the total mapping shown in figure 5.7.

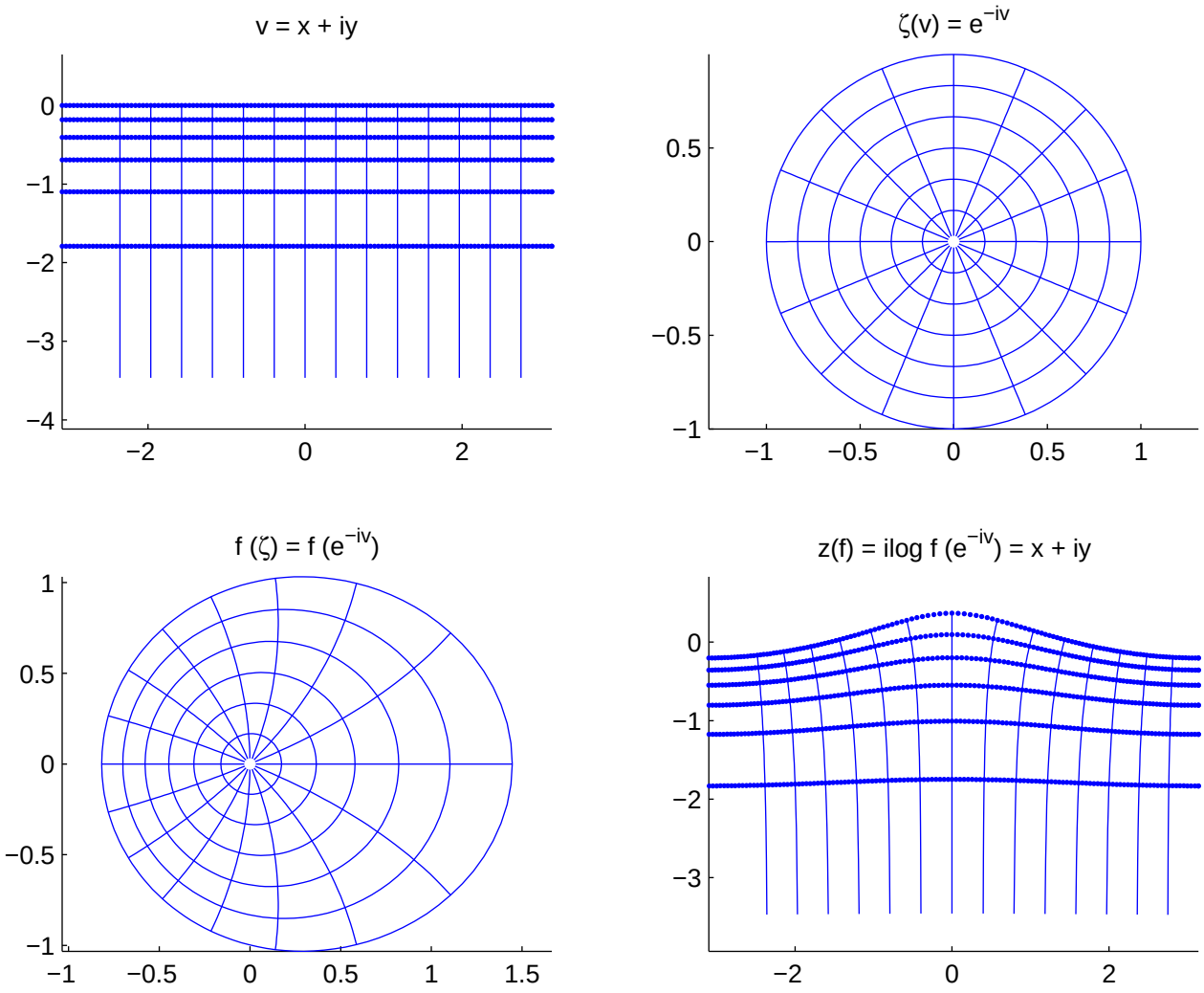

Figure 5.7: Geometry of the Total Conformal Map

Since motion on the surface of a fluid is determined by what is happening inside the fluid and recalling that the potential function satisfies Laplace's equation throughout the interior of the fluid, we need to find a solution to the velocity potential problem in order solve the time dependent wave problem. Luckily, since the solutions to the Laplace equation

are invariant under conformal mapping, rather than attempting to solve the problem in the physical plane we can solve it on the unit disk.

5.2.1 The Dirichlet Problem on the Unit Disk

The solution to the Dirichlet problem for Φ on the unit disk is a basic result found in most partial differential equations textbooks and here we follow the discussion given by McOwen in [Mc03]. For convenience we will express Φ in terms of polar coordinates. Thus we want to find $\Phi = \Phi(r, \theta)$ solving

$$\begin{cases} \Delta\Phi(r,\theta) = \dfrac{\partial^2\Phi}{\partial r^2} + \dfrac{1}{r}\dfrac{\partial\Phi}{\partial r} + \dfrac{1}{r^2}\dfrac{\partial^2\Phi}{\partial\theta^2} = 0 & \text{for} \quad 0 \le r < 1 \\ \Phi(1,\theta) = \widehat{\Phi}(\theta) & \text{for} \quad r = 1 \end{cases} \tag{5.13}$$

where $\widehat{\Phi}$ is a real function on the boundary. Making the substitution $r = e^{-t}$ and setting $u(r, \theta) = X(t)Y(\theta)$ we use the method of separation of variables to solve obtain a solution. We get that the solution is a Fourier expansion of the boundary function.

$$\Phi(1,\theta) = \widehat{\Phi}(\theta) = a_0 + \sum_{k=1}^{\infty} a_k \cos k\theta + b_k \sin k\theta \tag{5.14}$$

on the boundary, and

$$\Phi(r,\theta) = a_0 + \sum_{k=1}^{\infty} a_k r^k \cos k\theta + b_k r^k \sin k\theta = a_0 + \sum_{k=1}^{\infty} r^k (a_k \cos k\theta + b_k \sin k\theta) \tag{5.15}$$

on the interior. Since $r^k \cos k\theta$ and $r^k \sin k\theta$ are harmonic, the Laplace equation is linear and our solution to the Dirichlet problem is unique. Writing equations 5.14 and 5.15 in terms of complex numbers for convenience, we have

$$\widehat{\Phi}(\theta) = c_0 + \sum_{k=-\infty}^{\infty} c_k e^{ik\theta} \qquad \text{and} \qquad \Phi(r,\theta) = c_0 + \sum_{k=-\infty}^{\infty} c_k r^{|k|} e^{ik\theta}$$

with $c_k = \overline{c_{-k}}$ since we have a real-valued solution.

5.2.2 Solving the Dynamic and Kinematic Boundary Conditions

Now that we have a solution for the velocity potential, we can use this to solve the dynamic and kinematic boundary conditions. Recalling that the equations describing these

conditions both rely on the partial derivatives of the potential function Φ with respect to both x and y, we need to find the respective partials in terms of polar coordinates. Taking the partials with respect to r and θ on the boundary we get

$$\widehat{\Phi}_r = \Phi_r\Big|_{r=1} = \frac{\partial \Phi}{\partial r}\Big|_{r=1} = \frac{\partial}{\partial r}\left[c_0 + \sum_{k=-\infty}^{\infty} c_k r^{|k|} e^{ik\theta}\right]_{r=1} = \sum_{k=-\infty}^{\infty} c_k |k| e^{ik\theta}$$

and

$$\widehat{\Phi}_\theta = \Phi_\theta\Big|_{r=1} = \frac{\partial \Phi}{\partial \theta}\Big|_{r=1} = \frac{\partial}{\partial \theta}\left[c_0 + \sum_{k=-\infty}^{\infty} c_k r^{|k|} e^{ik\theta}\right]_{r=1} = \sum_{k=-\infty}^{\infty} c_k (ik) e^{ik\theta}.$$

Finally, we need to find the derivative of the total conformal mapping function as seen in equation 5.1. With the total mapping function defined as $z(\theta) = i \log\left(f(e^{-i\theta})\right)$, this derivative is given by

$$\frac{dz}{d\theta} = \frac{d}{d\theta}\left[i \log\left(f(e^{-i\theta})\right)\right] = i\, \frac{f'(e^{-i\theta})}{f(e^{-i\theta})}(-i)e^{-i\theta} = \frac{f'(e^{-i\theta})}{f(e^{-i\theta})} e^{-i\theta}. \tag{5.16}$$

Therefore, if we can find the conformal map f and an initial $\widehat{\Phi}$ we can solve both surface conditions giving us the instantaneous rates of change with respect to time for both the surface η and velocity potential Φ. Rather than attempting to explicitly calculate these by hand we will instead follow a numerical approach very similar to the construction we have made here utilizing Fornberg's method to find the conformal mapping function f and Euler's method to solve the differential equations describing the boundary conditions.

5.2.3 Numerical Approach

Rather than solving our problem explicitly, we wish to approach the time dependent water wave problem of infinite depth numerically. The approach remains very similar and we will closely follow the methods used in our previous explanation. It is important to note that the methods and code used by the author are edited from methods and code provided by Fornberg in [Forwave] and Birger in [Birge], respectively. The algorithms referenced here are available in Appendix B. Beginning with algorithm B.1, `simply_connected_stokes_final.m`, we start by initializing variables and including the

coefficients to define the Stokes wave given by the vector at. In one loop of this algorithm we start with constructing the Stokes wave by the following.

```
for l = 1:nt,
      arg = l*phi;
      x = x + at(l)*sin(arg);
      y = y + at(l)*cos(arg);
     end}
```

Note that this process also yields the initial value for our potential function $\Phi =$ `phi`. Next we put the physical wave onto the complex $z-$plane by `z = x + iy` and then map it to a simply connected region in the $\zeta-$plane by the exponential map `zeta = exp(-i*z)`.
The real and imaginary parts are separated and used as inputs for Fornberg's method which finds a conformal map `zeta` between the unit disk and the simply connected region by the following lines.

```
xp = real([zeta zeta(1)]); yp = imag([zeta zeta(1)]);
[zeta,e,s,h,erri,tl] = fornintsp_4_td(n,8,xp,yp,sinit);
```

We can then find the physical wave by inverting our maps using `z = i*log(zeta)`.
Finally, the potential function Φ is evaluated at the conformal points using a cubic spline by `phi = periodic_splinetx(s_spl,[phi phi(1)],s)` concluding one loop. This process is then repeated four times because Fornberg's original paper notes that it was important for the points to be in conformal locations prior to being passed to the Euler's method algorithm.
Once the points are in conformal locations, we pass the data to `eulerwave_final.m`, given by algorithm B.2, to solve the three main fluids equations and update the boundary correspondences and potential function. The first part of the algorithm initializes variables and then we begin the first pass through our loop. We proceed through the conformal mapping process once again. Recalling that our solution to the Dirichlet problem for the disk was

determined by Fourier series, `fft(phi)` applies the fast Fourier transform to the conformal Phi solving the problem for our given data. The next two lines find the partial derivatives for Φ with respect to r and θ, respectively.

```
phi_a = -real(ifft(1i*w.*Phi));
phi_b = real(ifft(abs(w).*Phi));
```

This is done in the same manner as the following calculation. The truncated series for $f(z)$ using Fourier points are

$$f(z) = \sum_{k=0}^{\frac{n}{2}+1} c_k z^k \qquad \text{and} \qquad f'(z) = \sum_{k=0}^{\frac{n}{2}+1} k\ c_k z^{k-1}. \tag{5.17}$$

Setting $z = e^{-iv}$ and multiplying by e^{-iv} we get

$$f'(e^{-iv})e^{-iv} = \sum_{k=0}^{\frac{n}{2}+1} k\ c_k e^{-i(k-1)v} e^{-iv} = \sum_{k=1}^{\frac{n}{2}+1} k\ c_k e^{-ikv} = \text{fft}[f(e^{-iv})] \tag{5.18}$$

The next line,

```
dzdv = -1i*n*ifft(1i*(0:n-1).*[c(1:n/2+1)  zeros(1, n/2-1)])./zeta;
```

calculates the derivative of the total conformal mapping function needed from 5.1. We then use a basic Euler scheme to solve the differential equations describing the kinematic and dynamic free surface boundary conditions. More specifically, we will be solving for the change of the surface points in time Δt given by the following equations.

$$\begin{aligned} \Delta\Phi &= \left(-g\eta + \frac{1}{2}(\Phi_x^2 + \Phi_y^2)\right)\Delta t \\ \Delta x &= \Phi_x \Delta t \\ \Delta y &= \Phi_y \Delta t \end{aligned}$$

This is given numerically by the following,

```
phi = (-g.*imag(z)+0.5*(phi_a.^2 + phi_b.^2)./abs(dzdv).^2)*dt + phi;
z = exp(1i*angle(dzdv)).*(phi_a + 1i*phi_b)./abs(dzdv)*dt + z;
```

which gives us the updates for boundary correspondence points and the potential function we need and completes one time step for the wave propagation algorithm. The following are outputs from an experiment where one period is completed after 1000 time steps.

We begin by looking at the wave progression for time steps m = 200, in red, and m = 400, in black, while comparing it to the original wave.

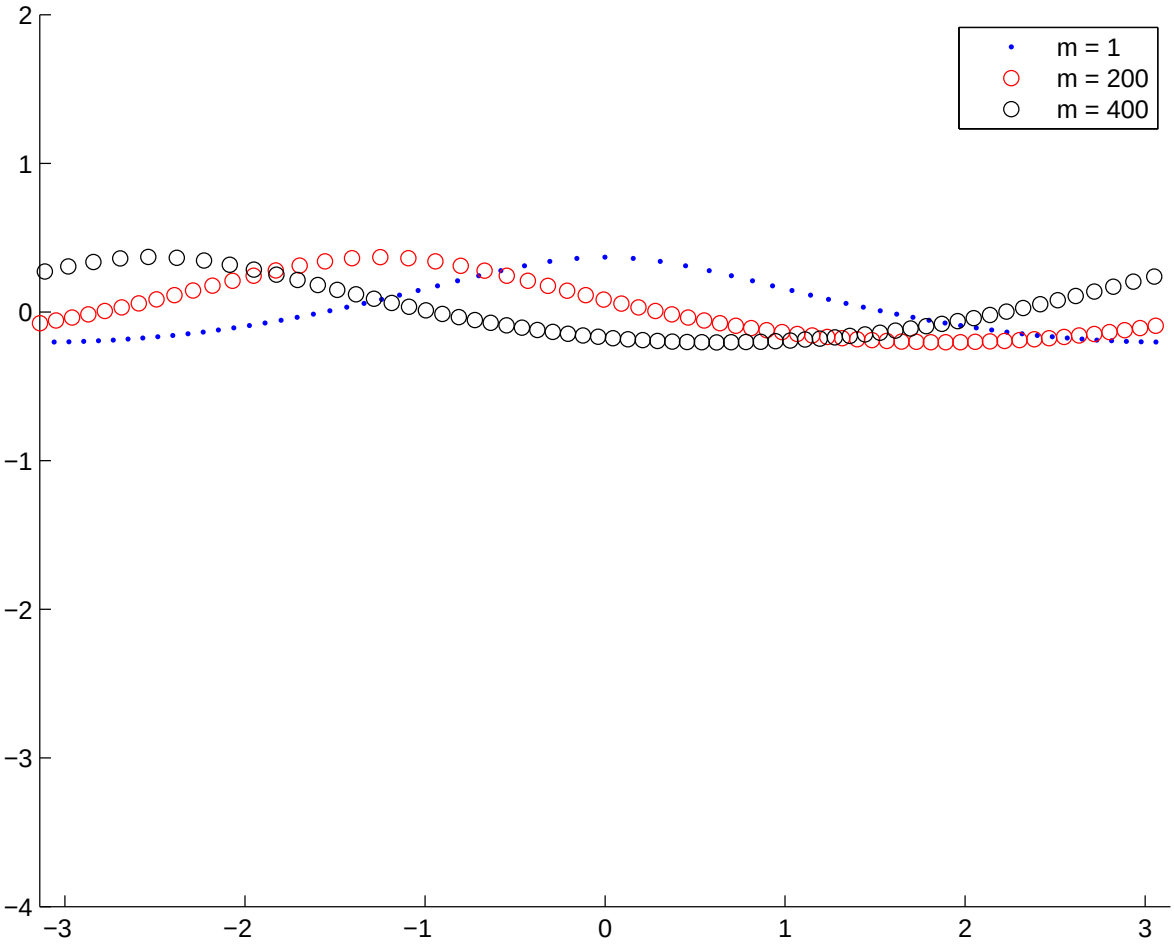

Figure 5.8: Wave Evolution after Time Steps m = 1, m = 200, m=400

Notice that the wave progresses to the left as time increases, which relates to the negative velocity potential function. We now compare different time steps for the wave along with their related conformal maps.

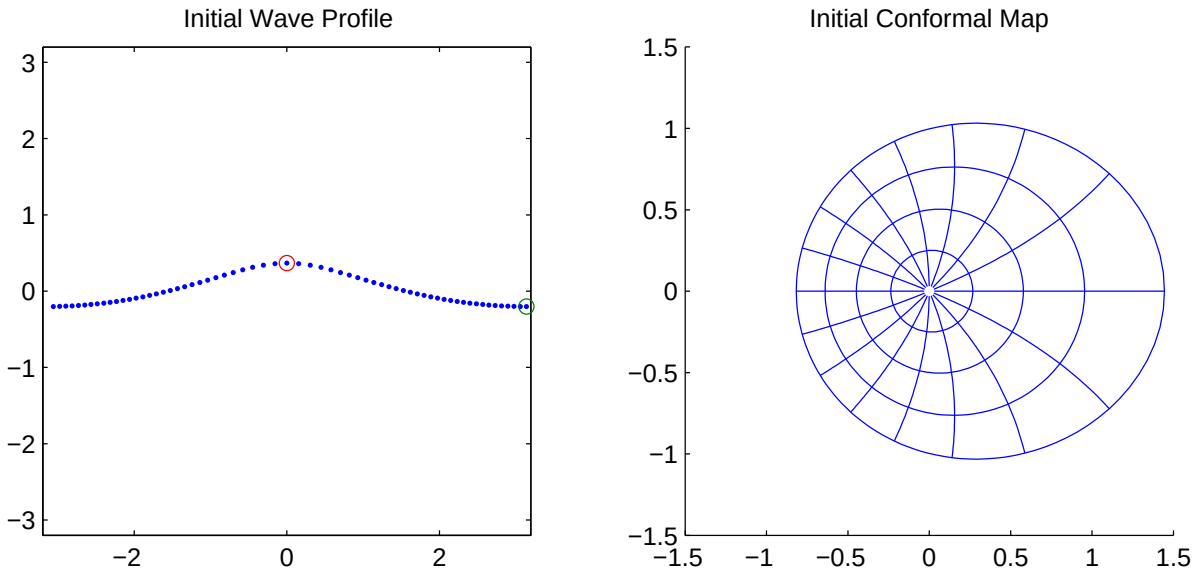

Figure 5.9: Initial Wave and Conformal Map

In figure 5.9 we see the initial wave with a marking indicating the first and center elements by the green and red circles respectively and its corresponding conformal map. Figure 5.10 gives us the wave and map after 100 time steps.

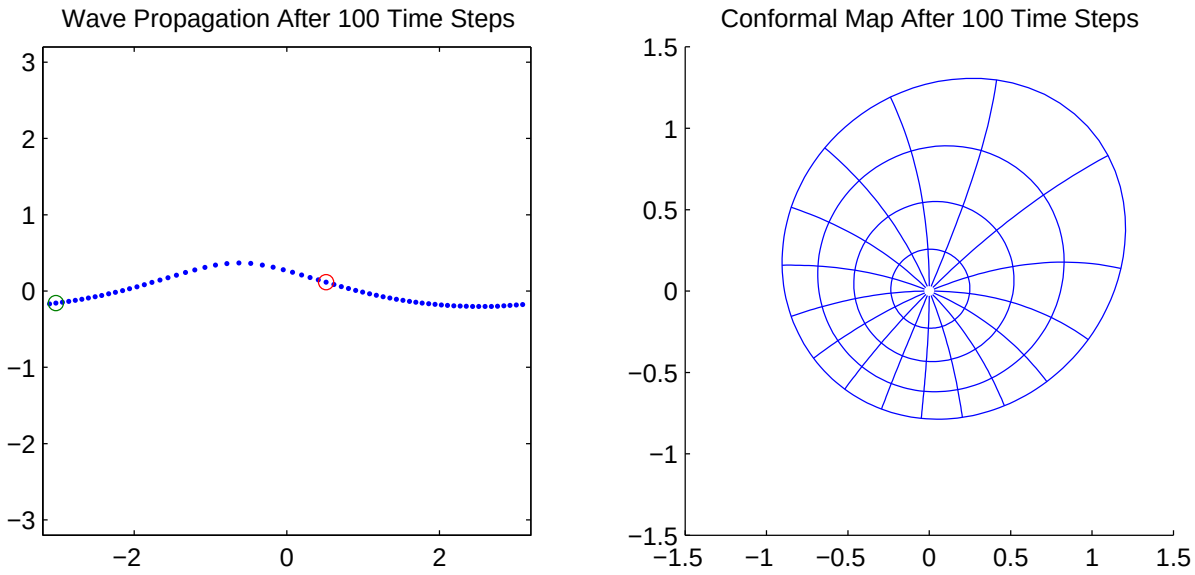

Figure 5.10: Wave and Conformal Map for m = 100

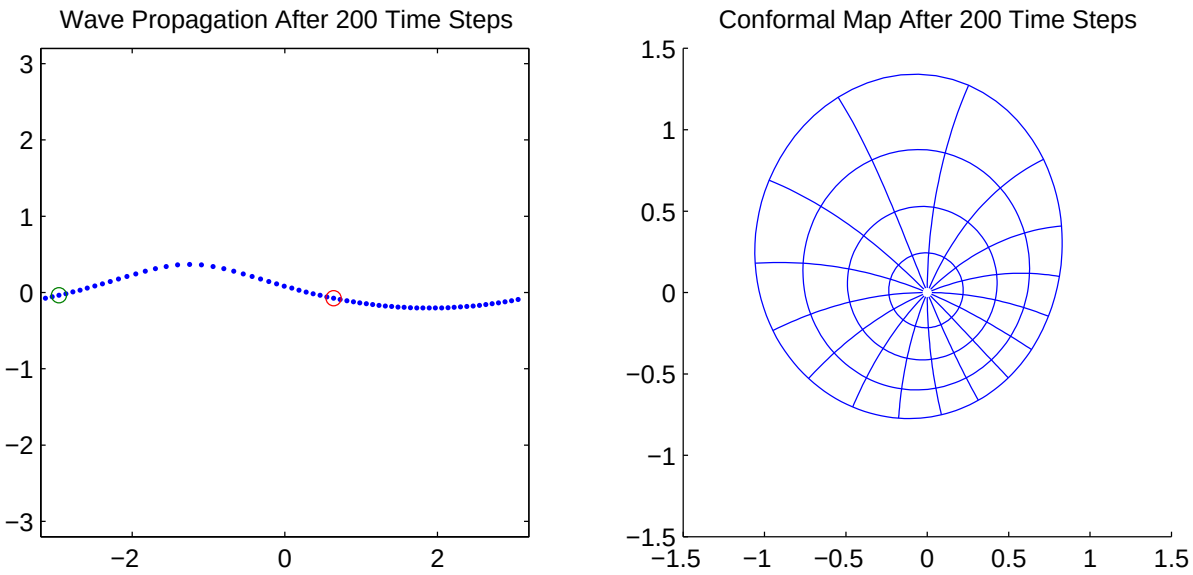

Figure 5.11: Wave and Conformal Map for m = 200

Figures 5.11 and 5.12 shows the wave and conformal map after 200 and 300 time steps, respectively. Notice that the rate of rotation of the simply connected region and its boundary are different, indicating the differences in motion between the surface particles and the particles inside the fluid.

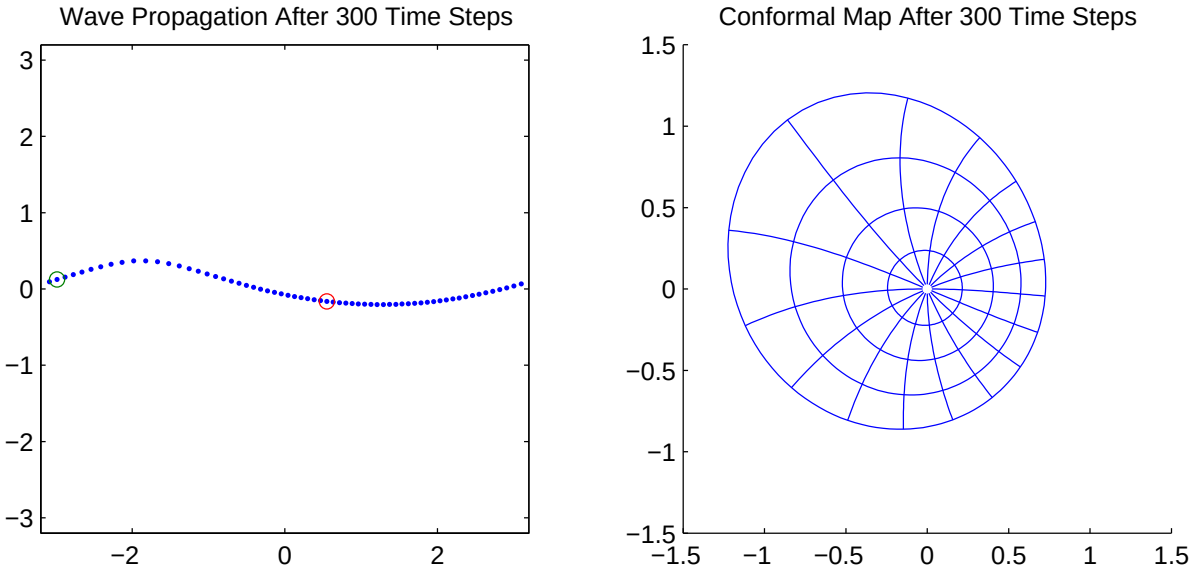

Figure 5.12: Wave and Conformal Map for m = 300

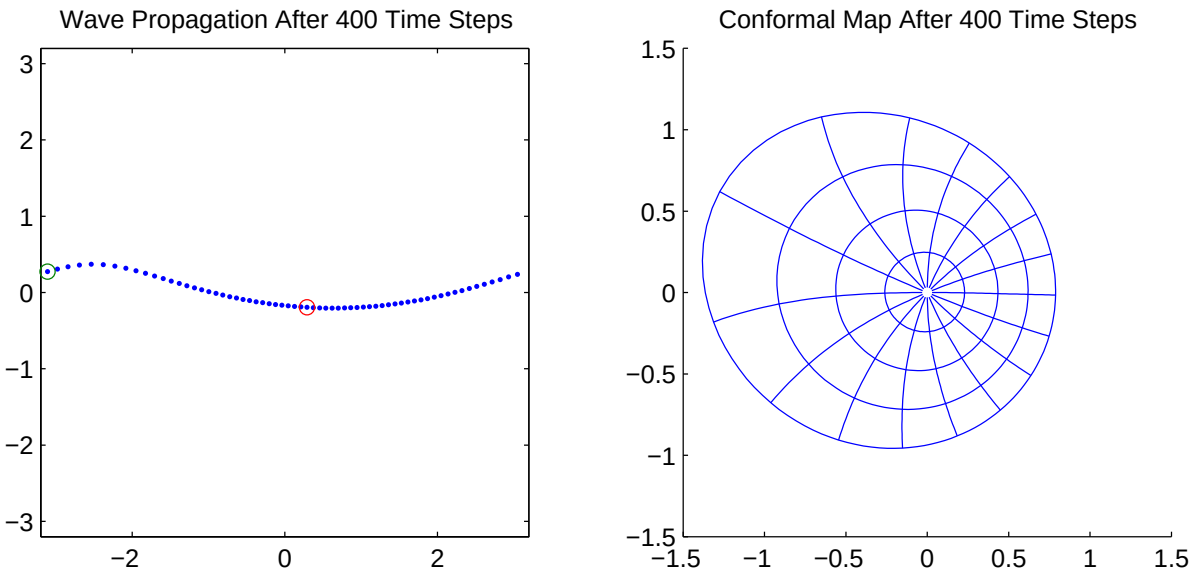

Figure 5.13: Wave and Conformal Map for m = 400

Figures 5.13 and 5.14 shows the wave and conformal map after 400 and 500 timesteps, respectively. After 500 time steps the wave has traveled half of a period as is indicated by the graph of the wave being centered on the trough. The graph of the conformal map gives us a better indication of the inequality in motion of the bounded region and its boundary.

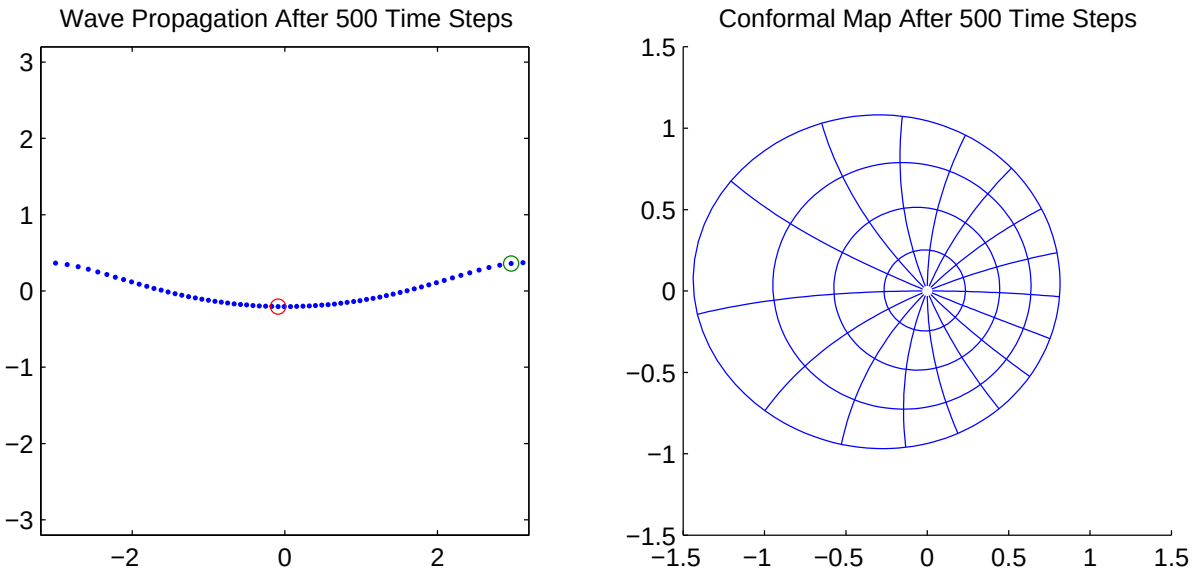

Figure 5.14: Wave and Conformal Map for m = 500

5.3 Time Dependent Water Wave with a Finite Depth

The second case we will consider is the problem of time dependent water wave propagation on water with a finite depth d. We have already derived the equations describing the $2\pi-$periodic Stoke's wave with a finite depth in Section 5.1 as were given by equations (5.7, 5.8)

$$-x = \theta + \sum_{k=1}^{\infty} \frac{a_k}{k}(1 + r_0^{2k}) \sin k\theta \qquad \text{and} \qquad y = \sum_{k=1}^{\infty} \frac{a_k}{k}(1 - r_0^{2k}) \cos k\theta.$$

With the initial velocity potential Φ. An analogous argument to that given in Section 5.1 is made here. We start by mapping the wave to a doubly connected region using the exponential map shown in figure 5.15.

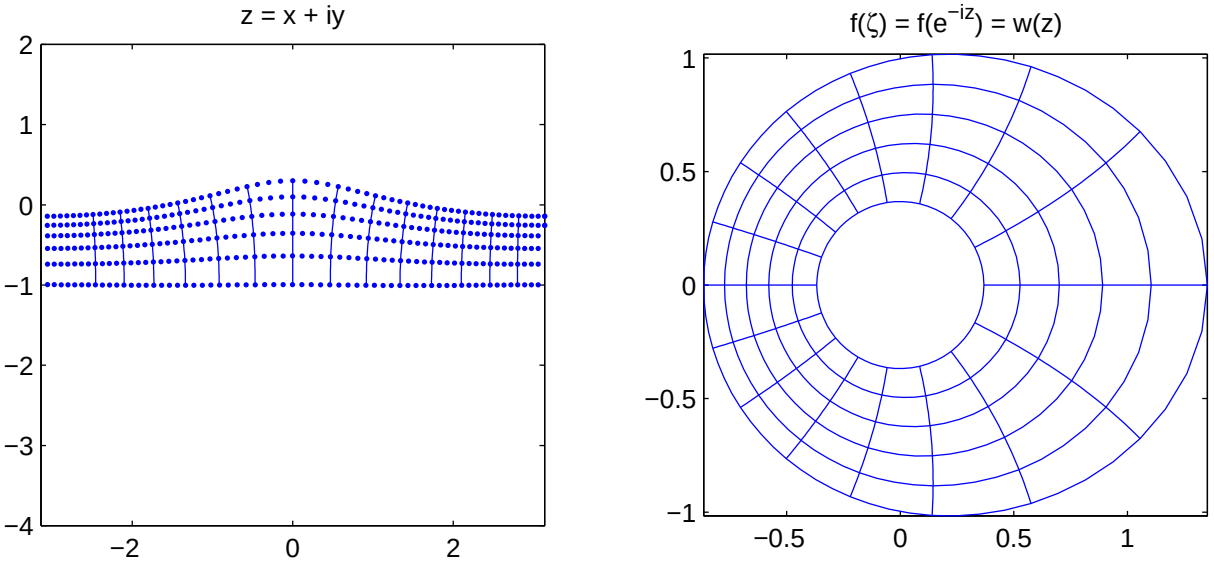

Figure 5.15: Mapping of Physical Wave to Doubly Connected Region

Next, we find a conformal correspondence between the annulus and the double connected region as shown in figure 5.16. The total mapping function is the the same as the infinite depth case, given by $z = i \log f(e^{-i\theta})$ for $\theta \in [-\pi, \pi]$, with the total mapping shown by figure 5.17.

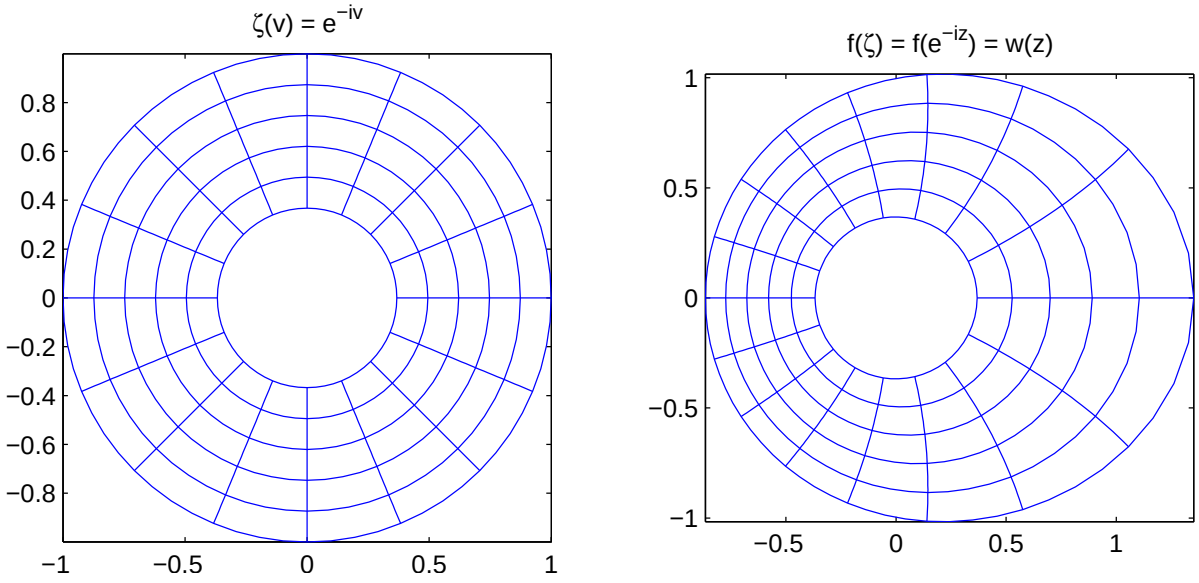

Figure 5.16: Conformal Map from the Annulus to the Doubly Connected Region

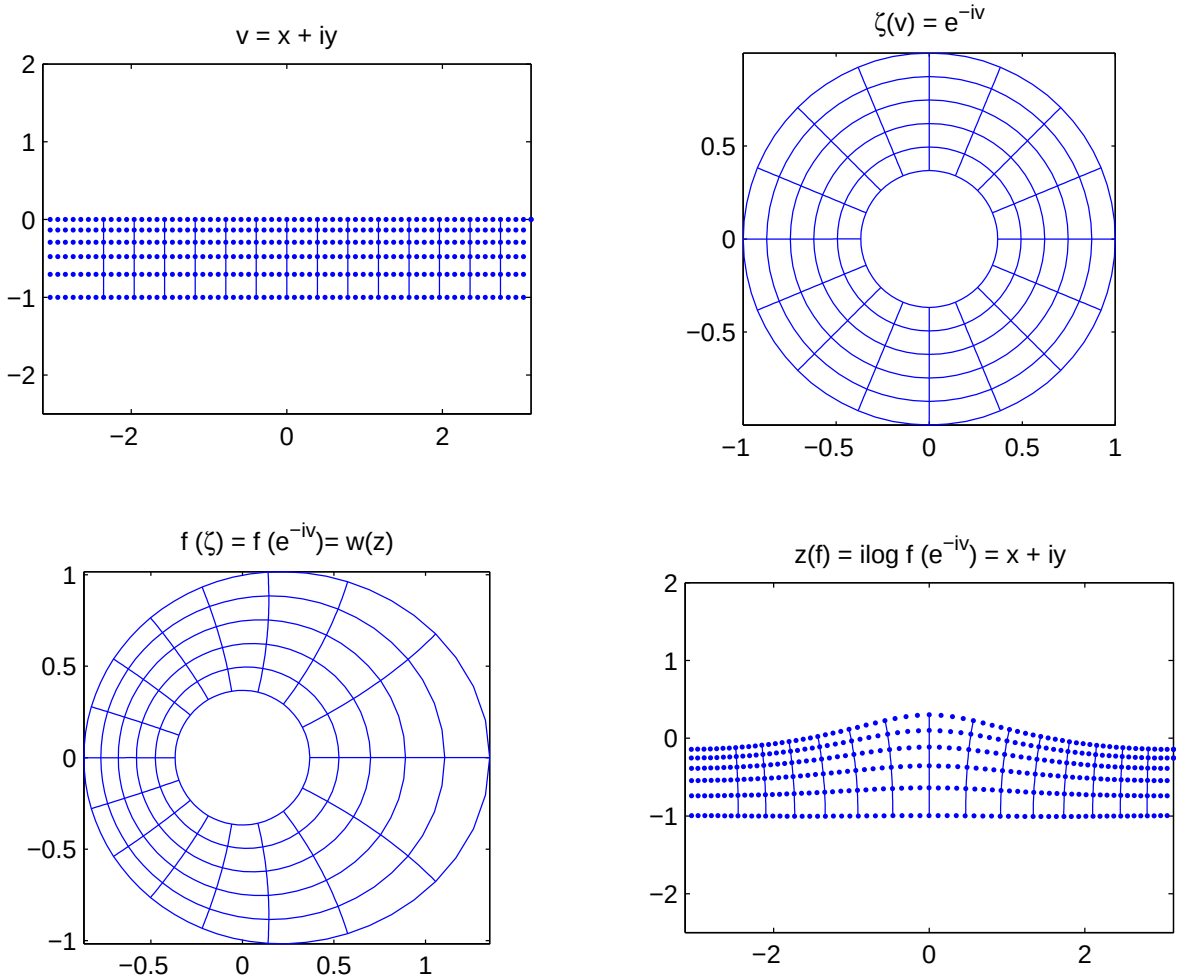

Figure 5.17: Geometry of the Total Conformal Map

So instead of solving for the potential in the unit disk as we did in the last case, we now have to solve the Dirichlet problem on the annulus, $r_0 \leq |z| \leq 1$.

5.3.1 The Dirichlet Problem on the Annulus

Let us consider the Dirichlet problem for Φ on the annulus. Recall that in the case of infinite depth, the only conditions were the periodic boundary data. However, in the case of finite depth we must account for the fact that no water will pass through the bottom. This gives us a Neumann condition on the bottom, which in turn becomes a Neumann condition on the inner boundary of the annulus under the conformal map. Again we express Φ in terms of polar coordinates and so we are trying to find $\Phi(r, \theta)$ solving

$$\begin{cases} \Delta\Phi(r,\theta) = \dfrac{\partial^2\Phi}{\partial r^2} + \dfrac{1}{r}\dfrac{\partial\Phi}{\partial r} + \dfrac{1}{r^2}\dfrac{\partial^2\Phi}{\partial\theta^2} = 0 & \text{for} \quad \rho < r < 1 \\ \Phi(1,\theta) = \widehat{\Phi}(\theta) & \text{for} \quad r = 1 \\ \dfrac{\partial\Phi}{\partial r} = 0 & \text{for} \quad r = \rho, \quad 0 < \rho < 1 \end{cases} \tag{5.19}$$

where $\widehat{\Phi}$ is a real function on the outer boundary. We wish to construct a solution similar to the one we found for the unit disk case. We begin by trying to find possible solutions $u = u(r, \theta)$ such that u satisfies

$$\Delta u(r,\theta) = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial\theta^2} = 0. \tag{5.20}$$

First, consider $u_1 = r^n \cos n\theta \quad$ with $\quad n = \pm 1, \pm 2, \pm 3, \ldots$ and the partials

$$(u_1)_r = nr^{n-1}\cos n\theta, \qquad (u_1)_{rr} = n(n-1)r^{n-2}\cos n\theta, \qquad (u_1)_{\theta\theta} = -n^2 r^n \cos n\theta.$$

Substituting these values into equation 5.20 yields

$$r^2\Delta u_1 = \big(n(n-1)r^n + nr^n - n^2 r^n\big)\cos n\theta = (n^2 - n^2)r^n \cos n\theta = 0$$

which shows $u_1 = r^n \cos n\theta$ is a solution to equation 5.20, i.e. harmonic in the annulus. Likewise for $u_2 = r^n \sin n\theta \quad$ with $\quad n = \pm 1, \pm 2, \pm 3, \ldots$ and the partials

$$(u_2)_r = nr^{n-1}\sin n\theta, \qquad (u_2)_{rr} = n(n-1)r^{n-2}\sin n\theta, \qquad (u_2)_{\theta\theta} = -n^2 r^n \sin n\theta,$$

we quickly see that $u_2 = r^n \sin n\theta$ is also a solution to equation 5.20. Thus, any linear combination of the form $u = u_1 \pm iu_2 = r^n(\cos n\theta \pm i \sin n\theta)$ with $n = \pm 1, \pm 2, \pm 3, \ldots$ will be a solution to equation 5.20. Rewriting this in exponential form we have $u = r^{\pm k} e^{\pm(\mp)ik\theta}$, for k an integer.

Finally consider $u_3 = \log(r)$ with the partials

$$(u_3)_r = r^{-1}, \qquad (u_3)_{rr} = -r^{-2}, \qquad (u_3)_{\theta\theta} = 0,$$

we can again quickly see that this is also a solution to equation 5.20.

Combining these solutions by superposition we have a possible solution for Φ in the annulus of the form

$$\Phi(r,\theta) = d_0 + a \log r + \sum_{k=-\infty;\, k\neq 0}^{\infty} \left(d_k r^k + e_k r^{-k}\right) e^{ik\theta} \tag{5.21}$$

We now attempt to rewrite the coefficients in more convenient terms by enforcing the conditions in equation 5.19. On the outer boundary, that is when $r = 1$, we have that Φ must satisfy

$$\Phi(1,\theta) = d_0 + \sum_{k=-\infty;\, k\neq 0}^{\infty} (d_k + e_k)\, e^{ik\theta} = c_0 + \sum_{k=-\infty;\, k\neq 0}^{\infty} c_k e^{ik\theta} = \widehat{\Phi}(\theta) \tag{5.22}$$

where the second series is the Fourier series expansion of $\widehat{\Phi}$ with Fourier coefficients c_k. Since $\widehat{\Phi}$ is real then $c_k = \overline{c}_{-k}$. This equality implies that $d_0 = c_0$ and $c_k = d_k + e_k$.

Next we examine the inner boundary condition by first taking the partial of equation 5.21 for r

$$\begin{aligned}
\frac{\partial}{\partial r}\Phi(r,\theta) &= \frac{a}{r} + \sum_{k=-\infty;\, k\neq 0}^{\infty} \left(k\, d_k r^{k-1} - k\, e_k r^{-k-1}\right) e^{ik\theta} \\
&= \frac{a}{r} + \sum_{k=-\infty;\, k\neq 0}^{\infty} k\, r^{-k-1} \left(d_k r^{2k} - e_k\right) e^{ik\theta}.
\end{aligned}$$

On the inner boundary, that is when $r = \rho$, the partial derivative must satisfy

$$\left.\frac{\partial \Phi}{\partial r}\right|_{r=\rho} = \frac{a}{\rho} + \sum_{k=-\infty;\, k\neq 0}^{\infty} k\, \rho^{-k-1} \left(d_k \rho^{2k} - e_k\right) e^{ik\theta} = 0. \tag{5.23}$$

Rewriting this equation we have

$$-\frac{a}{\rho} = \sum_{k=-\infty;\, k\neq 0}^{\infty} k\ \rho^{-k-1}\left(d_k\rho^{2k} - e_k\right) e^{ik\theta}$$

however a/ρ is a constant and the right side of the equation is a series of oscillating functions implying that $a = 0$ since $\rho \neq 0$. So equation 5.23 simplifies to

$$\sum_{k=-\infty;\, k\neq 0}^{\infty} k\ \rho^{-k-1}\left(d_k\rho^{2k} - e_k\right) e^{ik\theta} = 0,$$

implying that $d_k\rho^{2k} - e_k = 0$ since the other terms in the series are nonzero. Combining these equations with the two equations gained from equation 5.22, we have that the coefficients must satisfy the following conditions:

$$d_0 = c_0, \quad c_k = d_k + e_k, \quad a = 0, \quad \text{and} \quad e_k = \rho^{2k} d_k. \tag{5.24}$$

From these conditions we see that $c_k = d_k + \rho^{2k} d_k = (1 + \rho^{2k}) d_k$ which, combined with the last equality in equation 5.24, results in

$$d_k = \frac{1}{1+\rho^{2k}} c_k \quad \text{and} \quad e_k = \frac{\rho^{2k}}{1+\rho^{2k}} c_k. \tag{5.25}$$

Thus we can write the coefficients d_k and e_k in terms of our Fourier coefficients c_k from $\widehat{\Phi}$. So our solution given by equation 5.21 becomes

$$\begin{aligned}
\Phi(r,\theta) &= c_0 + \sum_{k=-\infty;\, k\neq 0}^{\infty} \left(\frac{r^k}{1+\rho^{2k}} + \frac{\rho^{2k}\ r^{-k}}{1+\rho^{2k}}\right) c_k e^{ik\theta} \\
&= c_0 + \sum_{k=-\infty;\, k\neq 0}^{\infty} \left(\frac{1+\rho^{2k} r^{-2k}}{1+\rho^{2k}}\right) r^k c_k e^{ik\theta} \\
&= c_0 + \sum_{k=-\infty;\, k\neq 0}^{\infty} \left(\frac{1+(\rho/r)^{2k}}{1+\rho^{2k}}\right) r^k c_k e^{ik\theta}.
\end{aligned} \tag{5.26}$$

Indeed, this solution satisfies the conditions from equation 5.19 since

$$\begin{aligned}
\Phi(1,\theta) &= c_0 + \sum_{k=-\infty;\, k\neq 0}^{\infty} \left(\frac{1+\rho^{2k}}{1+\rho^{2k}}\right) c_k e^{ik\theta} \\
&= c_0 + \sum_{k=-\infty;\, k\neq 0}^{\infty} c_k e^{ik\theta} = \widehat{\Phi}(\theta)
\end{aligned}$$

and

$$\begin{aligned}
\left.\frac{\partial \Phi}{\partial r}\right|_{r=\rho} &= \sum_{k=-\infty;\, k\neq 0}^{\infty} \left(\frac{k r^{k-1} - k\rho^{2k}\, r^{-k-1}}{1+\rho^{2k}}\right) c_k e^{ik\theta}\Bigg|_{r=\rho} \\
&= \sum_{k=-\infty;\, k\neq 0}^{\infty} \left(\frac{k \rho^{k-1} - k\rho^{2k}\, \rho^{-k-1}}{1+\rho^{2k}}\right) c_k e^{ik\theta} \\
&= \sum_{k=-\infty;\, k\neq 0}^{\infty} \left(\frac{k \rho^{k-1} - k\rho^{2k-k-1}}{1+\rho^{2k}}\right) c_k e^{ik\theta} \\
&= \sum_{k=-\infty;\, k\neq 0}^{\infty} \left(\frac{k \rho^{k-1} - k\rho^{k-1}}{1+\rho^{2k}}\right) c_k e^{ik\theta} = 0
\end{aligned}$$

Therefore we have found a potential function Φ solving the boundary value problem given by equation (5.17). Since the components of our series are harmonic, the solution is unique.

5.3.2 Solving the Dynamic and Kinematic Boundary Conditions

As in the infinite depth case since we now have a solution for the velocity potential allowing us to solve the dynamic and kinematic boundary conditions as long as we can find the necessary pieces of information. Starting by taking the partials with respect to r and θ we get

$$\Phi_r = \sum_{k=-\infty;\, k\neq 0}^{\infty} k \left(\frac{r^{k-1} - \rho^{2k} r^{-k-1}}{1+\rho^{2k}}\right) c_k e^{ik\theta}$$

and

$$\begin{aligned}
\Phi_\theta &= \frac{\partial}{\partial \theta}\left[c_0 + \sum_{k=-\infty;\, k\neq 0}^{\infty} \left(\frac{r^k + \rho^{2k} r^{-k}}{1+\rho^{2k}}\right) c_k e^{ik\theta}\right] \\
&= \sum_{k=-\infty;\, k\neq 0}^{\infty} ik \left(\frac{r^k + \rho^{2k} r^{-k}}{1+\rho^{2k}}\right) c_k e^{ik\theta}.
\end{aligned}$$

Taking $r = 1$ and writing these in terms of our boundary function $\widehat{\Phi}$ we have

$$\widehat{\Phi}_r = \sum_{k=-\infty;\, k\neq 0}^{\infty} k \left(\frac{1-\rho^{2k}}{1+\rho^{2k}}\right) c_k e^{ik\theta} \qquad \text{and} \qquad \widehat{\Phi}_\theta = \sum_{k=-\infty;\, k\neq 0}^{\infty} ikc_k e^{ik\theta}.$$

Notice that the result for $\widehat{\Phi}_\theta$ is the same as the disk case. Now, let us look closer at Φ_r inside the annulus. For $0 < \rho < 1$ and $k < 0$ we have

$$\frac{1-\rho^{2k}}{1+\rho^{2k}} = \frac{\rho^{-2k}-1}{\rho^{-2k}+1} = -\frac{1-\rho^{2k}}{1+\rho^{2k}} = -\frac{1-\rho^{2|k|}}{1+\rho^{2|k|}}$$

Therefore, multiplying be $k < 0$ we have

$$k\frac{1-\rho^{2k}}{1+\rho^{2k}} = k\frac{1-\rho^{2|k|}}{1+\rho^{2|k|}}$$

Combining this result with our earlier result yields

$$\widehat{\Phi}_r = \sum_{k=-\infty;\, k\neq 0}^{\infty} k\left(\frac{1-\rho^{2|k|}}{1+\rho^{2|k|}}\right) c_k e^{ik\theta}$$

on the annulus.

Finally, the derivative of the total mapping function $dz/d\theta$ will still be given by

$$z = z(\theta) = i\log f(e^{-i\theta}) \quad \text{for} \quad \theta \in [-\pi, \pi].$$

but it is important to note that the conformal mapping functions f are not the same. Again, rather than computing the solutions for our free surface conditions directly, we will attempt to approach this problem numerically by mirroring the approach used for the infinite depth case.

5.3.3 Numerical Approach

We finally consider a numerical solution to the time dependent water wave problem of finite depth. Here we attempt to extend our process for the simply connected case to cover the doubly connected problem. The algorithms referenced here are available in Appendix C. Beginning with algorithm C.1, we initialize our variables and create the numerical representation of our Stokes wave. Because we are considering the finite depth case, we must included the values σ_k and δ_k in the equations for the wave described by equations 5.10 and 5.11. In one loop of this algorithm we start with constructing the Stoke's wave and initial potential function, Φ, with the following code.

```
sig_at = (1+rhoinit.^(2*(1:nt))).*at;
     del_at = (1-rhoinit.^(2*(1:nt))).*at;
        phi = phi/c2 + x;
        x = phi;
```

```
 y = zeros(1,n);
 for l = 1:nt,
  arg = l*phi;
  x = x + sig_at(l)*sin(arg);
  y = y + del_at(l)*cos(arg);
 end
phi = c2*(phi - x);
```

Next we follow the process as before by mapping the physical wave onto the complex $z-$plane by `z = x + iy` and then mapping it to a doubly connected region in the $\zeta-$plane by the exponential map `zeta = exp(-i*z)`, once for each boundary.

The real and imaginary parts are separated for each boundary and used as inputs for Fornberg's method which finds a conformal map `zeta` between the unit annulus and the doubly connected region by the following lines.

```
xp1 = real([zeta1 zeta1(1)]); yp1 = imag([zeta1 zeta1(1)]);
xp2 = real([zeta2 zeta2(1)]); yp2 = imag([zeta2 zeta2(1)]);
[zeta1,s1,h1,zeta2,s2,h2,rho,erri,errirho,tl1,tl2] =...
    fornannsp2(n,8,xp1,yp1,xp2,yp2,sinit1,sinit2,rhoinit);
```

We can then find the physical wave by inverting our maps using `z = i*log(zeta)`, once for each boundary.

Finally, the potential function Φ is evaluated at the conformal points using a cubic spline by `phi = periodic_splinetx(s_spl1,[phi phi(1)],s1');` concluding one loop. This process is then repeated four times because as before it is important for the points to be in conformal locations prior to being passed to the Euler's method algorithm.

Once the points are in conformal locations, we pass the data to `eulerwaveann2.m`, given in algorithm C.2, to solve the three main fluids equations and update the boundary

correspondences and potential function. The first part of the algorithm initializes variables and then we begin the first pass through our loop. We proceed through the conformal mapping process once again. Recalling that our solution to the Dirichlet problem for the annulus was determined by Fourier series, `fft(phi)` applies the fast Fourier transform to the conformal Phi solving the problem for our given data. The next four lines find the partial derivatives for Φ with respect to r and θ but again we must make an appropriate change for the annulus case.

```
phi_a = -real(ifft(1i*w.*Phi));
rhom1 = (1-rho.^(0:n/2))./(1+rho.^(0:n/2));
rhomult = [rhom1 rhom1(n/2:-1:2)];
phi_b = real(ifft(rhomult.*w.*Phi));
```

This is done in the same manner as the following calculation. Suppose $f(z)$ has a power series expansion. Then

$$f(z) = \sum_{k=-\frac{n}{2}+1}^{\frac{n}{2}} c_k z^k \qquad \text{and} \qquad f'(z) = \sum_{k=-\frac{n}{2}+1}^{\frac{n}{2}} k c_k z^{k-1} \tag{5.27}$$

Setting $z = e^{-iv}$ and multiplying by e^{-iv} yields

$$f'(e^{-iv})e^{-iv} = \sum_{k=-\frac{n}{2}+1}^{\frac{n}{2}} k c_k e^{-i(k-1)v} e^{-iv} \tag{5.28}$$

for $k = -1$ then we have $e^{-i(-2)v}e^{-iv} = e^{2iv}e^{-iv} = e^{iv} = e^{-i(-1)v}$ which gives us

$$= \sum_{k=-\frac{n}{2}+1}^{\frac{n}{2}} k c_k e^{-ikv} \tag{5.29}$$

$$= \sum_{k=1}^{\frac{n}{2}} k c_k e^{-ikv} + \sum_{k=-\frac{n}{2}+1}^{-1} k c_k e^{-ikv} \tag{5.30}$$

$$= \sum_{k=1}^{\frac{n}{2}} k c_k e^{-ikv} - \sum_{j=1}^{\frac{n}{2}-1} j c_{-j} e^{ikv} = \text{fft}[f(e^{-iv})] \tag{5.31}$$

which are the Laurent coefficients. The few lines

```
coef1 = ((0:n/2).*c1(1:n/2+1)); coef2 = ((-n/2+1:-1).*c1(n/2+2:n));
c = [coef1 coef2];
dzdv = -1i*n*ifft(1i*c)./zeta1; dz(end+1,:) = dzdv;
```

calculate the total derivative, $dz/d\theta$, of the conformal map as is needed from equation 5.1. Finally, we again use a basic Euler method scheme to solve the differential equations describing the kinematic and dynamic free surface boundary conditions. This gives us updates for boundary correspondence points and the potential function for each time step. Note that we only update the upper(outer) boundary `z1` and the potential function.

```
phi = (-g.*imag(z1)+0.5*(phi_a.^2 + phi_b.^2)./abs(dzdv).^2)*dt + phi;
z1 = exp(1i*angle(dzdv)).*(phi_a + 1i*phi_b)./abs(dzdv)*dt + z1;
```

This completes one time step for the wave propagation algorithm. In the case of our experiment represented here, one period will require 1000 time steps.

We now examine some outputs from our code for Stokes waves on water of finite depth. Unlike the simply connected case, our algorithm does not show movement of the fluid inside the doubly connected region. However we do see updates for the surface particles of the wave as shown in figure 5.18 through figure 5.21.

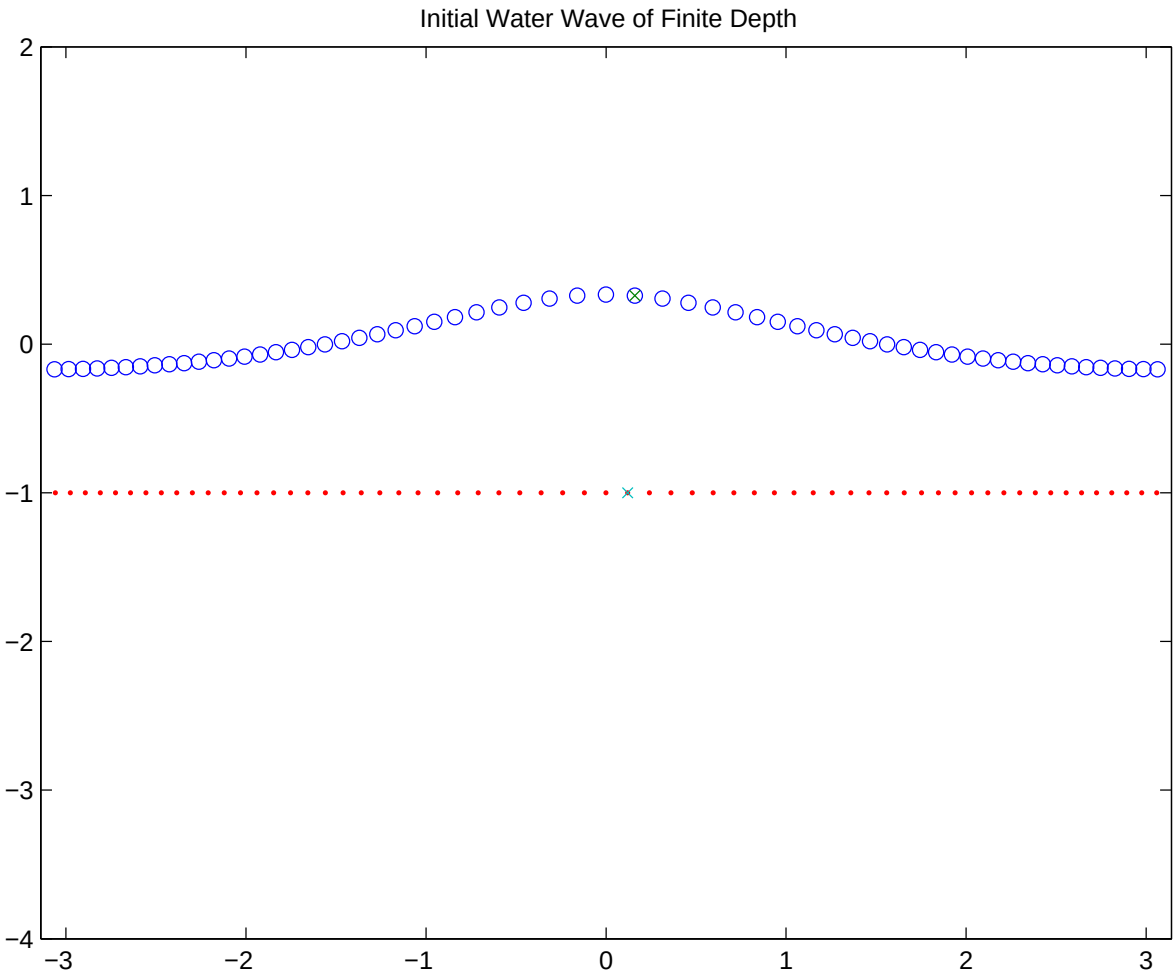

Figure 5.18: Water Wave with Finite Depth for m = 1

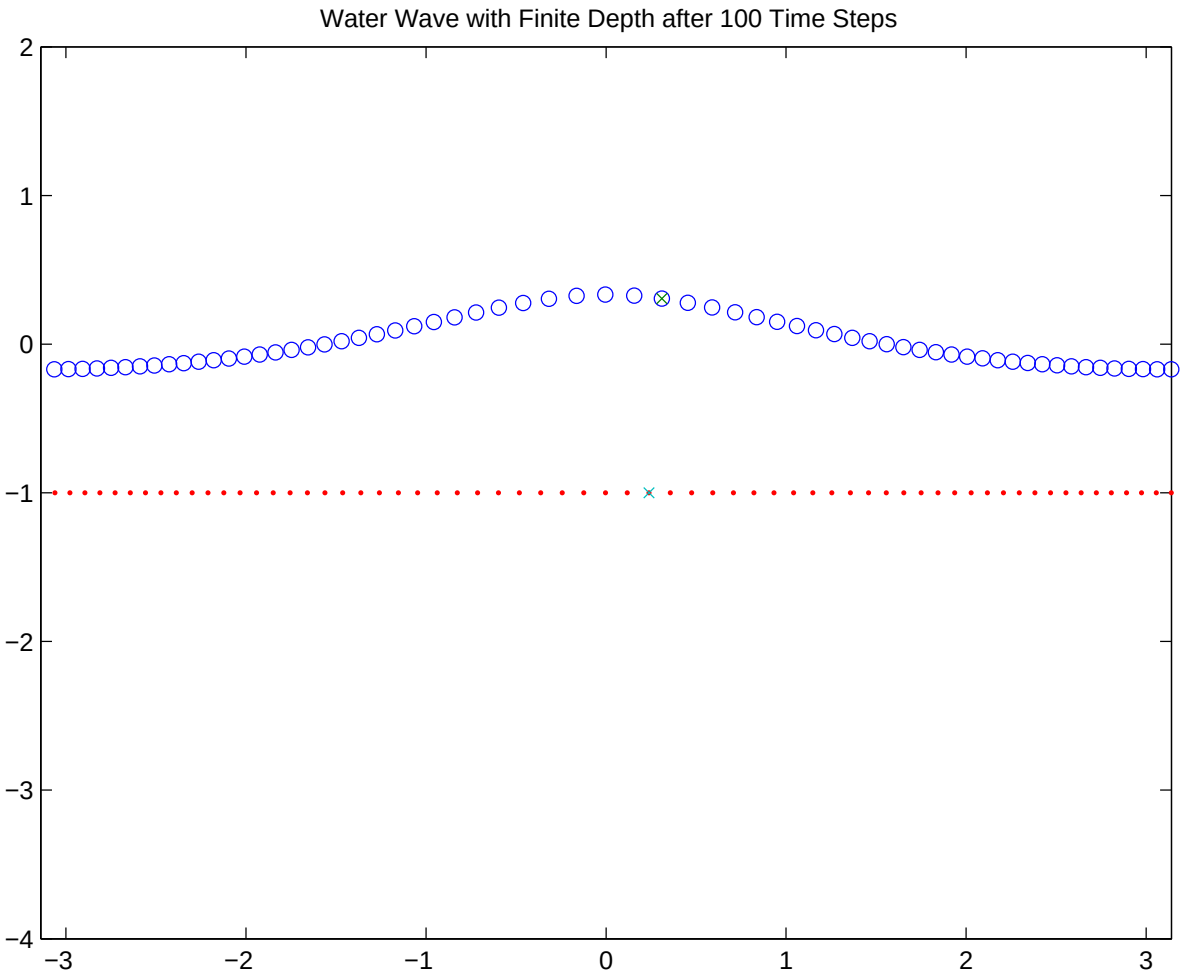

Figure 5.19: Water Wave with Finite Depth for m = 100

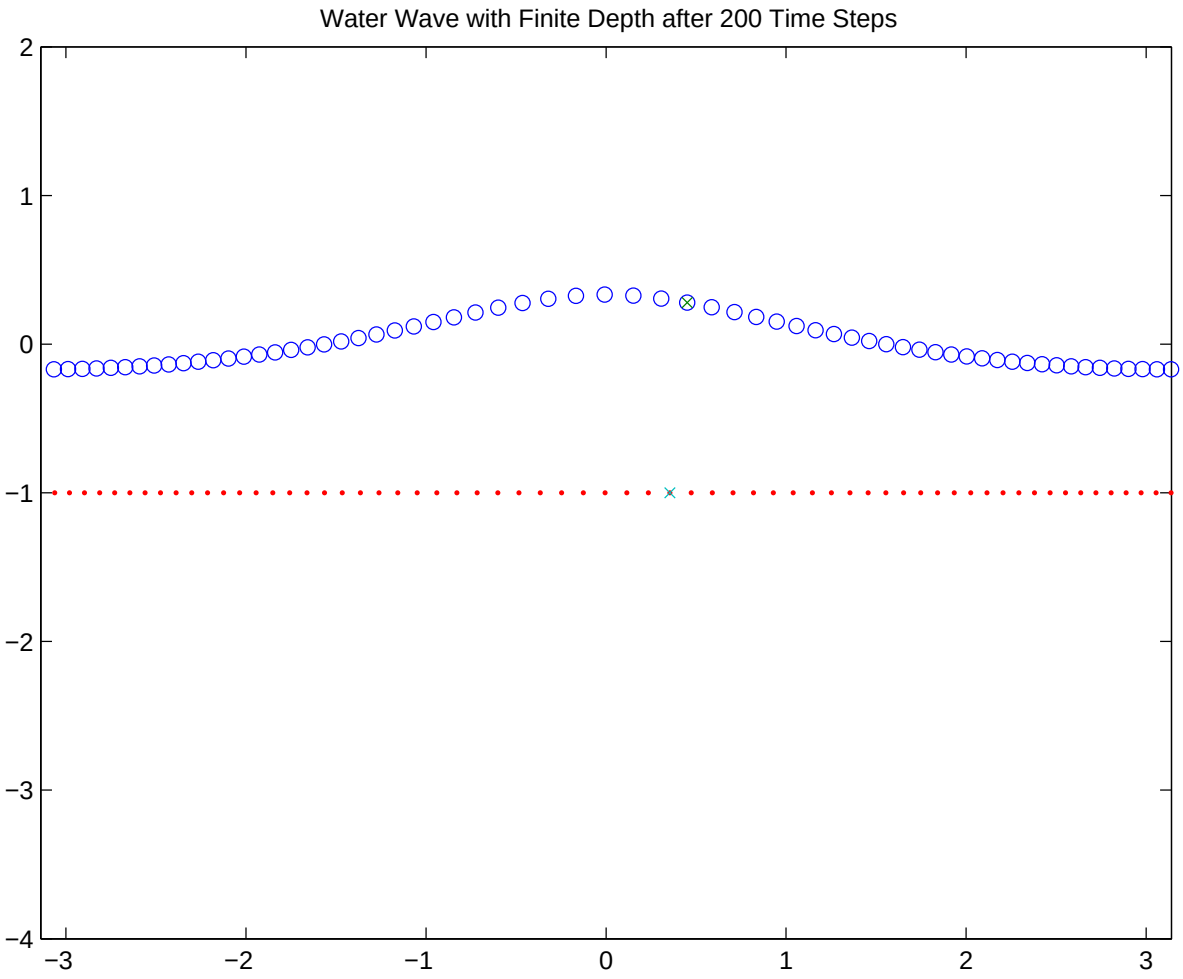

Figure 5.20: Water Wave with Finite Depth for m = 200

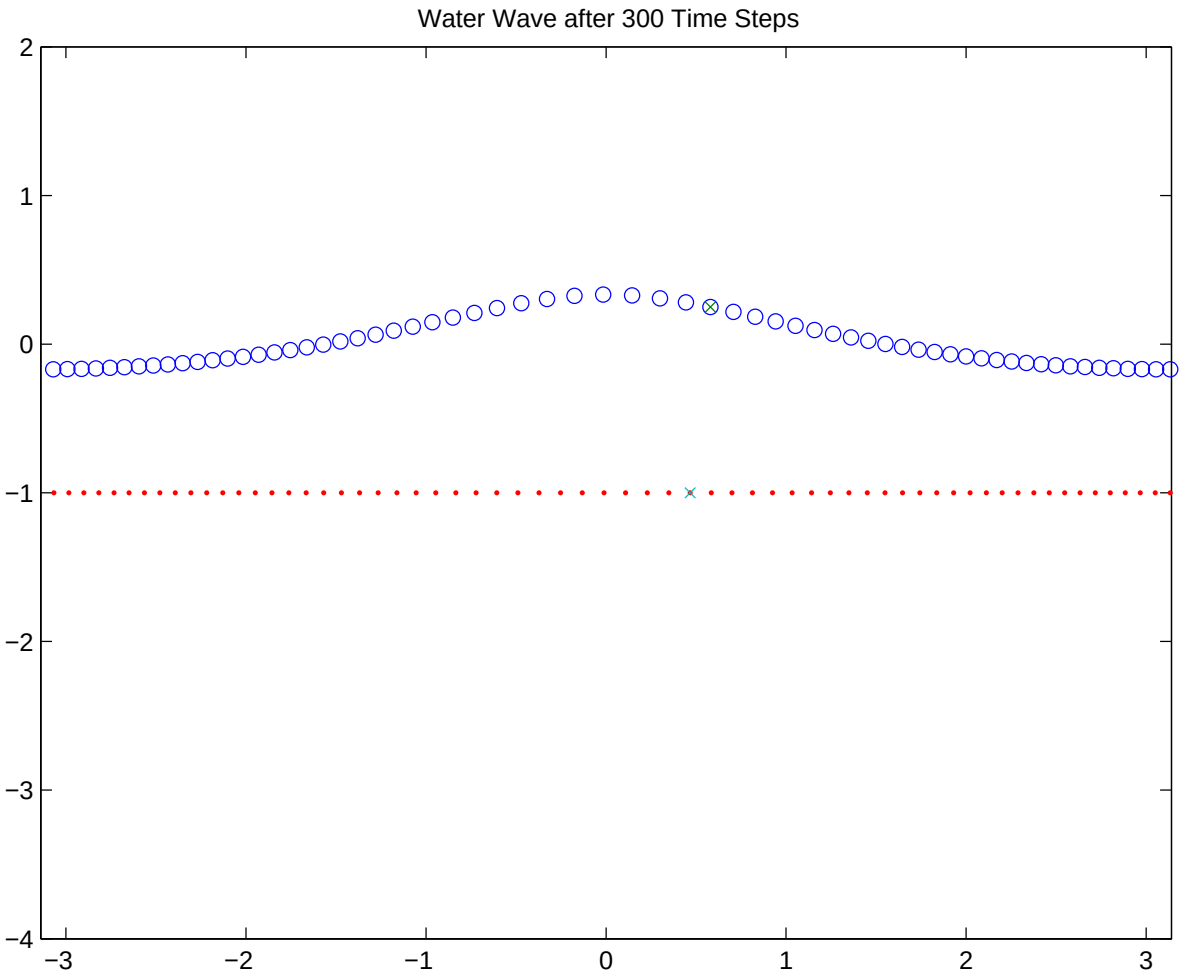

Figure 5.21: Water Wave with Finite Depth for m = 300

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusions

In the process of this study we have made two primary accomplishments. We were able to modify code from Fornberg in [Forwave] and Birger in [Birge] solving the Stoke's wave propagation problem for water with an infinite depth, that is the simply connected case, to use DeLillo's version of Fornberg's method from [DW06] and our own version of a cubic spline algorithm, modified from Moeler in [Mol04]. Our approach to this problem resulted in acceptable results showing the movement of the wave clearly along with appropriate surface particle behavior. We were also able to extend our approach for the simply connected case to the problem of Stokes wave propagation on water with a finite depth, that is the doubly connected case, using DeLillo's extension of Fornberg's method for the annulus from [DW06]. However our approach to this case did not show the wave motion but did indicate proper surface particle behavior. This failing, along with instabilities caused choices of small r_0 or long time intervals, leads us to possible future studies on the matter to hopefully better understand the problems and processes involved.

6.2 Future Work

As is stated above, future study of this problem is necessary to fully understand the processes involved, particularly in the doubly connected case. One such area would be to better understand the Initial Value Problems involved in solving for the Stokes wave propagation, specifically the optimal choice for initial velocity potential. Since we derived our equations describing the Stokes wave similarly to Schwartz in [Sch74], a more thorough comparison and understanding of Schwartz series representations might garner a better understanding of how to vary wave height and speed more directly in order to better modify our study of the waves involved. It would also be beneficial to better understand the insta-

bilities caused by small r_0 or long time intervals. One might also consider the possibility of using higher order ordinary differential equation solvers instead of Euler's method. A better choice of ordinary differential equation solver might give us a better accuracy for each time step. Finally, we would also like a better understanding of how to determine accuracy and how it changes for different choices of evaluation points, that is how varying n changes our accuracy.

REFERENCES

REFERENCES

[Ahl] L. V. Ahlfors, *Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable*, McGraw-Hill, New York, 1979. pp. 229–230

[Birge] J. R. Birge, *Independent Study Report: Numerical Conformal Mapping Code for Solution of Deep Water Waves*, University of Colorado, Boulder, Applied Mathematics Depatment, 2000.

[Craik04] A. D. D. Craik, *George Gabriel Stokes on water wave theory*, Annu. Rev. Fluid Mech., 36 (2004), pp. 23–42.

[Craik05] A. D. D. Craik, *George Gabriel Stokes on water wave theory*, Annu. Rev. Fluid Mech., 37 (2005), pp. 1–28.

[Deb94] L. Debnath, *Nonlinear Water Waves*, Academic Press, New York, 1994.

[De14] T. DeLillo, *Tutorial on Fourier series methods for numerical conformal mapping of smooth domains*, 2014, http://www.math.wichita.edu/ delillo/TD_tutorial.pdf

[De15] T. DeLillo, *Numerical Conformal Mapping using Fourier Series*, book in preparation, 2015.

[DP98] T. DeLillo and J. Pfaltzgraff, *Numerical conformal mapping methods for simply and doubly connected regions*, SIAM J. Sci. Comput., 19 (1998), pp. 155–171.

[DW06] T. DeLillo and L. Wang, *A MATLAB Toolbox (FFTCONF) for computing conformal maps with Fourier series methods*, preprint (2006).

[Forwave] B. Fornberg, *Time dependent water wave code based on conformal mappings*, Caltech report, c.1980.

[For80] B. Fornberg, *A numerical method for conformal mappings*, SIAM J. Sci. Stat. Comput, 1 (1980), pp. 386–400.

[For84] B. Fornberg, *A numerical method for conformal mapping of doubly connected regions*, SIAM J. Sci. Statis. Comput., 4 (1984), pp. 771–783.

[Hen82] P. Henrici, *Essentials of Numerical Analysis*, Wiley, New York, 1982. pp. 260–267

[Hen86] P. Henrici, *Applied and Computational Complex Analysis*, vol. III, Wiley, New York, 1986.

[Mc03] R. C. McOwen, *Partial Differential Equations: Methods and Applications*, Pearson Education, Upper Saddle River, 2003. p. 105

REFERENCES (continued)

[Mol04] C. B. Moler, *Numerical Computing with MATLAB*, SIAM, Philadelphia, 2004. pp. 102–115, http://www.mathworks.com/moler/index_ncm.html

[Sch74] L. W. Schwartz, *Computer extension and analytic continuation of Stokes' expansion for gravity waves* J. Fluid Mech., 62, (1974), pp. 553–578.

[Stok47] G. G. Stokes, *On the theory of oscillatory waves*, Trans. Camb. Philos. Soc., 8 (1847), pp. 441–455.

[Weg2] R. Wegmann, *On Fornberg's numerical method for conformal mapping*, SIAM J. Numer. Anal., 23 (1986), pp. 1199–1213.

APPENDICES

APPENDIX A

PERIODIC CUBIC SPLINE ALGORITHMS

Algorithm A.1 periodic_splinetx.m

```
function v = periodic_splinetx(x,y,u)
%SPLINETX  Textbook periodic spline function.
%  v = splinetx(x,y,u) finds the piecewise cubic interpolatory
%  spline S(x), with S(x(j)) = y(j), and returns v(k) = S(u(k)).
%  Copyright 2014 Cleve Moler %  Copyright 2014 The MathWorks, Inc.
%  First derivatives
   h = diff(x); delta = diff(y)./h;
   d = periodic_splineslopes(h,delta); d = [d d(1)];
%  Piecewise polynomial coefficients
   n = length(x);
   c = (3*delta - 2*d(1:n-1) - d(2:n))./h;
   b = (d(1:n-1) - 2*delta + d(2:n))./h.^2;
% shift u to interval [x(1) x(end)]
  u = u - floor(u/(x(end)-x(1)))*(x(end)-x(1));
%  Find subinterval indices k so that x(k) <= u < x(k+1)
   k = ones(size(u));
   for j = 2:n-1
      k(x(j) <= u) = j;
   end
%  Evaluate spline
   s = u - x(k);
   v = y(k) + s.*(d(k) + s.*(c(k) + s.*b(k)));
% ------------------------------------------------------------
function d = periodic_splineslopes(h,delta)
%  SPLINESLOPES  Slopes for cubic spline interpolation.
%  splineslopes(h,delta) computes d(k) = S'(x(k)).
%  Uses periodic end conditions.
%  Diagonals of tridiagonal system
   n = length(h); %+1;
   a = zeros(size(h)); b = a; c = a; r = a;
   a(1:n-1) = h(2:n); a(n) = h(1);
   b(1) = 2*(h(n)+h(1)); b(2:n) = 2*(h(2:n)+h(1:n-1));
   c(1) = h(n); c(2:n) = h(1:n-1);
%  Right-hand side
   r(1) = 3*(h(1)*delta(n)+h(n)*delta(1));
   r(2:n) = 3*(h(2:n).*delta(1:n-1)+ ...
               h(1:n-1).*delta(2:n));
%  Solve tridiagonal linear system
   d = periodic_tridisolve(a,b,c,r);
```

APPENDIX A

Algorithm A.2 periodic_tridisolve.m

```
function x = periodic_tridisolve(a,b,c,d)
%   TRIDISOLVE  Solve tridiagonal system of equations.
%     x = TRIDISOLVE(a,b,c,d) solves the system of linear equations
%     b(1)*x(1) + c(1)*x(2) = d(1),
%     a(j-1)*x(j-1) + b(j)*x(j) + c(j)*x(j+1) = d(j), j = 2:n-1,
%     a(n-1)*x(n-1) + b(n)*x(n) = d(n).
%   The algorithm does not use pivoting, so the results might
%   be inaccurate if abs(b) is much smaller than abs(a)+abs(c).
%   More robust, but slower, alternatives with pivoting are:
%     x = T\d where T = diag(a,-1) + diag(b,0) + diag(c,1)
%     x = S\d where S = spdiags([[a; 0] b [0; c]],[-1 0 1],n,n)
%   Copyright 2014 Cleve Moler
%   Copyright 2014 The MathWorks, Inc.
    x = d;
    n = length(x);
    p = a(n);
for j = 1:n-3
   mu = a(j)/b(j);
   b(j+1) = b(j+1) - mu*c(j);
   x(j+1) = x(j+1) - mu*x(j);
   a(j) = -mu*p;
   mu2 = c(n)/b(j);
   c(n) = -mu2*c(j);
   b(n) = b(n) - mu2*p;
   x(n) = x(n) - mu2*x(j);
   p = a(j);
end
%   Guassian Elimination on the n-2 step
    b(n-1) = b(n-1) - (a(n-2)/b(n-2))*c(n-2);
    c(n-1) = c(n-1) - (a(n-2)/b(n-2))*a(n-3);
    x(n-1) = x(n-1) - (a(n-2)/b(n-2))*x(n-2);
%   Bottom corner element on the n-2 step
    a(n-1) = a(n-1) - (c(n)/b(n-2))*c(n-2);
    b(n) = b(n) - (c(n)/b(n-2))*a(n-3);
    x(n) = x(n) - (c(n)/b(n-2))*x(n-2);
%   Guassian Elimination on the n-1 step
    b(n) = b(n) - (a(n-1)/b(n-1))*c(n-1);
    x(n) = x(n) - (a(n-1)/b(n-1))*x(n-1);
%   Initializing the backsolving algorithm
    x(n) = x(n)/b(n);
    x(n-1) = (x(n-1) - c(n-1)*x(n))/b(n-1);
for j = n-2:-1:2
   x(j) = (x(j) - c(j)*x(j+1) - a(j-1)*x(n))/b(j);
end
%   Backsolving for first row
    x(1) = (x(1) - c(1)*x(2) - a(n)*x(n))/b(1);
```

APPENDIX B

SIMPLY CONNECTED STOKES WAVE ALGORITHMS

Algorithm B.1 simply_connected_stokes_final.m

```
c2 = 1.04155;
at = [.25 .131040 .077741 .048696 .03145 .020715 .013832 ...
    .009329 .006341 .004336 .00298 .002057 .001424 .000989 ...
    .000689 .000481 .000336 .000235 .000165 .000116 .000082 ...
    .000057 .00004 .000029 .00002 .000014 .00001 .000007 ...
    .000005 .000004 .000003 .000002 .000001 .000001 .000001];
nt = length(at);

m = 1000; msteps=1000; dt = 2*pi/c2/msteps; n = 64;

tzeta = linspace(0, 1, n+1); tinit = linspace(0, 1, n+1).';
tinit(end) = []; sinit = tzeta(1:n)';

at = at./(1:nt); x = linspace(pi, -pi, n+1);
x = x(1:n); y = zeros(1,n); phi = zeros(1,n);

kstep = 4;
for k = 1:kstep
    phi = phi/c2 + x;
    x = phi;
    y = zeros(1,n);
    for l = 1:nt,
        arg = l*phi;
        x = x + at(l)*sin(arg);
        y = y + at(l)*cos(arg);
    end
    phi = c2*(phi - x);
z = x + 1i*y; %maps wave onto complex plane
zeta = exp(-1i*z); %maps wave into simply connected region
xp = real([zeta zeta(1)]); yp = imag([zeta zeta(1)]);
[zeta,e,s,h,erri,tl] = fornintsp_4_td(n,8,xp,yp,sinit); %conformal map step
zeta = zeta.';
s = s';
z = 1i*log(zeta); %mapping simply connected region back to wave
s_spl=cumsum(h)'; s_spl=[0 s_spl(1:n)];
phi = periodic_splinetx(s_spl,[phi phi(1)],s);
sinit = s/tl;
end
itmax2 = 10;
[xi, yi, phis, cs,dz,si,errii] = eulerwave_final(z, phi, m, dt,itmax2);
```

APPENDIX B

Algorithm B.2 eulerwave_final.m

```
function [xi, yi, phis, cs,dz,si,errii] = eulerwave_final(z, phi, m, dt,itmax)

g = 1; %gravity
%initializing output variables
 xi = []; yi = []; phis = []; cs = []; si = []; errii=[]; dz = [];
%initializing iteration variables
 n = length(z); k = 0; w = [0:n/2, -n/2+1:-1]; w(n/2+1) = 0;
%spline knot locations
 tzeta = linspace(0, 1, n+1);
%initial conformal mapping guess
 sinit = tzeta(1:n)';

while k < m
 k = k + 1;
%mapping wave profile to bounded region
 zeta = exp(-1i*z);
%Calculate conformal map via Fornberg
 xp = real([zeta zeta(1)]);
 yp = imag([zeta zeta(1)]);
 [zeta,e,s,h,erri,tl] = fornintsp_4_td(n,itmax,xp,yp,sinit);
 zeta = zeta.';
 c = fft(zeta)/n;
%Conformal Map back to physical space.
 z = 1i*log(zeta);
%record results
 xi(end+1, :) = real(z); yi(end+1, :) = imag(z);
 phis(end+1, :) = phi; cs(end+1, :) = c;
 si(end+1, :) = s'; errii(end+1,:)=erri';
%Periodic Cubic Spline of Potential function
 s_spl=cumsum(h)'; s_spl=[0 s_spl(1:n)];
 phi = periodic_splinetx(s_spl,[phi phi(1)],s');
 sinit = s/tl;
%Calculate derivatives to Laplace Solution.
 Phi = fft(phi);
 phi_a = -real(ifft(1i*w.*Phi));
 phi_b = real(ifft(abs(w).*Phi));
%calculate complex derivative of total conformal mapping.
 dzdv = -1i*n*ifft(1i*(0:n-1).*[c(1:n/2+1)  zeros(1, n/2-1)])./zeta;
 dz(end+1,:) = dzdv;
%First order step of point marker locations and potential
%function values.
 phi = (-g.*imag(z)+0.5*(phi_a.^2 + phi_b.^2)./abs(dzdv).^2)*dt + phi;
 z = exp(1i*angle(dzdv)).*(phi_a + 1i*phi_b)./abs(dzdv)*dt + z;
end
```

APPENDIX C

DOUBLY CONNECTED STOKES WAVE ALGORITHMS

Algorithm C.1 dbl_conn_stokes_schwartzcoeff.m

```
c2 = 1.04155; %wave speed
at = [.25 .131040 .077741 .048696 .03145 .020715 .013832 .009329 .006341 ...
    .004336 .00298 .002057 .001424 .000989 .000689 .000481 .000336 .000235 ...
    .000165 .000116 .000082 .000057 .00004 .000029 .00002 .000014 .00001 ...
    .000007 .000005 .000004 .000003 .000002 .000001 .000001 .000001];
nt = length(at);
d = 1; m = 300; msteps=1000; dt = 2*pi/c2/msteps; n = 64;

tzeta = linspace(0, 1, n+1); tinit = linspace(0, 1, n+1).'; tinit(end) = [];
sinit1 = tzeta(1:n)'; sinit2 = sinit1; rhoinit = .5; at = at./(1:nt);

x = linspace(pi, -pi, n+1); x = x(1:n); y = zeros(1,n); phi = zeros(1,n);

kstep = 4;
for k = 1:kstep
    sig_at = (1+rhoinit.^(2*(1:nt))).*at;
    del_at = (1-rhoinit.^(2*(1:nt))).*at;
    phi = phi/c2 + x;
    x = phi;
    y = zeros(1,n);
    for l = 1:nt,
        arg = l*phi;
        x = x + sig_at(l)*sin(arg);
        y = y + del_at(l)*cos(arg);
    end
    phi = c2*(phi - x);

z1 = x + 1i*y; z2 = x - 1i*d; zeta1 = exp(-1i*z1); zeta2 = exp(-1i*z2);

xp1 = real([zeta1 zeta1(1)]); yp1 = imag([zeta1 zeta1(1)]);
xp2 = real([zeta2 zeta2(1)]); yp2 = imag([zeta2 zeta2(1)]);
[zeta1,s1,h1,zeta2,s2,h2,rho,erri,errirho,tl1,tl2] =...
    fornannsp2(n,8,xp1,yp1,xp2,yp2,sinit1,sinit2,rhoinit);

zeta1 = zeta1.'; zeta2 = zeta2.'; z1 = 1i*log(zeta1); z2 = 1i*log(zeta2);

s_spl1=cumsum(h1)'; s_spl1=[0 s_spl1(1:n)];
phi = periodic_splinetx(s_spl1,[phi phi(1)],s1');
sinit1 = s1/tl1; sinit2 = s2/tl2; rhoinit = rho;
end
itmax2 = 10;
[xi, yi, phis, cs,dz,si,errii] = eulerwaveann2(z1,z2,phi,m,dt,itmax2);
```

Algorithm C.2 eulerwaveann2.m

```
function [xi1,yi1,phis,cs1,dz,si1,errii] = eulerwaveann2(z1,z2,phi,m,dt,itmax)
% z1 is wave (top boundary) % z2 is floor (bottom boundary)
g = 1; %gravity
%initializing output variables
xi1 = []; yi1 = []; xi2 = []; yi2 = []; phis = []; cs1 = []; dz = [];
si1 = []; errii=[]; dz = [];
%initializing iteration variables
n = length(z1); k = 0; w = [0:n/2, -n/2+1:-1]; w(n/2+1) = 0;
%spline knot locations
tzeta = linspace(0, 1, n+1);
%initial conformal mapping guess
tinit = linspace(0, 1, n+1).'; tinit(end) = []; sinit1 = tzeta(1:n)';
sinit2 = sinit1; rhoinit = .5;

while k < m
    k = k + 1;
    %mapping wave profile to bounded region
    zeta1 = exp(-1i*z1); zeta2 = exp(-1i*z2);
    %Calculate conformal map via Fornberg
    xp1 = real([zeta1 zeta1(1)]); yp1 = imag([zeta1 zeta1(1)]);
    xp2 = real([zeta2 zeta2(1)]); yp2 = imag([zeta2 zeta2(1)]);
    [zeta1,s1,h1,zeta2,s2,h2,rho,erri,errirho,tl1,tl2] =...
        fornannsp2(n,itmax,xp1,yp1,xp2,yp2,sinit1,sinit2,rhoinit);
    erri; errirho; zeta1 = zeta1.'; zeta2 = zeta2.'; c1 = fft(zeta1)/n;
    t1 = s1'/tl1;
    z1 = 1i*log(zeta1); z2 = 1i*log(zeta2); %recovering physical wave
    %record results
    xi1(end+1, :) = real(z1); yi1(end+1, :) = imag(z1);
    xi2(end+1, :) = real(z2); yi2(end+1, :) = imag(z2);
    phis(end+1, :) = phi; cs1(end+1, :) = c1; si1(end+1, :) = s1';
    errii(end+1,:)=erri';
    %Calculate derivitaves to Laplace Solution.
    s_spl1=cumsum(h1)'; s_spl1=[0 s_spl1(1:n)];
    phi = periodic_splinetx(s_spl1,[phi phi(1)],s1');
    sinit1 = s1/tl1; sinit2 = s2/tl2; rhoinit=rho;
    Phi = fft(phi);
    phi_a = -real(ifft(1i*w.*Phi));  % (same as disk case)
    rhom1 = (1-rho.^(0:n/2))./(1+rho.^(0:n/2));
    rhomult = [rhom1 rhom1(n/2:-1:2)];
    phi_b = real(ifft(rhomult.*w.*Phi));
    %calculate complex derivative of total conformal mapping.
    coef1 = ((0:n/2).*c1(1:n/2+1)); coef2 = ((-n/2+1:-1).*c1(n/2+2:n));
    c = [coef1 coef2];
    dzdv = -1i*n*ifft(1i*c)./zeta1; dz(end+1,:); dz(end+1,:) = dzdv;
    %First order step of point marker locations and potential function values.
    phi = (-g.*imag(z1)+0.5*(phi_a.^2 + phi_b.^2)./abs(dzdv).^2)*dt + phi;
    z1 = exp(1i*angle(dzdv)).*(phi_a + 1i*phi_b)./abs(dzdv)*dt + z1;
end
```