Numerical Tests of the Fokas Method for Helmholtz-type Partial Differential Equations: Dirichlet to Neumann Maps

Christopher-Ian Raphael Davis
University of Colorado at Boulder

Follow this and additional works at: https://scholar.colorado.edu/appm_gradetds
Part of the Applied Mathematics Commons

Recommended Citation
https://scholar.colorado.edu/appm_gradetds/17
Numerical Tests of the Fokas Method for Helmholtz-type Partial Differential Equations: Dirichlet to Neumann Maps

by

Christopher-Ian Raphaël Davis

B.S., University of Colorado at Boulder, 2008

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Masters of Science
Department of Applied Mathematics
2011
This thesis entitled:
Numerical Tests of the Fokas Method for Helmholtz-type Partial Differential Equations: Dirichlet to Neumann Maps
written by Christopher-Ian Raphaël Davis
has been approved for the Department of Applied Mathematics

Bengt Fornberg

James Curry

Natasha Flyer

Date ______________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.
A method for solving boundary value problems for linear partial differential equations in convex polygons developed by A.S. Fokas in the late 1990s is introduced. In order to solve well-posed boundary value problems using the novel Fokas approach, certain global relations must be derived. These global relations yield so-called Dirichlet to Neumann maps which not only allow us to solve Helmholtz-type PDEs using the Fokas method, but they are also of interest in their own right. Given a convex polygon and a prescribed set of boundary conditions associated with a PDE, the Dirichlet to Neumann map enables us to numerically recover unknown boundary conditions with relatively high accuracy without solving the PDE on the interior. The numerical implementation of the Dirichlet to Neumann map is shown to be an efficient and accurate method for resolving unknown boundary conditions. The map is also analyzed and certain parameters are optimized. With an accurate Dirichlet to Neumann map, solving the modified Helmholtz and the Helmholtz equations via the Fokas method becomes possible.
Acknowledgements

I would like to thank with the utmost gratitude my mentor Bengt Fornberg. It is hard for me to describe just how much I learned from working with him. Following his bright insight through the confusion that research can sometimes be is most appreciated! I would also like to thank Natasha Flyer for useful discussions regarding the Fokas methodology. I want to thank Professor Curry and Professor Segur for their interest in my work and for their valuable insights and comments. For the people closest to me, my parents, my siblings, and my girl friend, I want to thank you all for your prolonged support. Yes I do! Thank You!
Contents

Chapter

1 Introduction 1

2 Novel Boundary Integral Method: Helmholtz and Modified Helmholtz Equations 3
   2.1 Outline of Derivation of the Novel Boundary Integral Method 3
   2.2 Global Relations 4

3 Helmholtz and Modified Helmholtz Equations 7
   3.1 Governing Equations and Geometry 7
   3.2 The Fokas Boundary Integral Formulation 8
      3.2.1 Solution for the Modified Helmholtz Equation 8
      3.2.2 Solution for the Helmholtz Equation 9
   3.3 The Global Relations: Dirichlet to Neumann Maps 11

4 Preliminary Tests with Analytic Solution: Dirichlet To Neumann Map 13
   4.1 The Helmholtz Equation on the Unit Square and Rectangle 15
   4.2 The Helmholtz Equation on a Trapezoid 22
   4.3 The Modified Helmholtz on a Trapezoid 24

5 Distributions and Performance 26
   5.1 Genetic Algorithm on the Unit Square and the Trapezoid 26
   5.2 Distribution Tests 38
The Helmholtz Equation: an Eigenvalue Problem

6.1 On the Unit Square
   6.1.1 Eigenmodes and Eigenvalues
   6.1.2 Singular Values of the Error
   6.1.3 Singular Values of the Linear System

6.2 Trapezoid
   6.2.1 Eigenmodes and Eigenvalues
   6.2.2 Singular Values of the Linear System

6.3 Testing Eigenmodes in the Dirichlet to Neumann Map

Corner Correction for the Helmholtz Equation

7.1 Singular Expansion in the Case of a Singular Corner

7.2 Singular Terms in the Global Relations

7.3 Implementation

Numerical Solution to the Modified Helmholtz and the Helmholtz Equations

8.1 Solution to the Modified Helmholtz Equation

8.2 Solution to the Helmholtz Equation

Conclusions

9.1 Future Work

Bibliography

Appendix

A The Fokas Method via Green Functions

B Eigen-problem of the Helmholtz equation on the Trapezoid
C  $H$ Function Implementation  94

D  Complex Laplacian  104

E  Minimal Polynomials  106

F  Sample Code: Helmholtz Dirichlet to Neumann map for the Unit Square  111
Tables

Table

4.1 Parametrization in \( t \) for the unit square .................................................. 18
4.2 Parametrization in \( t \) for the trapezoid ............................................................ 23

6.1 The first 36 eigenvalues of the Helmholtz equation on the square domain ordered by
wavenumbers. ........................................................................................................ 50
6.2 The first 31 eigenvalues of the Helmholtz Equation on the trapezoidal domain ordered
by magnitude. The placement in the table does not reflect wavenumbers but rather
magnitude. ........................................................................................................ 60
6.3 The max norm over many \( k \)-values of the result \( b \) for the first 100 wavenumbers.
Note that all numerical values are multiplied by \( 10^{-13} \). We see that the linear
system returns a good approximation to the zero vector. ........................................... 66
Figures

Figure

3.1 Geometrical definitions for part of a polygon with corners \( z_j \) and \( z_{j+1} \) joined by side 
\( S_j \) making an angle \( \alpha_j \) with the horizontal. ........................................... 8

4.1 Schematic representation of the linear system produced with a 4-sided polygon. Notice that the matrix is taller than it is wide as the system is overdetermined. ....... 15

4.2 Square domain with prescribed boundary conditions ........................................... 16

4.3 Rectangular domain with prescribed boundary conditions ..................................... 16

4.4 The solution (4.4) on the square domain with \( \beta = 1 \). ........................................... 17

4.5 The solution (4.5) on the rectangular domain with \( \beta = 1 \). ................................... 17

4.6 The solution (4.4) on the square domain with \( \beta = 5 \). We note the more oscillatory behavior on the domain. .......................................................... 18

4.7 The solution (4.5) on the rectangular domain with \( \beta = 5 \). We note of course the more oscillatory behavior on the domain. ...................................................... 18

4.8 Legendre degree is 15. 60 \( k \)-values. Top row: the four recovered functions on the square superimposed with their explicit counterparts. Bottom row: the errors between the two. The maximum error along all four sides is on the order of \( 10^{-9} \). ... 19

4.9 Maximum of the errors along all four sides of the square plotted against the degree of the Legendre expansion along each side. Note on this log-linear plot that we have spectral convergence! ................................................................. 20
4.10 Legendre degree is 15. 60 $k$-values. Top row: the four recovered functions on the rectangle superimposed with their explicit counterparts. Bottom row: the errors between the two. The maximum error along all four sides is on the order of $10^{-8}$.  

4.11 The trapezoidal domain considered in this section along with the boundary conditions used here.  

4.12 The solution to the Helmholtz equation with the prescribed boundary conditions on the trapezoid. Note that the solution is the same as for the unit square in equation (4.4) of §4.1.  

4.13 100 $k$-values, Legendre degree $M$ is 20. Top row: the four recovered functions on the trapezoid compared to their explicit counterparts. Bottom row: the errors in that comparison. The maximum error along all four sides is on the order of $10^{-6}$. The error is poor near the corner joining sides 3 and 4 because there is a singularity there. See Chapter 7 for more information on the nature of the singularity and a framework for its possible treatment.  

4.14 100 $k$-values, Legendre degree is 20. Top row: the four recovered functions on the trapezoid compared to their explicit counterparts. Bottom row: the errors in that comparison. The maximum error along all four sides is on the order of $10^{-9}$.  

4.15 Maximum of the errors along all four sides of the trapezoid plotted against the degree of the Legendre expansion along each side. Note that this is a log-linear plot.  

5.1 Set of clouds by generations that minimize the error. All plots are on the same scale. $M = 7$. Note that the distributions stagnate after the 50th generation. In conjunction with the next Figure 5.2 we see that the error is consistently decreasing for as long as the distributions change.  

5.2 Error vs Generations. $M = 7$. Note that the error decreases until the new generations do not differ from the older ones.
5.3 Condition number vs Generations. $M = 7$. The condition number drops by one order and then stalls after the 50th generation. .................................................. 29

5.4 Set of clouds by generations that minimize error. $M = 11$. We note that the distributions stop varying after about the 60th generation. .................................................. 30

5.5 Error vs Generations. $M = 11$. As with the $M = 7$ test, the error falls and then stagnates when the generations stop changing. .................................................. 30

5.6 Condition number vs Generations. $M = 11$. The condition number follows the same trend as the error again. .................................................. 31

5.7 Set of clouds by generations that minimize the condition number of the linear system $A$ being solved. $M = 7$. We note that the distributions stop varying after about the 60th generation just as the case of minimizing the error. .................................................. 32

5.8 Error vs generations. $M = 7$. The error improves but does not consistently decrease as we are only optimizing the condition number. However, the error is always acceptable for the chosen parameters. .................................................. 32

5.9 Condition number vs generations. $M = 7$. The condition number decreases nicely as the generations pass especially in the first few generations. It seems that the spreading of the points in the plane be more important than the reorganizing of the points within a certain distance from the origin. .................................................. 33

5.10 Set of clouds by generations that minimize the condition number of the linear system $A$ being solved. $M = 11$. We note that the distributions stop varying after about the 60th generation just as the case of minimizing the error. .................................................. 34

5.11 Error vs generations. $M = 11$. In all, the error drops down by two orders of magnitude. 34

5.12 Condition number vs generations. $M = 11$. The condition number is very well minimized from $10^8$ to $10^2$. .................................................. 35
5.13 On the trapezoid of Figure 4.11. Set of clouds by generations that minimize the condition number of the linear system $A$ being solved. $M = 16$. We note that the distributions stop varying after about the 60th generation just as the case of minimizing the error and the condition number for the square case. The features are very much the same as with the square tests: odes spread out from the original unit box and then reorganize themselves.

5.14 Error vs generations. $M = 16$. Even though we are minimizing the condition number in this test, the error decreases from $10^{-3}$ to $10^{-8}$. The error is evidently better with $M = 16$ but the improvement is rather significant.

5.15 Condition number vs generations. $M = 16$. The condition number is reduced by ten orders of magnitude. Quite remarkable.

5.16 Error (Top row) and condition number (middle row) vs reach of distributions (one distribution per column from left to right starting with the 2-D Gaussian, the Random, the Uniform, and the Halton distribution). The bottom row shows the distribution at its optimal reach $R$. $M = 7$, with 50 $k$-values. Note that all plots on a given row is on the same scale. We see that the reach $R$ has a drastic effect on the performance of the method for all distributions.

5.17 Error and condition number vs reach of distributions. $M = 11$, with 50 $k$-values. Note that all plots on a given row are on the same scale. The method does better overall with more terms in the Legendre expansion. Note the relationship between the error and the condition number is clearer now.
5.18 Error and condition number vs reach of distributions. $M = 16$, with 50 $k$-values.
Note that all plots on a given row are on the same scale. We note that the method performs even better now but the optimal regime seems to have narrowed. This is most likely due to the fact that our system is becoming less and less overdetermined with 100 equations for 64 unknowns. However, the error is still considerably reduced. Note that the condition number is rather high in this test. It is also minimized when the error is minimized.

5.19 Error and condition number vs reach of distributions. $M = 7$, with 500 $k$-values.
Note that all plots on a given row are on the same scale. Note that the error decreases and then remains low, unlike with only 50 $k$-values where the error increased after reaching an optimal regime. The two random distributions seem to perform better, but only marginally so.

5.20 Error and condition number vs reach of distributions. $M = 11$, with 500 $k$-values.
Note that all plots on a given row are on the same scale. We note that the feature are the same as with $M = 7$ but we did lose four orders of magnitude in the error.

5.21 Error and condition number vs reach of distributions. $M = 16$, with 500 $k$-values.
Note that all plots on a given row are on the same scale. We note that the error is reduced to $10^{-13}$. Also note that across all distributions and reach, the worst the method yields is $10^{-2}$. It is remarkable that by simply changing the reach of the method we decreased the error by ten orders of magnitude.

6.1 Square domain with boundary conditions that will yield our eigenmodes. Note that the top side is a normal derivative.

6.2 Several eigenfunctions of the Helmholtz equation
6.3 Maximum error for all sides as we increase $\beta$. Tick marks are at all eigenvalues $\beta_{m,n}$. We see that the spikes coincide with a subset of eigenvalues $\beta_{m,n}$ (at the vertical lines). The error being large for $\beta > 3$, we suspect that spikes are present at all $\beta_{m,n}$ and are not visible.

6.4 Maximum error for all sides as we increase both $\beta$ and $R$. We see that there is a narrow range of $R$ values that yield low errors. The exponential nature of the integrands of the global relations can explain why the method implemented as such cannot be accurate for $k$-values near the origin or too far away in the complex plane.

6.5 Maximum error for all sides as we increase $\beta$. The reach is now a function of $\beta$. The vertical lines are placed at all $\beta_{m,n}$. It is hard to discern if the method is singular at each line.

6.6 Maximum error for all sides as we increase $\beta$ close to the third eigenvalue (ranked by magnitude). The reach is now a function of $\beta$. We note that the error does indeed spike at the eigenvalue.

6.7 Condition number of the linear system as we increase $\beta$. We see the same behavior as in Figure 6.4, namely that there is a regime of $R$ values that minimizes the condition number.

6.8 Condition number of the linear system as we increase $\beta$ with $R(\beta)$. The vertical lines are placed at all $\beta_{m,n}$. We see that all eigenvalues cause the condition number to spike and thus cause the numerical method to perform poorly.

6.9 Trapezoidal domain with boundary conditions which we wish to use to find eigenmodes.

6.10 First 12 eigenmodes on the trapezoid with $\beta$ displayed above.
6.11 Condition number of the linear system as we increase $\beta$. We see the same behavior as in Figure 6.4, namely that there is a regime of $R$ values that minimizes the condition number of the linear system. We however are able to see the spikes in the condition number for all eigenvalues on this plot. We note that most spikes would be unnoticeable if the reach of the method was fixed.

6.12 Same plot as in Figure 6.11 but from a different angle to more clearly see the characteristics of the surface. We clearly see here the narrow valley in which the condition number is minimized.

6.13 Condition number of the linear system as we increase $\beta$ with $R(\beta)$. The vertical lines are placed at all $\beta_{m,n}$ that were computed numerically. We see that all eigenvalues cause the condition number to spike and thus cause the numerical method be singular.

7.1 Trapezoidal domain with boundary conditions. We center our coordinate system at the corner of sides 3 and 4.

7.2 Dirichlet to Neumann map results without the singular correction. We note that sides 1 and 2 perform well but that sides 3 and 4 have large errors near the corner due to the singularity there. 500 Halton nodes with $M = 20$.

7.3 Dirichlet to Neumann map results with the singular correction. We note that sides 1 and 2 perform better but that sides 3 and 4 have even larger errors near the corner due to the singularity there. All the parameters are the same as in Figure 7.2.

8.1 The four rays $l_j$, $j = 1, \ldots, 4$ from the origin to $\infty$ along which we numerically integrate to get our solution.

8.2 The solution to the modified Helmholtz equation computed via the BI-F formulation on the trapezoid. We note that the function values grow exponentially.

8.3 Error over the trapezoid for the solution to the modified Helmholtz equation. We note that the error is bounded by $10^{-11}$.
A.1 Part of the contour of integration in the $k$ plane for the Helmholtz equation . . . . . . 90

B.1 log-log plot of performance of the the algorithm code against itself. The linear trend
suggests that the method is of second order. . . . . . . . . . . . . . . . . . . . . . . . 93

C.1 Computational depth of Steed’s method applied to the $H$ function C.1 up to a reach
of 500 in the complex plane. 1000 Halton nodes . . . . . . . . . . . . . . . . . . . . . . 96

C.2 Computational depth of Lentz’s method applied to the $H$ function C.1 up to a reach
of 500 in the complex plane. 1000 Halton nodes . . . . . . . . . . . . . . . . . . . . . . 97

C.3 Computational depth of the Lentz’s method on the upper incomplete Gamma func-
tion (C.4) up to a radius of 500 in the complex plane. Approximately 1000 Halton
nodes since we removed the nodes on or near the branch cut. We note the exceptional
difference in computation depth compared to the previous two figures. . . . . . . . . 98

C.4 Digits retained for the real part of Steed’s method applied to the $H$ function up to
a radius of 700. 2000 Uniform nodes. We can see excellent performance for the left
half plane but very poor performance for the right half plane. (The jump in accuracy
is not exactly on the imaginary axis but rather at about $z = 30$. . . . . . . . . . . . 99

C.5 Digits retained for the imaginary part of Steed’s method applied to the $H$ function
up to a radius of 700. 2000 Uniform nodes. As with the real part in Figure C.4, we
see excellent performance for the left half plane and poor performance for the right
half. We also note the points on the real axis (with no imaginary component) do
dwell well in the right plane. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100

C.6 Digits retained for the real part of Lentz’s method applied to the $H$ function up to
a radius of 700. 2000 Uniform nodes. We can see excellent performance for the left
half plane but very poor performance for the right half plane. . . . . . . . . . . . . . 100
C.7 Digits retained for the imaginary part of Lentz’s method applied to the $H$ function up to a radius of 700. 2000 Uniform nodes. As with the real part in Figure C.6, we see excellent performance for the left half plane and poor performance for the right half. We also note the points on the real axis (with no imaginary component) do well in the right plane. ................................................................. 101

C.8 Digits retained for the real part of Lentz’s method applied to the upper incomplete Gamma function up to a radius of 700. About 2000 Uniform nodes. We see that the method performs best on the right half plane as opposed to the left half plane (Figures C.4 - C.7). The left half plane is not as accurate but we still retain at worst 12 digits. ................................................................. 101

C.9 Digits retained for the imaginary part of Lentz’s method applied to the upper incomplete Gamma function up to a radius of 700. About 2000 Uniform nodes (excluding the points on the branch cut). The results are very similar to those of Figure C.8 in that the accuracy is adequate throughout the complex plane with best performance in the right half plane. ................................................................. 102

E.1 Maximum of the errors along all four sides of the trapezoid plotted against the degree of the Legendre expansion along each side. Note that this is a log-linear plot. .... 106

E.2 Maximum error along each side plotted against the degree $M$ or the Legendre expansion. ................................................................. 107

E.3 Ninth degree polynomial approximation for all four boundary functions for the modified Helmholtz equation. We appreciate the presence of 11 extrema in the errors and the fact that the extrema are near uniformly bounded. ......................... 108

E.4 Maximum error along each side separately for varying degree. We note that side 2 is the hardest to approximate and that the jagged behavior is also present here. All plots are on the same scale. ................................................................. 109
Chapter 1

Introduction

A novel boundary integral method developed by Athanassios S. Fokas over the past two decades is introduced and partially derived. The boundary integral method by Fokas applies to a wide variety of problems including linear and certain non-linear partial differential equations. The method is flexible as it can be applied to various geometries such as the half line, the quarter plane, the real line, closed and open convex polygons, circular polygons and other geometries. The core of this thesis is based on the formulation derived by Fokas and Dassios in a series of papers including [4], [5], and by Fokas in [11], [13], [12] and in his book on the topic [14] related to, among many other things, solving Helmholtz-type PDEs in convex polygons. The solution to Helmholtz-type PDEs is given in terms of integrals of transforms of boundary conditions. The formulation by Fokas requires full knowledge of both Dirichlet and Neumann boundary conditions on all sides the polygon in question. Well posed problems can only arise from the specification a Dirichlet, a Neumann or a Robin boundary condition on each side of a polygon. It is then necessary to recover the Dirichlet and the Neumann boundary conditions independently of each other in order to use the solutions developed by Fokas. Though the full solution to the modified Helmholtz equation is studied here, the core of this thesis is based on so-called Dirichlet to Neumann maps which allow us to numerically recover unknown boundary conditions from known ones. These maps are of interest in their own right as they allow to us solve for unknown boundary conditions without ever solving for the interior of the polygon in question. Several collocation based numerical methods for implementing these Dirichlet to Neumann maps exists [17], [33] for the case of the Laplace
equation. The case of the modified Helmholtz equation and the Laplacian are also numerically implemented with collocation methods in [34]. The method presented in this thesis is based on the work by Fornberg and Flyer in [16]. They avoid the need for any collocation based methods by expanding the unknown boundary conditions in Legendre polynomials and making use of an integral relation (4.2) to greatly facilitate the problem and essentially eliminate all integrals. The method was developed for the Laplacian but was very easily modified for use with the Helmholtz equation and the modified Helmholtz equation as we will see in Chapter 4.

In order to set the work of this thesis in a useful context, the solutions to Helmholtz-type PDEs will be derived to explicitly show their dependence on both Dirichlet and Neumann boundary conditions. This initial introduction and derivation (Chapters 2, 3 and Appendix A) contains no original work and is mostly based on the aforementioned works of Fokas and Dassios and the work of his student E. A. Spence [34]. The remainder of the thesis is based on the implementation, analysis and optimization of the Dirichlet to Neumann (D2N) maps in Chapters 4 to 7. Several geometries are studied. The effects of the eigenvalues and eigenmodes of the Helmholtz equation are explored and a framework for treating singularities in the corners of polygons is developed. We would like to point the reader to a recent work by Fornberg and Flyer [16] on the topic of the Fokas methodology for the Laplace equation. Singular corners are treated there and the Dirichlet to Neumann map is compared to traditional boundary integral formulations. Lastly, some preliminary tests of the full solution to the modified Helmholtz equation are performed. The Dirichlet to Neumann maps for both the Helmholtz equation and the modified Helmholtz equation are shown to be spectrally accurate and thus competitive methods when compared to existing numerical implementations of the Dirichlet to Neumann maps of the Fokas methodology.
Chapter 2

Novel Boundary Integral Method: Helmholtz and Modified Helmholtz Equations

2.1 Outline of Derivation of the Novel Boundary Integral Method

We will overview here the Boundary Integral Method developed by Athanassios S. Fokas (BIF) and the different ways in which it can be derived for the Helmholtz and the Modified Helmholtz equations on polygonal domains. We note that none of the work in this chapter is original. In fact most of the work is by Fokas and Dassios in a series of papers and a book [4], [5], [13], [14] and one of his students E. A. Spence in his PHD thesis [34]. We start here with the following second order elliptic partial differential equation:

$$\Delta q(x,y) - 4\lambda q(x,y) = 0 \quad (2.1)$$

For $\lambda < 0$ we get the Helmholtz equation and for $\lambda > 0$ we get the modified Helmholtz equation. We note that (2.1) can be recast into a complex form by rewriting $q(x,y)$ in terms of $z$ and $\bar{z}$ and apply the chain rule to arrive at 2.2. Detail on this computation can be found in Appendix D.

$$\Delta q(z,\bar{z}) - \lambda q(z,\bar{z}) = 0 \quad (2.2)$$

Though the framework of the Fokas boundary integral method applies to other PDEs such as the Laplacian and the inhomogeneous versions of the Laplacian, the Helmholtz equation and the modified Helmholtz equation [34], we shall only focus on equation (2.2). From this point on, there are multiple paths for the derivation of the solutions to (2.2). One such path involves reformulating the problem in terms of a general dispersive evolution equation of order $n$ admitting the form [13]:

$$q_t + w(-i\partial x)q = 0 \quad (2.3)$$
where \( w(k) \) is a polynomial of degree \( n \), \( \beta t < x < \infty, 0 < t < T \), \( \beta = \tan \theta \), \( 0 \leq \theta < \pi/2 \), and \( T \) is a positive constant. From this form, a Lax pair is formed, a domain specified and a simultaneous spectral analysis is performed [13]. Such an analysis yields a solution to (2.3). This method is very powerful in that it is more general than the one we shall follow. We shall arrive at the same results following the derivation found in [34]. The derivation there is based on Green functions and their representations in the complex plane. The solution to our PDE (2.2) will be given in Chapter 3 while the derivation will be left in part to Appendix A.

As mentioned in Chapter 1, most of this thesis is dedicated to one aspect of the Fokas methodology, the so-called Dirichlet to Neumann maps. The Dirichlet to Neumann (D2N) maps are essential to the representation of the solution in the Fokas methodology and they arise in the Lax pair derivation as a global relation which needs to be met. However, it turns out that the D2N maps can be derived independently. We derive the global relations for our PDE (2.2) now in the following section in the context of PDEs on convex polygons.

## 2.2 Global Relations

We derive here the Dirichlet to Neumann maps, or the global relations, for (2.2) on convex polygons. The derivation for the Dirichlet to Neumann maps are very similar in the case of the Helmholtz equation and the modified Helmholtz equation. Starting with the Helmholtz equation, (2.2) with \( \lambda < 0 \), we assume that a solution \( q(z, \bar{z}) \) exists. We know that the Helmholtz equation admits the one parameter family of solutions:

\[
\exp \left[ -i \beta \left( kz + \frac{\bar{z}}{k} \right) \right] \tag{2.4}
\]

With this in hand, consider the complex 1-form:

\[
W = \exp \left[ -i \beta \left( kz + \frac{\bar{z}}{k} \right) \right] \left[ (q_z + ik\gamma q) \, dz + \left( \frac{\gamma q_z + \beta q}{ik} \right) \, d\bar{z} \right] \tag{2.5}
\]

We will show that (2.4) satisfies the Helmholtz equation if and only if the differential (2.6) of the complex 1-form (2.5) is closed. Recall that the differential \( dW \) of a complex 1-form \( W = g\, dz + h\, d\bar{z} \)
is the complex two-form given by (2.6) [38]:

\[ dW = dg \wedge dz + dh \wedge d\bar{z} = (g_{\bar{z}} - h_z) \, d\bar{z} \wedge dz \]  

(2.6)

We also recall that for a 1-form \( W \) to be closed, the differential \( dW \) must equal to zero, that is, \( dW = 0 \). Upon using (2.6) with (2.5), simplifying and using the fact that the wedge operator \( \wedge \) is skew symmetric, we arrive at the following result:

\[ dW = \exp \left[ -i\beta \left( k z + \frac{\bar{z}}{k} \right) \right] (1 - \gamma) \left( q_{\bar{z}} + \beta^2 q \right) \]  

(2.7)

From (2.7) it is clear that the complex 1-form \( W \) is closed if and only if (2.4) satisfies the Helmholtz equation. At this point, we parametrize the sides of the polygon with the parameter \(-1 \leq s \leq 1\) and note that if \( z \) is on the boundary of the polygon, then,

\[ q_z dz = \frac{1}{2} (\dot{q} + iq_n) ds \]  

(2.8)

where \( \dot{q} \) represents the derivative of \( q \) along the boundary of the polygon and \( q_n \) represents the derivative of \( q \) normal to the boundary of the polygon. Parameterizing the polygon, making appropriate substitutions and setting \( \gamma = -1 \) we arrive at our complex 1-form in terms of \( q, q_z \) and \( q_{\bar{z}} \):

\[ W = \exp \left[ -i\beta \left( k z + \frac{\bar{z}}{k} \right) \right] \left( q_z + ik\beta q \right) dz - \left( q_{\bar{z}} + \frac{i\beta}{k} q \right) d\bar{z} \]  

(2.9)

\( W \) being closed and our polygonal domain being simply connected, we apply Green’s Theorem in two dimensions:

\[ \int_{\partial \Omega} W = \int \int_{\Omega} dW = 0 \]  

(2.10)

We now have an expression relating the Dirichlet and the Neumann boundary conditions on each side of the polygon:

\[ \int_{\partial \Omega} e^{-i\beta (kz + \bar{z})/k} \left( q_n + \beta \left( \frac{-1}{k} \frac{\partial \bar{z}}{\partial s} + k \frac{\partial z}{\partial s} \right) q \right) ds = 0 \]  

(2.11)

where \( k \in \mathbb{C} \) is our free parameter from our one parameter family of solutions 2.4. The fact that \( k \) is a free parameter will enable us to numerically implement the Dirichlet to Neumann maps for
convex polygons. As is turns out, in order to solve for \( q(z, \bar{z}) \) one needs not only (2.11) but also its Schwartz conjugate [14]. The Schwartz conjugate:

\[
\int_{\partial \Omega} e^{i\beta(k\pi + z/k)} \left( q_n + \beta \left( -\frac{1}{k} \frac{\partial z}{\partial s} + k \frac{\partial \bar{z}}{\partial s} \right) q \right) ds = 0 \tag{2.12}
\]

where the derivatives on each side of the polygon are normal and exterior to each side. The global relations (2.11) and (2.12) are the two key equations which allow for numerical Dirichlet to Neumann maps to be performed. Most of this thesis is based on the numerical implementation and analysis of the (D2N) for the Helmholtz equation.

The derivation for the modified Helmholtz equation is very similar, substituting \( \beta \) with \( i\beta \), and can be found in [14]. The global relations which ensue are given in Chapter 3, equations (3.11) and (3.12). Though we have derived the global relations independently of the solution for the interior of polygons, we shall outline the derivation of those solutions to place more importance on the usefulness of the global relations as we shall see in Chapter 3.
Chapter 3

Helmholtz and Modified Helmholtz Equations

3.1 Governing Equations and Geometry

Please note that none of the work of this chapter is original, it is based mostly on Spence [34] and Fokas [14]. We go through basic definitions to setup the general problem of finding the solutions to the modified Helmholtz equation and the Helmholtz equation. We consider here the general linear partial differential equation (3.1).

$$\frac{\partial^2 q}{\partial z^2} + \frac{\partial^2 q}{\partial \bar{z}^2} - \lambda q = 0$$  \hspace{1cm} (3.1)

As we have mentioned earlier, $\lambda < 0$ yields the Helmholtz equation and $\lambda > 0$ the modified Helmholtz equation. The domains considered in this thesis are the unit square, a rectangle, and a trapezoidal domain shown in Figures 4.2, 4.3, and 4.11 respectively. The defining corner points are labeled $z_j$, $j = 1, \ldots, n$ ordered anticlockwise starting from the bottom left at the origin: $z_1 = 0$. For all the cases tested in this thesis, $n = 4$. The sides themselves are labeled $S_j$, $j = 1, \ldots, n$, and are defined as the segments joining $z_j$ to $z_{j+1}$. Note that $z_{n+1} = z_1$. We also define $\alpha_j = \arg (z_{j+1} - z_j)$. The domain itself is labeled as $\Omega$, while the periphery of the domain is labeled as $\partial \Omega$. Some of these definitions are depicted in Figure 3.1.
3.2 The Fokas Boundary Integral Formulation

The Fokas boundary integral method can be applied to, among other geometries, open or closed convex polygons. A brief description of how one can derive the BI-F for convex polygons was described in Chapter 2 while another derivation via Green functions is very well done and explained in detail by E. A. Spence in his PHD Thesis [34] and is briefly outlined in Appendix A. The core of this thesis being the testing of various aspects of the Fokas methodology we do not spend too much time introducing the derivations of the equations in use as that task has been remarkably done in several resources such as [34] and [14].

3.2.1 Solution for the Modified Helmholtz Equation

Assuming that the solution to the modified Helmholtz equation $q(z, \bar{z})$ exists, and that it is sufficiently smooth on $\Omega$, it can be written as

$$q(z, \bar{z}) = \frac{1}{4\pi i} \sum_{j=1}^{n} \int_{l_j} \exp\left(i\beta\left(kz - \frac{\bar{z}}{k}\right)\right) \hat{q}_j(k) \frac{dk}{k} (3.2)$$

where $k \in \mathbb{C}$. The spectral functions $\hat{q}_j(k)$, $j = 1, \ldots, n$ are defined as

$$\hat{q}_j(k) = \int_{z_j}^{z_{j+1}} \exp\left(-i\beta\left(kz - \frac{\bar{z}}{k}\right)\right) \left[q_n + \beta \left(\frac{1}{k} \frac{dz}{ds} + k \frac{d\bar{z}}{ds}\right)q\right] ds (3.3)$$

and the rays $l_j$, $j = 1, \ldots, n$, oriented from 0 to $\infty$ are defined by

$$l_j = \{k \in \mathbb{C} : \arg k = -\alpha_j\} (3.4)$$
We note here that the solution to the modified Helmholtz equation involves the spectral functions (3.3) which in turn involve both $q$ and $q_n$ on $\partial \Omega$. We also note that $q_n$ is the normal derivative of $q$ on the boundary taken in the outward direction, that is, towards the exterior of the polygon. Because of this dependence on both Dirichlet and Neumann boundary conditions, we realize that in order to numerically or explicitly find a solution to the modified Helmholtz equation with the Fokas methodology, one needs to know both Dirichlet and Neumann boundary conditions on the entirety of $\partial \Omega$. We will see in §3.2.2 that the same is true for the Helmholtz equation. As was mentioned in Chapter 1, for well posed problems, either the Dirichlet or the Neumann or a linear combination of the two, Robin, conditions are specified on each side of our polygonal domain. In other words, to find the solution to our PDE, it is necessary to perform a Dirichlet to Neumann map to recover unknown boundary conditions in their separated forms to then resolve the integrals (3.3). We will see in §3.3 that certain global relations that can be derived in several ways lead to numerical Dirichlet to Neumann maps. The numerical implementation of these maps for the case of the Helmholtz equation is central to this thesis and is explored in Chapters 4, 5, 6, and 7.

### 3.2.2 Solution for the Helmholtz Equation

Assuming that the solution to the Helmholtz equation $q(z, \bar{z})$ exists, and that it is sufficiently smooth on $\Omega$, it can be written as

$$q(z, \bar{z}) = \frac{1}{4\pi i} \sum_{j=1}^{n} \int_{L_{outj}} \exp \left( i\beta \left( kz + \frac{\bar{z}}{k} \right) \right) \hat{q}_j(k) \frac{dk}{k}$$

where $k \in \mathbb{C}$. The spectral functions $\hat{q}_j(k)$, $j = 1, \ldots, n$ are defined as

$$\hat{q}_j(k) = \int_{z_j}^{z_{j+1}} \exp \left( -i\beta \left( kz - \frac{\bar{z}}{k} \right) \right) \left[ q_n + \beta \left( \frac{1}{k} \frac{dz}{ds} + k \frac{dz}{ds} \right) q \right] ds$$

and the rays $L_{outj}$, $j = 1, \ldots, n$, oriented from 0 to $\infty$ are defined by

$$L_{outj} = \{ k \in \mathbb{C} : k = le^{-i\alpha_j} \}$$

and $l$ is the contour depicted in Figure A.1 of Appendix A. We are essentially rotating the contour A.1 by $\alpha_j$ clockwise. These contours are not as simple as 3.4 for reasons explained in Appendix A.
We can see that there are differences between (3.2) and (3.5), especially in the contours of integration for the solution. Reasons for the different contours are explained in Appendix A. However we can argue that the variations reflect the fundamental differences that the modified Helmholtz and the Helmholtz exhibit. On the one hand, the modified Helmholtz does not have any eigenvalues and is guaranteed to have a unique solution as long as appropriate boundary conditions are supplied. On the other hand, the Helmholtz equation has eigenmodes and eigenvalues. We shall explore this issue in more detail - finding explicit eigenvalues on the unit square and numerical approximations for the trapezoid and exploring their effects on the Dirichlet to Neumann map - in Chapter 6. For conceptual purposes here we shall consider a one-dimensional case of how the eigenvalues of the discretized system for the Helmholtz and the modified Helmholtz equations differ. We shall start with the usual convention for our PDEs:

$$\Delta q - \lambda q = 0$$  \hspace{1cm} (3.8)

Considering only the Laplacian for now ($\lambda = 0$), we discretized in one dimension with a three points stencil as is usually done in finite difference methods. The resulting $n \times n$ matrix $A$ is as follows:

$$A = \begin{pmatrix}
-2 & 1 & 0 \\
1 & -2 & 1 \\
0 & \ddots & \ddots & 0 \\
& \ddots & \ddots & 1 \\
0 & 1 & -2
\end{pmatrix}$$  \hspace{1cm} (3.9)

As was shown in more general terms in [9], the eigenvalues $\lambda_s$, $s = 1, \cdots, n$ of (3.9) can be explicitly found to be

$$\lambda_s = -2 - 2 \cos \left( \frac{s\pi}{n+1} \right)$$  \hspace{1cm} (3.10)

Of course we can also bound the eigenvalues using the Gershgorin theorem yielding the bound $|\lambda_s - 2| \leq 2$ and as long as we do not take $n$ to be $\infty$, none of the eigenvalues will be zero. With the introduction of non-zero $\lambda$, a shift will occur on the diagonal of (3.9). The diagonal elements become $-2 - \lambda$. We can deduce that the eigenvalues will shift depending on $\lambda$. For $\lambda > 0$ the eigenvalues
will shift to the left and thus because of this, the discretized modified Helmholtz equation never has zero eigenvalues. This can also be understood explicitly as the modified Helmholtz does not exhibit eigenfunctions nor eigenvalues. For the case of \( \lambda < 0 \) the eigenvalues will shift to the right, causing them to pass through zero one by one. Thus, for the Helmholtz equation, we have zero determinant for \( \lambda = -\lambda_s \). That is, for special values of \( \lambda \) the discretized system has no solution or infinitely many solutions. This can also be understood for the explicit version of the Helmholtz equation. These issues will be explored further in Chapter 6.

We mention here that the solutions of the Helmholtz equation and the modified Helmholtz equation on arbitrarily shaped convex polygons is an integral solution involving certain transforms of the boundary conditions. Discussions with Professor James Curry pointed out the interesting fact that for regular geometries such as squares, rectangles, and even certain triangles, separation of variables yields solutions as combination of trigonometric solutions. The case of the square and rectangles are well known to be solvable via Fourier superposition while the case of certain triangles are more involved and can be solved following the work of Lamé [25], [24], [31]. Those three geometries admit trigonometric solutions while the Fokas methodology presents solutions which bear no resemblance to trigonometric superpositions. Though in very simple cases the integral solutions (3.2) and (3.5) do yield trigonometric modes it is not clear to me if the same is true in general.

As we have mentioned earlier, in order to use equations (3.2) and (3.5), we need to be able to acquire both the Dirichlet and the Neumann boundary conditions. We present more results by Athanassios Fokas in §3.3 which will allow us to do that via certain global relations.

### 3.3 The Global Relations: Dirichlet to Neumann Maps

As was done for the Helmholtz equation in §2.2 we can derive global relations for the modified Helmholtz equation. The relations are then

\[
\int_{\partial\Omega} e^{-i\beta(kz-\pi/k)} \left( \left| \frac{dz}{ds} \right| q_n + \beta \left( \frac{1}{k} \frac{d\sigma}{ds} + k \frac{dz}{ds} \right) q \right) ds
\]

(3.11)
together with its Schwartz conjugate:

$$\int_{\partial \Omega} e^{i\beta (kz - z/k)} \left( \frac{dz}{ds} \right) q_n + \beta \left( 1 \frac{dz}{ds} + k \frac{d\bar{z}}{ds} \right) q \ ds$$  \hspace{1cm} (3.12)

As a reminder from §2.2, the global relations for the Helmholtz equation are:

$$\int_{\partial \Omega} e^{-i\beta (kz + z/k)} \left( \frac{dz}{ds} \right) q_n + \beta \left( -1 \frac{dz}{ds} + k \frac{d\bar{z}}{ds} \right) q \ ds$$  \hspace{1cm} (3.13)

together with its Schwartz conjugate:

$$\int_{\partial \Omega} e^{i\beta (k\bar{z} + z/k)} \left( \frac{dz}{ds} \right) q_n + \beta \left( -1 \frac{dz}{ds} + k \frac{d\bar{z}}{ds} \right) q \ ds$$  \hspace{1cm} (3.14)

With the global relations (3.11), (3.12), (3.13), and (3.14), we have the preliminary tools to compute numerical Dirichlet to Neumann maps. With these maps it is then possible to compute the solutions on the interior of polygons. We shall see how the Dirichlet to Neumann maps can be implemented numerically in Chapter 4.
Chapter 4

Preliminary Tests with Analytic Solution: Dirichlet To Neumann Map

In this chapter we first explore the Dirichlet to Neumann map by implementing it for the Helmholtz equation on various geometries and comparing the results with explicit solutions on those domains. We also implement the map for the modified Helmholtz for completeness. The purpose here is to get a feel for the performance of the method as we vary some parameters that we have available. The global relations in use in this section are those of §3.3. The implementation for the Dirichlet to Neumann maps was performed in Matlab and is extensively similar to the implementation of the D2N for the Laplace equation by Fornberg and Flyer in [16], and some earlier work [14], some of which is briefly described in this section. Regardless of the geometry we are working on, the global relations are already in terms of a parameter $s$ such that we can freely parametrize the boundaries. In all the cases we shall explore, the parametrization shall be over $[-1, 1]$.

The global relations for the Helmholtz equation and the modified Helmholtz equation and their Schwartz conjugates can be treated in the same way apart from with regards to small details. We shall therefore go through the implementation process for the Dirichlet to Neumann map. One of the key concepts allowing for the numerical implementation of a D2N map is that $k$ is an arbitrary complex number. As such, we can create a system of equations for as many $k$ values as we wish. We shall then sample $N$ points in the $k$-plane, constituting a $k$-cloud of $k$ values. Different sampling strategies and their effects on the performance of the map will be explored in Chapter 5. We then have an arbitrarily large system of equations. The unknowns are functions of $s$, either Dirichlet or
Neumann functions on the boundaries. We now perform the second key step in the method. As was done in [16], we represent the unknown functions with Legendre polynomials of degree $M$. For each side $v = 1, \cdots, n$, we expand

$$\sum_{m=0}^{M} c_m^{(v)} P_m(t) \quad (4.1)$$

One advantage of this expansion is that the integrals of our unknowns now have for integrands an exponential times a Legendre polynomial. Such integrals can be done in closed form using the identity

$$\int_{-1}^{1} e^{\alpha t} P_m(t) dt = \frac{2^{m+1} \alpha^m m!}{(2m+1)!} \, {}_{0}F_{1} \left( m + \frac{3}{2}, \left( \frac{\alpha}{2} \right)^2 \right) = \frac{\sqrt{2\pi\alpha}}{\alpha} I_{m+\frac{1}{2}}(\alpha) \quad (4.2)$$

where $I_{m+1/2}$ are Bessel functions of the first kind of order $m + 1/2$. To recapitulate, for each $k$-value, the global relation is written out e.g.

$$\int_{\partial \Omega} e^{-i\beta(kz+z/k)} \left( \frac{dz}{ds} q_n + \beta \left( -\frac{1}{k} \frac{dz}{ds} + k \frac{dz}{ds} \right) q \right) ds \quad (4.3)$$

If for example a Dirichlet condition is prescribed on each side then all $q$ are known functions or data and all $q_n$ are unknown. We separate the knowns and the unknowns and compute the integrals involving the known functions either explicitly or numerically using adaptive quadratures. These computed integrals then become the right hand side of our linear system. The unknowns we expand with Legendre polynomials of degree $M$. Contemplating that we have $n$ sides to our polygon, we then have $n \times (M+1)$ unknown Legendre coefficients to solve for. Letting each row of our unknown matrix represent the global relation for each $k$-value, we then place in the first $M + 1$ columns the $M + 1$ coefficients for side 1. In rows $M + 2$ to $2(M + 1)$ the $M + 1$ coefficients of side 2, and so on for all side up to side $n$. This setup can be viewed schematically in figure 4.1 for the case of a 4-sided polygon. We make a small note here that sampling $N$ $k$-values in the complex plane yields $2N$ equations in our system since we use both the global relation and its Schwartz conjugate simultaneously. That is, sampling $n \times M$ $k$-values will overdetermine our linear system.
In order to solve for the coefficients, we overdetermine the linear system by sampling many \( k \)-values in the \( k \)-plane and using Matlab’s ‘\( \backslash \)’ solver. A sample code showing the how the method is implemented in detail is shown in Appendix F. Once the coefficients are solved for, we can reconstruct our Legendre representation and have an approximation of degree \( M \) for our unknowns. We compare this reconstruction to explicitly known boundary information derived from explicit solutions over our domains. We first test the method with the Helmholtz equation on the unit square and a rectangle in §4.1.

### 4.1 The Helmholtz Equation on the Unit Square and Rectangle

We will first consider the Helmholtz equation on the unit square and a rectangle. Both domains and the particular boundary conditions that were used are shown in Figures 4.2 and 4.3.
For both the unit square and the rectangle, we solve the Helmholtz equation with the prescribed boundary conditions via standard separation of variables. For the square, we get the following solution:

\[
q_{\text{square}} = \text{csch}\left(\frac{1}{4}\sqrt{9\pi^2 - 16\beta^2}\right) \sin \left(\frac{3\pi y}{2}\right) \sinh\left(\frac{1}{4}(1 - 2x)\sqrt{9\pi^2 - 16\beta^2}\right) \\
+ \text{sech}\left(\sqrt{\pi^2 - 4\beta^2}\right) \sin (\pi x) \left(\sqrt{\pi^2 - 4\beta^2} \cosh \left((y - 1)\sqrt{\pi^2 - 4\beta^2}\right) - \sinh\left(y\sqrt{\pi^2 - 4\beta^2}\right)\right)
\]  \hspace{1cm} (4.4)

For the rectangle we get a simpler solution since the boundary conditions are simpler:

\[
q_{\text{rectangle}} = \cosh \left((y - 1)\sqrt{\pi^2 - 4\beta^2}\right) \text{sech} \left(\sqrt{\pi^2 - 4\beta^2}\right) \sin (\pi x)
\]  \hspace{1cm} (4.5)
The solutions to the Helmholtz equation on these domains with $\beta = 1$ and with the prescribed boundary conditions can be seen in Figures 4.4 and 4.5. The solution for the square is sinusoidal in both $x$ and $y$, while the solution over the rectangle is sinusoidal in $x$ and hyperbolic in $y$. We note that for $\beta > \pi/2$, the solution will actually become sinusoidal in $y$ for the rectangular geometry as $\pi^2 - 4\beta^2$ will be negative, forcing the square root to be imaginary and thus the hyperbolic cosine to be a cosine. As $\beta$ grows larger, we have more oscillatory behavior as can be seen with $\beta = 5$ in Figures 4.6 and 4.7.

Figure 4.4: The solution (4.4) on the square domain with $\beta = 1$.

Figure 4.5: The solution (4.5) on the rectangular domain with $\beta = 1$. 
Figure 4.6: The solution (4.4) on the square domain with $\beta = 5$. We note the more oscillatory behavior on the domain.

Figure 4.7: The solution (4.5) on the rectangular domain with $\beta = 5$. We note of course the more oscillatory behavior on the domain.

At this point, we replace our parameter $s$ with $t$ and parametrize the four sides of each domain with $-1 \leq t \leq 1$. We define the four corners of our polygon with $z_1, \ldots, z_4$, with $z_1 = 0$ being at the origin. We then define sides $S_j$, $j = 1, \ldots, 4$ as the segments joining corners $z_j$ to $z_{j+1}$. In both geometries of this section, and all other sections, $S_1$ is then the bottom side joining corners $z_1 = 0 + 0i$ and $z_2 = 1 + 0i$. We traverse the boundary of polygons in the counterclockwise manner. Useful quantities of the parametrization for the unit square are summarized in Table 4.1.

Table 4.1: Parametrization in $t$ for the unit square

<table>
<thead>
<tr>
<th>Side</th>
<th>Formula</th>
<th>$\frac{dz}{dt}$</th>
<th>$\frac{dz}{ds}$</th>
<th>$\frac{ds}{dt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Side 1</td>
<td>$z = \frac{t+1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>Side 2</td>
<td>$z = 1 + i\frac{t+1}{2}$</td>
<td>$\frac{1}{2}i$</td>
<td>$\frac{1}{2}i$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>Side 3</td>
<td>$z = i - \frac{t-1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>Side 4</td>
<td>$z = -i\frac{t+1}{2}$</td>
<td>$\frac{1}{2}i$</td>
<td>$\frac{1}{2}i$</td>
<td>$\frac{1}{2}$</td>
</tr>
</tbody>
</table>

Similar tables can be constructed for all geometries in consideration in this thesis. At this point, we sample a large $k$-cloud in the complex $k$-plane and construct our linear system. For both the square and the rectangle, we shall prescribe the boundary conditions as in Figures 4.2 and 4.3. That is, we supply explicit expressions for $q^{(1)}$, $q^{(2)}$, $q^{(3)}$, and $q^{(4)}$, where the subscript $n$ represents
a Neumann condition and the absence of that subscript represents a Dirichlet condition. Our task is to recover \( q_n^{(1)} \), \( q_n^{(2)} \), \( q_n^{(3)} \), and \( q_n^{(4)} \). Using a Legendre expansions of degree \( M \) for each side, we typically use \( 4M \) or \( 5M \) \( k \)-values to overdetermine the linear system. For the following results that general guideline is followed. We see next some of the results of the method.

In Figure 4.8 we see that the four recovered functions cannot graphically be differentiated from their explicit counterparts. We show both the explicit function and the numerical approximation together for two reasons: first, it gives us an opportunity to see what the functions look like, and second, in some test cases (e.g. Chapter 7) the two will differ graphically. On the bottom row we
see the errors associated with each recovered functions. At worst, we recover the unknowns with an order $10^{-6}$ error. The errors behave very well except near the corners of the square. This is to be expected as even though the function values are continuous at the corners, the derivatives are not necessarily continuous. In section §4.3 we avoid this problem by choosing an analytic solution over our domain to ensure that the function and all of its derivatives are continuous everywhere, including at the corners. We also try to address the problem of singularities present at the corners in Chapter 7. We note that the results can be improved by using more terms in the Legendre expansion and also by overdetermining the system even more. To get a better feel for the performance of the method for this particular geometry, we will look at the worst error along all four sides as we increase the degree of the Legendre expansion $M$. In Figure 4.9 we can see from the linear trend of this log-linear plot that the method actually is spectrally convergent.

Figure 4.9: Maximum of the errors along all four sides of the square plotted against the degree of the Legendre expansion along each side. Note on this log-linear plot that we have spectral convergence!
We now look at some results regarding the rectangular geometry.

![Graphs showing approximations and errors for Legendre degree 15 on a rectangle.](image)

**Figure 4.10:** Legendre degree is 15. 60 $k$-values. Top row: the four recovered functions on the rectangle superimposed with their explicit counterparts. Bottom row: the errors between the two. The maximum error along all four sides is on the order of $10^{-8}$.

In Figure 4.10 we can see that again, the four recovered functions are graphically indiscernible from their explicit counterparts. As with the square, the error is well bounded. With the chosen parameters of $M = 15$ and $N = 4M = 60$, the worst error is bounded at $10^{-8}$. We note that the behavior at the corners is not as bad as with the square. This can be understood from the more simplistic boundary conditions imposed on the rectangle.
4.2 The Helmholtz Equation on a Trapezoid

We now consider the Helmholtz equation on another polygonal domain. The domain in consideration along with the particular set of boundary conditions used can be seen in Figure 4.11 while the solution itself can be seen in Figure 4.12.

As with the square and rectangular domains, we parametrize the trapezoid with $-1 \leq t \leq 1$ to arrive at the parameterizations of the four sides shown in Table (4.2):
We implement the numerical Dirichlet to Neumann map as with the case of the square and the rectangle and arrive at the following results:

Figure 4.13: 100 \( k \)-values, Legendre degree \( M \) is 20. Top row: the four recovered functions on the trapezoid compared to their explicit counterparts. Bottom row: the errors in that comparison. The maximum error along all four sides is on the order of \( 10^{-6} \). The error is poor near the corner joining sides 3 and 4 because there is a singularity there. See Chapter 7 for more information on the nature of the singularity and a framework for its possible treatment.
From Figure 4.13 we can see that the errors are acceptable nearly everywhere except at the corners, where there are singularities. An attempt shall be made in Chapter 7 to address the issue of singularities in the corners. For now however we are pleased with the results of the Dirichlet to Neumann map. We continue our testing of the method with the modified Helmholtz equation now.

### 4.3 The Modified Helmholtz on a Trapezoid

For the case of the modified Helmholtz equation we perform very similar tests to arrive at similar results. The performance of the Dirichlet to Neumann map for the modified Helmholtz can be seen in the following Figure with $M = 20$:

![Graphs showing approximation and errors for modified Helmholtz equation](image)

Figure 4.14: 100 $k$-values, Legendre degree is 20. Top row: the four recovered functions on the trapezoid compared to their explicit counterparts. Bottom row: the errors in that comparison. The maximum error along all four sides is on the order of $10^{-9}$. 
From Figure 4.14 that the D2N for the modified Helmholtz equation perform equally well as the one for the Helmholtz equation. We do still have singularities in the corners of the polygon. We wish to see if the method is also spectral. As with the Helmholtz equation, we test the D2N for increasing values of $M$:

![Graph](image)

Figure 4.15: Maximum of the errors along all four sides of the trapezoid plotted against the degree of the Legendre expansion along each side. Note that this is a log-linear plot.

Figure 4.15 suggests that the Dirichlet to Neumann map for the modified Helmholtz equation is also spectrally convergent. We do note however the jagged behavior of the curve and leave its exploration to Appendix E in which we use minimal polynomial approximations to find that the jagged behavior is not an artifact of the D2N. The problem arises from the difficulty for certain degrees of polynomial expansions to approximate certain curves. In our case, one of the sides just happens to be hard to approximate with odd $M$ Legendre expansions. This would explain the jagged behavior. This concludes our preliminary tests of the Dirichlet to Neumann maps for the Helmholtz equation and the modified Helmholtz equation on various geometries.
The purpose of this chapter is to study the effects of the k-cloud distributions on the performance of the D2N map method. These tests were run with the Helmholtz equation only but it is expected that the modified Helmholtz equation would yield similar results. The set-up is the same as in all chapters of this thesis, forming an overdetermined linear system by sampling $N$ $k$-values in the complex $k$-plane and solving it. Initially, $k$-values were formed by sampling uniformly in the $k$-plane and keeping all nodes within a radius $R$. As we will see, this radius is rather important. We describe that radius $R$ as being the reach of the $k$-cloud or of the distribution. As is done by Fornberg and Flyer in [16], we typically use Halton nodes which we will define later (§5.2) because it turns out that they work well for all situations with the Laplacian, the Helmholtz equation and the modified Helmholtz equation. To get a good feel as to why Halton nodes are appropriate in all situations we perform a series of optimizing tests in this chapter. One of the key conclusions of this study is that the reach $R$ of a distributions is more impactful than the distribution itself. Our tests here involve implementing Matlab’s genetic algorithm toolbox to try and optimize particular cloud sets with various parameters and then testing several distributions for varying $R$.

5.1 Genetic Algorithm on the Unit Square and the Trapezoid

We start with the Helmholtz equation on the unit square with rather simple boundary conditions as follows: $q^{(1)} = \sin (\pi x)$, $q^{(2)} = q^{(3)} = q^{(4)} = 0$. The solution is found via separation of variables and the complementary sides are explicitly calculated to be used for comparison. We use
the D2N map to numerically recover the unknowns on all sides and compare. We then implement the genetic algorithm to try and find an optimal pattern of complex k values in the plane that would minimize the maximum error along the perimeter of the polygon. The initial distribution is a cloud of 50 complex k-values randomly distributed in a square of side length 30 centered at the origin. Two different cost functions are separately set to be minimized in two different tests. We first minimize the errors between the analytical expressions for the unknown boundary functions and the D2N map. The second series of tests is set to minimize the condition number of the linear system that is used to solve D2N map. Our linear system being in general not square, we compute the condition number with Matlab’s "cond" command. The condition number is computed using the 2-norm, or the ratio of the largest singular value to the smallest. The singular values are computed via a singular value decomposition. Each test is discussed bellow. Consistently throughout these tests, $\beta = 1$. The case for other $\beta$ is studied a bit further in Chapter 7. The degree of the Legendre polynomial used in each test is denoted as $M$, which we will vary as well. Our first test involves minimizing the error using $M = 7$ for the specified boundary conditions. We see our first results in Figures 5.1, 5.2, 5.3.
Figure 5.1: Set of clouds by generations that minimize the error. All plots are on the same scale. $M = 7$. Note that the distributions stagnate after the 50th generation. In conjunction with the next Figure 5.2 we see that the error is consistently decreasing for as long as the distributions change.

Figure 5.2: Error vs Generations. $M = 7$. Note that the error decreases until the new generations do not differ from the older ones.
Figure 5.3: Condition number vs Generations. $M = 7$. The condition number drops by one order and then stalls after the 50th generation.

We can see that out of the initial random distribution, the $k$-values are spreading farther out and then contracting slightly. From the error plot in Figure 5.2 we see that the error stabilizes at the 50th generation. It is also apparent from the distribution plots in Figure 5.1 as the distributions cease to change after the 50th generation. The condition number behaves in the same way. Overall, the error drops consistently but only by about almost one order of magnitude, while the condition number dropped by one full order. We carry on the tests now with $M = 11$ in Figures 5.4, 5.5, and 5.6.
Figure 5.4: Set of clouds by generations that minimize error. $M = 11$. We note that the distributions stop varying after about the 60th generation.

Figure 5.5: Error vs Generations. $M = 11$. As with the $M = 7$ test, the error falls and then stagnates when the generations stop changing.
Figures 5.4, 5.5, and 5.6 display the same trend as with $M = 7$, but the overall performance is better. From these results we can see that the condition number of the linear system being solved decreases as error is being minimized by the Genetic Algorithm. At this point we start to see a relationship between the condition number and the error. Though there we cannot discern any clear patterns in the optimal distributions, we do see some general features. The nodes seem to spread out from their initial bounding box. The nodes seem to be more concentrated near the origin. The second feature might be due to the fact that the nodes started out seemingly crowded but we notice that at the fourth generation, nodes were not concentrated in the center at all. It appears that the distributions, after spreading out, became denser near the origin. Keeping this in mind we perform more tests but we now use the genetic algorithm to minimize the condition Number of the linear system instead and see what happens to the error. We expect to see the error follow the condition number as generations pass. First results with $M = 7$ are shown in Figures 5.7, 5.8, and 5.9.
Figure 5.7: Set of clouds by generations that minimize the condition number of the linear system $A$ being solved. $M = 7$. We note that the distributions stop varying after about the 60th generation just as the case of minimizing the error.

Figure 5.8: Error vs generations. $M = 7$. The error improves but does not consistently decrease as we are only optimizing the condition number. However, the error is always acceptable for the chosen parameters.
Figure 5.9: Condition number vs generations. $M = 7$. The condition number decreases nicely as the generations pass especially in the first few generations. It seems that the spreading of the points in the plane be more important than the reorganizing of the points within a certain distance from the origin.

From Figures 5.7, 5.8, and 5.9, we see very similar results. We will now look at the generational changes in some more detail. It seems that in the first distribution changes, the points spread farther out from the origin. That is when the condition number and the error are both changing the most. Once the points have spread out in the complex plane it seems that they simply reorganize a little bit, having little effect on the error. From this behavior which can be seen in all plots from this chapter, it seems that the most important feature of the distributions is how far out in the complex plane they are and not the their particular structure. We shall investigate this possibility some more in §5.2. We repeat the last test with $M = 11$ in Figures 5.10, 5.11, and 5.12.
Figure 5.10: Set of clouds by generations that minimize the condition number of the linear system $A$ being solved. $M = 11$. We note that the distributions stop varying after about the 60th generation just as the case of minimizing the error.

Figure 5.11: Error vs generations. $M = 11$. In all, the error drops down by two orders of magnitude.
Figures 5.7 - 5.12 show us that overall, minimizing the condition number does minimize the error. This general trend is a good feature of the D2N map as it allows us to monitor the error by monitoring the condition number of the linear system. To ensure that we are not being to narrow in our search for distribution, we repeat the minimizing process with different boundary conditions and different geometries. The tests are numerous and lengthy. We will show next a result with different boundary conditions on a trapezoid but we shall refrain from displaying everything that was done with distribution tests. Other tests were conducted with higher degrees of the Legendre polynomials, more $k$-values, less $k$-values and other boundary conditions. The conclusions from all these tests are the same. To reinforce this statement, we now look at similar tests on the trapezoidal domain.

We repeat this entire process but with the trapezoidal shape of Figure 4.11 with the boundary conditions shown there. We omit displaying all the plots for all test cases again as it would be very lengthy. We only display one series of plots here associated with $M = 16$ and 50 $k$-values, while minimizing the condition number and monitoring the error. These results can be seen in Figures 5.13, 5.14, and 5.15.
Figure 5.13: On the trapezoid of Figure 4.11. Set of clouds by generations that minimize the condition number of the linear system $A$ being solved. $M = 16$. We note that the distributions stop varying after about the 60th generation just as the case of minimizing the error and the condition number for the square case. The features are very much the same as with the square tests: odes spread out from the original unit box and then reorganize themselves.

Figure 5.14: Error vs generations. $M = 16$. Even though we are minimizing the condition number in this test, the error decreases from $10^{-3}$ to $10^{-8}$. The error is evidently better with $M = 16$ but the improvement is rather significant.
Again, we see in Figures 5.13, 5.14, and 5.15 the same trend as with all previous cases on the square problem. The condition number significantly drops by ten orders of magnitude, while the error follows, dropping by five orders of magnitude. To conclude these initial tests with Matlab’s Genetic Algorithm, it seems that there are features shared by all tests. Those are the following:

- $k$-nodes spread from the original bounding box out in the complex plane. At that moment, both the error and the condition number of the linear system decrease the most.

- Once spread out, the points reorganize themselves with little to no effect on either the error or the condition number.

- The final distributions seem to be more concentrated at near the origin and more spread out away from the origin.

From the first feature observed, it seems that the best way to minimize the error is to spread points out in the complex plane. We shall re-define here the reach $R$ of a given distribution by how far out that distribution spans in the complex plane. We wish to test various distributions to see how the error and the condition number fare with respect to the reach of each distribution. In the following section, four different cloud distributions are tested throughout to see how they fare against the genetic algorithm’s results.
5.2 Distribution Tests

In this section we shall test various distributions against various parameters to see how those parameters influence the performance of the method in terms of the maximum error along each side and the condition number of the linear system being solved. The tested distributions are described and then tests and comparisons with §5.1 are carried out. The only parameters that are varied in the following tests are the reach $R$ of these distributions, that is, how far out on the complex plane these distributions reach out, the number of points $N$ being used, and the degree of the Legendre polynomial expansion $M$.

The Distributions

- **2-D Gaussian (Normal) Distribution** The first distribution is an attempt at producing a 2-D Gaussian distribution with a higher point density near the origin and some outlying points. This is done by sampling $N$ points from a normally distributed pseudo random distribution and then choosing half to be complex. The real and complex points are added together to generate $N/2$ complex points with the desired properties. With only a few points the distribution does not clearly exhibit the mentioned properties but with many points such as in the results of Figure 5.21, the features are more apparent.

- **Random Distribution** The second tested distribution is a pseudo random distribution in a square centered at the origin. As was done with the 2-D Gaussian distribution, we sample $N$ random points and pick half to be imaginary. We then add the imaginary points to the real ones to create $N/2$ pseudo random points complex points. The points are shifted and scaled to be within a square centered at the origin.

- **Uniform Distribution** The third distribution is a uniform distribution limited to a disk. We chose to limit the points to a disk as we want to limit the effects of preferential directions that would be more present if the points were simply allowed to be in a rectangular box. Again the disk feature is not apparent with few points but can clearly be seen in the results.
of Figure 5.21. In order to fit $N$ points on a disk, we approximate the number of points necessary on a square to result in approximately $N$ points on a disk. The approximation is rather crude but sufficient for our limited tests. We then have about $N$ points on a disk or radius $R$. Precaution is taken when using the Uniform distribution. From equation 3.13 we see that having $k = 0$ can be problematic so we remove that point from the distribution.

- **Halton Distribution** The last distribution is a Halton distribution on a disk. Halton points are defined and described very well in Fasshauer [10]. Applications range from quasi-Monte Carlo simulations (e.g. Halton [19]) to ray tracing (e.g. Wong et al. [39]). Halton points ensue from a hierarchical generalization of van der Corput sequences. The generalization can be found in [19] and is reviewed in Wong [39], [10], and other sources. Numerical algorithms exist in the literature including original code by Halton and Smith [20] and code available through Matlab central (haltonseq). The code used in this thesis is an efficient Matlab implementation created by Professor Fornberg. Even though Halton nodes are deterministic, they are considered quasi-random due to their low discrepancy. That is to say, they seem random and yet they do not clump up or spread out like random nodes might. We generate Halton nodes on a unit square centered at the origin and sample only the ones on a disk or radius $R$ so as to not impose preferential directions.

The random and 2-D Gaussian distributions are originally sampled in a box side length 2 centered at the origin and then stretched out by a factor of $R$ so as to mimic the reach factor on the disks with the Halton and the Uniform distributions. With these distributions at hand, we test the effects of the parameter $R$ on both the error and the condition number. That is, we take each distribution initially with $R = 1$ and then increase $R$ as we test each distribution. We also display the distribution which yielded the best error. The tests are run on the trapezoidal domain of Figure 4.11 with slight changes in the boundary conditions to make sure that we are not displaying results of the same test problem every time. The only changes were that we set $q^{(2)} = q^{(4)} = 0$. Everything else is the same. Please do note that extensive testing was performed on other geometries with
other boundary conditions and parameters. The results shown here are exemplary and do in fact well represent all tests performed. The test results are shown in Figures 5.16 to 5.21.

Figure 5.16: Error (Top row) and condition number (middle row) vs reach of distributions (one distribution per column from left to right starting with the 2-D Gaussian, the Random, the Uniform, and the Halton distribution). The bottom row shows the distribution at its optimal reach $R$. $M = 7$, with 50 $k$-values. Note that all plots on a given row is on the same scale. We see that the reach $R$ has a drastic effect on the performance of the method for all distributions.

Figure 5.16 is interesting. First of all, the reach $R$ of each distributions definitely has an impact on the performance of the method. We can clearly see that each distribution has its own optimal regime, a range of $R$ values in which both the error and the condition number is minimal. We also note that the condition number is behaving similarly to the error. Another thing to notice is that within their optimal regimes, each distribution performs nearly equally well at around $10^{-4}$. We repeat this kind of plot with $M = 11$. 
From Figure 5.17 with $M = 11$ we can see similar behaviors than with $M = 7$, but the errors and the condition numbers are greatly reduced overall. The $R$ regime at which each distribution performs best is longer now, making it easier to use the method at its best. We also note that the relationship between the condition number and the error is even more apparent in this case. The random distribution seems to perform the best. The results here are appreciable. Perhaps we are at a good balance between the degree of the Legendre expansion and the number of equations used to solve the linear system. We would like to see if the optimal $R$ regime, or plateau, further lengthens if we increase $M$ to 16.
Figure 5.18: Error and condition number vs reach of distributions. $M = 16$, with 50 $k$-values. Note that all plots on a given row are on the same scale. We note that the method performs even better now but the optimal regime seems to have narrowed. This is most likely due to the fact that our system is becoming less and less overdetermined with 100 equations for 64 unknowns. However, the error is still considerably reduced. Note that the condition number is rather high in this test. It is also minimized when the error is minimized.

The tests in Figure 5.18 with $M = 16$ are similar to the previous two tests in that the reach $R$ is still a very important parameter. However we do start to see some changes. It seems that the curves are a little bit more sporadic. It also seems that the optimal plateau we were looking for has actually narrowed. The error is overall much better than with $M = 11$ but we are still concerned about the narrowed plateau. We could explain this behavior from the fact that we are overdetermining the linear system less and less, having only 100 equations for 64 unknowns. As a general guideline we want to use at least 4 to 5 times the degree of polynomial expansion for our equations. Regardless, we still have very nice error and a nice trend in condition number, even though rather large. We note that the random distribution and the Halton distribution perform the best in this case. We
would like to repeat these tests but with 500 $k$-values instead to ensure that the overdetermining of the system affects each test equally.

![Graphs showing error and condition number vs reach of distributions.](image)

Figure 5.19: Error and condition number vs reach of distributions. $M = 7$, with 500 $k$-values. Note that all plots on a given row are on the same scale. Note that the error decreases and then remains low, unlike with only 50 $k$-values where the error increased after reaching an optimal regime. The two random distributions seem to perform better, but only marginally so.

We can see in Figure 5.19 that, even with relatively low $M$, increasing the number of points definitely has positive impacts on the method. The optimal plateau widens a lot and the method seems to be less dependent on the reach of the first two random distributions. We can definitely say that overdetermining the system is positively impacting the method. We repeat the above test with $M = 11$. 
Figure 5.20: Error and condition number vs reach of distributions. $M = 11$, with 500 $k$-values. Note that all plots on a given row are on the same scale. We note that the feature are the same as with $M = 7$ but we did lose four orders of magnitude in the error.

We can see in Figure 5.20 that increasing $M$ by four decreased the error by four orders of magnitude, further indicating spectral convergence as was described in §4.3. The features are the same as with $M = 7$. Again, the Random and 2-D Gaussian distributions perform equally well as the Uniform and the Halton distributions do. We repeat the test but with $M = 16$. 
Figure 5.21: Error and condition number vs reach of distributions. $M = 16$, with 500 $k$-values. Note that all plots on a given row are on the same scale. We note that the error is reduced to $10^{-13}$. Also note that across all distributions and reach, the worst the method yields is $10^{-2}$. It is remarkable that by simply changing the reach of the method we decreased the error by ten orders of magnitude.

Figure 5.21 demonstrates the capabilities of the method. With 500 $k$-values, we reach an error of $10^{-13}$ at worst on all sides. It is also noted that the worst the method performs across all shown parameters is $10^{-2}$ which is quite good. However, by monitoring the condition number one could easily reduce the error simply by increasing the reach of the distributions. We gained ten orders of magnitude simply by changing the reach of each distribution. It is not clear why noise is present with $M = 16$ but we do note its presence. We do note in general that optimal regimes for $R$ shift to the right as we increase the degree of the Legendre polynomial.

What we see from Figures 5.19 to 5.21 is that across all degrees and number of $k$-values tested, the reach of each distribution is very influential on the performance of the method. When each test is compared to the results of the genetic algorithm, the error differs by a maximum of one
order of magnitude. Another feature of these tests that was also present in the genetic algorithm study is that the condition number of the linear system and the error of the D2N map method are always minimized at the same reach for each distribution. This is very useful as the condition number of the linear system is independent of the boundary conditions imposed on the polygon. It also allows us to control the error by simply monitoring the condition number. As schematically described in Figure 4.1, we know that the condition number is only dependent on the geometry of the problem and obviously on the number of \( k \)-values as well as their distribution and reach, but not on the boundary conditions imposed on the geometry. Given these results, we can implement the D2N map method on different geometries with various boundary conditions and hope to minimize the error by minimizing the condition number.

The following are key features of the genetic algorithm tests and distribution tests of this chapter:

- Overdetermining the system has a positive impact on the method, reducing the error and the condition number.

- Increasing the degree of the Legendre polynomial approximation is beneficial as long as we are overdetermining the linear system.

- The reach \( R \) of the distributions is has a significant impact on the performance of the method, especially with lower numbers of \( k \)-values.

- Each tested distributions perform equally well in their respective optimal regimes.

- The tested distributions perform within one order of magnitude compared to results of the genetic algorithm optimization with the same parameters.

This concludes our study of the effects of the distribution of \( k \)-values on the performance of the D2N map method in regards to both the error of the method and the condition number of the linear system that transpires from the method. Our conclusion is that given a polygon with prescribed boundary conditions, we can use the D2N map method to perform a Dirichlet to
Neumann map with accuracies that will improve as the condition number of the linear system is improved. The condition number seemingly can be improved simply by changing the reach of a preset distribution. The chosen distribution does not have a significant impact on the performance of the method so long as we use that distribution’s optimal reach. As such we will consistently use Halton nodes in our tests because of their aforementioned properties.
Chapter 6

The Helmholtz Equation: an Eigenvalue Problem

In this chapter we shall concern ourselves with the eigenfunctions and eigenvalues of the Helmholtz equation and their effects on the numerical Dirichlet to Neumann map. As was briefly mentioned in §3.2.2, we expect the eigenvalues of the Helmholtz equation to be problematic in the D2N map. Traditional results are investigated and numerical results are explored. The effects of the eigenfunctions when $\beta$ is an eigenvalue of the Helmholtz equation are investigated for various geometries and generally for any boundary conditions on those geometries. We first start our discussion with the eigenvalue problem on the unit square. The form of the Helmholtz equation used in this chapter is the following in equation (6.1).

$$\frac{\partial^2 q}{\partial x^2} + \frac{\partial^2 q}{\partial y^2} + 4\beta^2 q = 0$$  \hspace{1cm} (6.1)

6.1 On the Unit Square

We have our definition of the Helmholtz equation with Helmholtz constant $4\beta^2$ on the unit square with boundary conditions shown in Figure 6.1. The boundary conditions are summarized as follows: zero Dirichlet boundary conditions on sides 1, 2, and 4, and a zero Neumann boundary condition on side 3. We chose this set of boundary conditions because most tests in this thesis include a Neumann boundary condition on side 3. We understand that eigenmodes are present for other configurations including zero Dirichlet conditions on all sides or any combination of Dirichlet or Neumann boundary conditions. However, when implementing the Dirichlet to Neumann map, we always chose side 3 to be a specified Neumann boundary condition. We therefore look for
eigenmodes with boundary conditions that match in type as was similarly done for the Laplacian in [23].

![Figure 6.1: Square domain with boundary conditions that will yield our eigenmodes. Note that the top side is a normal derivative.](image)

With these boundary conditions we find eigenmodes that satisfy the Helmholtz equation for particular values of $4\beta^2$, the eigenvalues associated with our choice of boundary conditions. These eigenvalues and eigenmodes are very important to us as they have a profound impact on the Dirichlet to Neumann map implementation.

### 6.1.1 Eigenmodes and Eigenvalues

For the case of the unit square, we find explicit expressions for the eigenmodes in terms of two wavenumbers $m$ and $n$. The eigenmodes are as follows:

$$q_{m,n}(x, y) = (-1)^{n+1} \sin (m\pi x) \sin \left( (n - \frac{1}{2})\pi y \right)$$

(6.2)
These eigenmodes occur only for specific values of the Helmholtz constant:

$$4\beta^2_{m,n} = \pi^2 \left( m^2 + (n - \frac{1}{2})^2 \right)$$  \hspace{1cm} (6.3)

For later purposes we derive $\beta_{m,n}$ to be (6.4) and refer to them as the eigenvalues of (6.1):

$$\beta_{m,n} = \frac{\pi}{2} \sqrt{m^2 + (n - 1/2)^2}$$  \hspace{1cm} (6.4)

We tabulate here numerical values of some of the eigenvalues $\beta_{m,n}$ for $m, n = 1, 2, \ldots, 6$ in Table 6.1, organized by wavenumbers:

Table 6.1: The first 36 eigenvalues of the Helmholtz equation on the square domain ordered by wavenumbers.

<table>
<thead>
<tr>
<th>Wavenumbers</th>
<th>$n = 1$</th>
<th>$n = 2$</th>
<th>$n = 3$</th>
<th>$n = 4$</th>
<th>$n = 5$</th>
<th>$n = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = 1$</td>
<td>1.7562</td>
<td>2.83179</td>
<td>4.2295</td>
<td>5.71778</td>
<td>7.24101</td>
<td>8.78102</td>
</tr>
<tr>
<td>$m = 2$</td>
<td>3.2382</td>
<td>3.92699</td>
<td>5.0290</td>
<td>6.33208</td>
<td>7.73527</td>
<td>9.19285</td>
</tr>
<tr>
<td>$m = 3$</td>
<td>4.7773</td>
<td>5.26861</td>
<td>6.13416</td>
<td>7.24101</td>
<td>8.49538</td>
<td>9.84101</td>
</tr>
<tr>
<td>$m = 5$</td>
<td>7.8931</td>
<td>8.1998</td>
<td>8.78102</td>
<td>9.58701</td>
<td>10.5665</td>
<td>11.6758</td>
</tr>
</tbody>
</table>

To get a better feel of what the eigenmodes look like, a couple of them with corresponding $m$, $n$, and $4\beta^2_{m,n}$ are shown in Figure 6.2:
With our explicit expressions for the eigenmodes and the eigenvalues we have concrete support against which to compare the results of the next section. We note that for the case of the trapezoidal geometry of §6.2, we do not find explicit eigenmodes and eigenvalues and must compute them numerically.
6.1.2  Singular Values of the Error

The goal of this chapter is to find whether or not the Dirichlet to Neumann map suffers when \( \beta \) is near an eigenvalue of the Helmholtz equation for the geometry in question. The geometry in question being the unit square, we monitor the performance of the method for a wide range of \( \beta \) values, increasing in magnitude from near zero to eight. We initially monitor the error associated with the D2N for particular boundary conditions as we increase \( \beta \). We start with rather simple boundary conditions as follows: \( q^{(1)} = \sin(\pi x) \), \( q^{(2)} = q^{(3)} = q^{(4)} = 0 \). A solution is found and the complementary sides are explicitly calculated to be used for comparison. We use the D2N map to numerically recover the unknowns on all sides with \( M = 5 \), \( R = 15 \), and Halton nodes. We present our findings in Figure 6.3.

![Figure 6.3: Maximum error for all sides as we increase \( \beta \). Tick marks are at all eigenvalues \( \beta_{m,n} \). We see that the spikes coincide with a subset of eigenvalues \( \beta_{m,n} \) (at the vertical lines). The error being large for \( \beta > 3 \), we suspect that spikes are present at all \( \beta_{m,n} \) and are not visible.](image-url)
The first notable feature of the error as a function of $\beta$ is that there are spikes at various values of $\beta$. The location of the spikes are labeled and traced with vertical lines. The locations coincide with a subset of eigenvalues $\beta_{m,n}$ computed previously. The notable spikes are those corresponding to eigenvalues with $m = 1$, and $n = 1, 2, \ldots, 5$. Though this result is promising, we wonder why not all eigenvalues are visible. Perhaps they are present but not noticeable because of the large errors for $\beta > 3$. In order to reduce the error for varying $\beta$, we will modify certain parameters at hand, namely the each $R$ of the Halton distribution in use. We set the degree of the Legendre expansions to $M = 20$ and overdetermine the linear system. We compute the error again but for varying $\beta$ and $R$ and plot our results in Figure 6.4.

![Figure 6.4: Maximum error for all sides as we increase both $\beta$ and $R$. We see that there is a narrow range of $R$ values that yield low errors. The exponential nature of the integrands of the global relations can explain why the method implemented as such cannot be accurate for $k$-values near the origin of too far away in the complex plane.](image)
From Figure 6.4 we can see that there is a narrow range of $R$ values that yield acceptable errors for a given value of $\beta$. This result is in accord with the results of Chapter 5 in which we found that there were optimal regimes of $R$ values for $\beta = 1$ for which the error was minimized. We see the same here but we notice that the optimal $R$ regimes shift with increasing $\beta$. With this information at hand, we parametrize a curve $R(\beta)$ so as to force $R$ to be in the optimal range when $\beta$ is varied. With a parametrized version of $R$, start over and monitor the error as a function of $\beta$. The results are in Figure 6.5.

![Figure 6.5: Maximum error for all sides as we increase $\beta$. The reach is now a function of $\beta$. The vertical lines are placed at all $\beta_{m,n}$. It is hard to discern if the method is singular at each line.](image)

We can now see that as long as we keep $R$ in its optimal regime, the error is minimized. Though we clearly see that more of the eigenvalues are present, there is a lot of noise and it is hard to
discern if all eigenvalues are present. We therefore investigate near each eigenvalue $\beta_{m,n}$ with more resolution and see that the method is indeed singular at each eigenvalue. It is simply to hard to discern spikes on the plot. We do not include all the plots here as they are too numerous and are all very similar, exhibiting a spike in error near the eigenvalue. We do show this test but near $\beta = 3.23$, the third eigenvalue (ranked in magnitude) to see whether or not the error does spike. This close up plot is shown in Figure 6.6. We will also show more satisfactory results in §6.1.3.

![Figure 6.6: Maximum error for all sides as we increase $\beta$ close to the third eigenvalue (ranked by magnitude). The reach is now a function of $\beta$. We note that the error does indeed spike at the eigenvalue.](image)

From Figure 6.6 we see that the error does indeed spike at the third eigenvalue. It is simply to hard to see on Figure 6.5. The same tests were performed for all eigenvalues that do not visibly spike on Figure 6.5 to ensure that the error does indeed spike at each eigenvalue. It is the case indeed. We shall reinforce this observation in the next section.
6.1.3 Singular Values of the Linear System

Though we found all eigenvalues of the Helmholtz equation cause our numerical implementation of the D2N to be singular, we did so for a particular problem with specific boundary conditions. In order to arrive at results that do not depend on specific boundary conditions, we look at the linear system itself. As was mentioned in §4, the matrix generated by sampling N points in the complex $k$-plane depends only on the PDE and the geometry in question (all boundary information enters on the right hand side of the matrix equation). That is to say, it is independent of the boundary conditions imposed on the geometry. In view of this, we seek to monitor the condition number of the linear system as we vary $\beta$ to see if the method is singular at the eigenvalues $\beta_{m,n}$. In view of the process of §6.1.2, we plot the condition number of the linear system versus both $\beta$ and $R$. The results are seen in Figure 6.7.

![Figure 6.7](image_url)

Figure 6.7: Condition number of the linear system as we increase $\beta$. We see the same behavior as in Figure 6.4, namely that there is a regime of $R$ values that minimizes the condition number.
From Figure 6.7 we see the same behavior as in Figure 6.4. That is, there is a narrow range of $R$ values that minimizes the condition number. In order to increase our chances of finding spikes in the condition number, we parametrize the reach $R$ as a function of $\beta$ again to ensure that for all $\beta$, the radius is in its optimal range. Though countless parameterizations are possible, a particular one that worked well for our tests is given here:

$$R(\beta) = -796.1 - 7415.6e^{-\beta^{1/4}} + 14504e^{-\beta^{1/3}} - 4884.6e^{-\beta^{1/2}} - 130.3e^{-\beta} + 96\beta - 5\beta^2 + 0.12\beta^3 \quad (6.5)$$

With this version of $R(\beta)$, we compute the condition number of the linear system as a function of $\beta$ from near zero to eight. The degree of the Legendre polynomial expansion was set to $M = 25$. We can see the resulting plot in Figure 6.8:

![Figure 6.8: Condition number of the linear system as we increase $\beta$ with $R(\beta)$. The vertical lines are placed at all $\beta_{m,n}$. We see that all eigenvalues cause the condition number to spike and thus cause the numerical method to perform poorly.](image-url)
It is apparent from Figure 6.8 that all eigenvalues of the Helmholtz equation can be identified and cause the method to be singular. We see that at each eigenvalue $\beta_{m,n}$, the condition number of the linear system spikes by at least five orders of magnitude. The linear system being independent of the boundary conditions imposed on the geometry, we know that this result holds generally for the Helmholtz equation on the unit square. These results make sense in view of the discussion of §3.2.2. We can expect similar results for different geometries such as a trapezoid. We shall conduct the same study for the trapezoidal shape of Figure 4.11 in §6.2. Closing this section we do note the method is very sensitive to the reach $R$ of the Halton distribution (or any other distribution for that matter) for varying $\beta$. We do note however that given the Helmholtz equation and a particular geometry, one could explore this sensitivity on $R$ by looking at the condition number of the linear system independently of the boundary conditions imposed on the system. Upon finding satisfactory regimes for $R$ in terms of $\beta$, the Dirichlet to Neumann map can be expected to perform better than for arbitrary values of $R$.

6.2 Trapezoid

In view of the previous section, §6.1, we will skip initial tests which were based on monitoring the error as $\beta$ increased. We shall look at the condition number of the linear system as functions of $R$ and $\beta$ to see if similar situations arise. We start with the Helmholtz equation (6.1) but on a trapezoidal domain this time. The domain in question with the boundary conditions that are imposed to find eigenmodes is shown as follows in Figure 6.9.
6.2.1 Eigenmodes and Eigenvalues

Not having explicit expressions for the eigenmodes or the eigenvalues of the Helmholtz equation on this geometry, we seek numerical methods. Following code written by Professor Fornberg, we form a generalized eigenvalue problem by discretizing the problem to second order and solving it using Matlab’s ”eig” routine. The code can be found in Appendix B. There are infinitely many eigenvalues and eigenmodes but we cannot hope to determine all of them. By discretizing the problem to a size $N$, we can recover at least $N$ eigenvalues and eigenmodes with enough accuracy. With increasing $N$, the accuracy is increased but so is the size of the problem. We are only finding eigenvalues and eigenmodes for $0 < \beta < 8$ and with enough accuracy to make sure that any spike in our numerical implementation of the Dirichlet to Neumann map is located at an eigenvalue of the Helmholtz equation and that all eigenvalues cause spikes. Even though we do not need extraordinary accuracy we still would like to know how well the generalized eigensolver performs. This performance test is left to Appendix B. The method is found to nearly of second order. Using
Bengt Fornberg’s code, we find and list the first 31 eigenvalues (organized by magnitude) and the first 12 eigenfunctions in Table 6.2 and Figure 6.10 respectively.

Table 6.2: The first 31 eigenvalues of the Helmholtz Equation on the trapezoidal domain ordered by magnitude. The placement in the table does not reflect wavenumbers but rather magnitude.

<table>
<thead>
<tr>
<th>Eigenvalues Ranked by Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st-7th</td>
</tr>
<tr>
<td>8th-14th</td>
</tr>
<tr>
<td>29th-31th</td>
</tr>
</tbody>
</table>

Figure 6.10: First 12 eigenmodes on the trapezoid with $\beta$ displayed above.

Having numerical approximations to all eigenvalues from zero to eight, we can make sure that if any
spikes in the condition number of the linear system do occur, they occur for $\beta$ being an eigenvalue of the Helmholtz equation. We also want to make sure that all eigenvalues cause the method to be singular.

6.2.2 Singular Values of the Linear System

We now proceed as we did in §6.1.3. That is, we monitor the condition number of the linear system for the trapezoid for varying $R$ and $\beta$. This can be seen in Figures 6.11 and 6.12 (A different angle to see the characteristics of the surface more clearly).

Figure 6.11: Condition number of the linear system as we increase $\beta$. We see the same behavior as in Figure 6.4, namely that there is a regime of $R$ values that minimizes the condition number of the linear system. We however are able to see the spikes in the condition number for all eigenvalues on this plot. We note that most spikes would be unnoticeable if the reach of the method was fixed.
Figure 6.12: Same plot as in Figure 6.11 but from a different angle to more clearly see the characteristics of the surface. We clearly see here the narrow valley in which the condition number is minimized.

From Figures 6.11 and 6.12 we see the same behavior as in Figure 6.7. That is, there is a narrow range of $R$ values that minimize the condition number. In order to increase our chances of finding spikes in the condition number, we parametrize the reach $R$ as a function of $\beta$ again to ensure that for all $\beta$, the reach is in its optimal range. A particular parametrization of $R$ that worked well for our tests:

$$R(\beta) = -2280.2 + 31035.6e^{\beta^{1/4}} - 28626.0e^{\beta^{1/3}} + 3845.0e^{\beta^{1/2}} + 43.1e^{\beta} + 20.1\beta \quad (6.6)$$

With this version of $R(\beta)$, we compute the condition number of the linear system as we vary $\beta$ from near zero to eight. The degree of the Legendre polynomial expansion was set to $M = 25$. We can see the resulting plot in Figure 6.13:
Figure 6.13: Condition number of the linear system as we increase $\beta$ with $R(\beta)$. The vertical lines are placed at all $\beta_{m,n}$ that were computed numerically. We see that all eigenvalues cause the condition number to spike and thus cause the numerical method be singular.

We can clearly see from Figure 6.13 that the condition number of the linear system spikes for various values of $\beta$. The tick marks and vertical lines are placed at all eigenvalues $\beta_{m,n}$ which were computed using the generalized eigensolver. We can see that all eigenvalues produce spikes in condition number, indicating that the method will perform poorly for those values. This result agrees with the results on the unit square. This behavior is to be expected as any particular solution one seeks to find with a set of boundary conditions can be superimposed with infinitely many eigenmodes while still satisfying the PDE and the boundary conditions. From the discussion of §3.2.2 we know that the eigenvalues of the linear system shift towards the origin as $\beta$ increases, causing the determinant to become zero. With zero determinant, the linear system is not invertible and there are infinitely many solutions (particular solutions with superimposed eigenmodes). We
also note from Figure 6.11 that the sensitivity to $R$ is not as pronounced as with the test on the unit square in Figure 6.7.

### 6.3 Testing Eigenmodes in the Dirichlet to Neumann Map

In the previous sections, we noted that for $\beta = \beta_{m,n}$, the linear system became singular and solutions were not meaningful. For the purposes of testing the numerical implementation of the Dirichlet to Neumann map as much as we can, we test it to see what it does when we supply it with data from eigenmodes. That is to say, we have our linear system $Ax = b$ in which we usually supply $b$, the known boundary data, to then solve for $x$, the unknown boundary data. We wish to try here to supply $x$ and see if $Ax$ does indeed return $b$ for the case of eigenfunctions. Returning to the problem on the unit square of Figure 6.1. The eigenfunctions are again:

$$q_{m,n}(x,y) = (-1)^{n+1} \sin (m\pi x) \sin \left( \left( n - \frac{1}{2} \right) \pi y \right)$$  \hspace{1cm} (6.7)

We wish to supply our Dirichlet to Neumann map with boundary data from these eigenmodes, that is, we shall supply $q_{n}^{(1)}$, $q_{n}^{(2)}$, $q_{n}^{(3)}$, and $q_{n}^{(4)}$ as our solution $x$ and see if $Ax$ returns the corresponding zero boundary conditions $b$ as $q^{(1)} = q^{(2)} = q^{(3)} = q^{(4)} = 0$. We derive the "known" functions from (6.7) to be:

$$q_{n}^{(1)}(x) = \frac{2n-1}{2} \pi \cos (n\pi) \sin (m\pi x)$$ \hspace{1cm} (6.8)

$$q_{n}^{(2)}(y) = m\pi \cos (m\pi) \sin \left( \pi \left( m - y \left( m - \frac{1}{2} \right) \right) \right)$$ \hspace{1cm} (6.9)

$$q_{n}^{(3)}(x) = \sin (m\pi x)$$ \hspace{1cm} (6.10)

$$q_{n}^{(4)}(y) = -m\pi \sin \left( \pi \left( m - y \left( m - \frac{1}{2} \right) \right) \right)$$ \hspace{1cm} (6.11)

Wishing to insert these functions in our linear system as $x$, we must expand each side $v = 1, \ldots, 4$ in Legendre polynomials up to order $M$ as we would with unknowns. The expansions are of the form:

$$\sum_{l=0}^{M} c_{l}^{(v)} P_{l}(t)$$ \hspace{1cm} (6.12)
We normally then have \( 4 \times M \) unknowns which are then solved by the method. In this case we wish to supply the Legendre coefficients. To compute them, we use the standard relation:

\[
    c_l^{(v)} = \frac{2l + 1}{2} \int_{-1}^{1} Q^{(v)}(x) P_l(x) \, dx
\]

where \( Q \) can represent either \( q \) or \( q_n \) depending on the side in question. Though the coefficients can be computed with (6.13), we wish to compute them with high accuracy. Numerical cancellations will occur in a direct computation of (6.13) using quadratures, causing the results to not be very accurate. In order to compute the Legendre coefficients to high degrees (\( M = 60 \) in this case) we notice that the boundary functions are all trigonometric functions and can thus be written in terms of exponentials. We can then use the Bessel integral relation introduced in Chapter 4. We rewrite it here:

\[
    \int_{-1}^{1} e^{\alpha t} P_l(t) \, dt = \sqrt{\frac{2\pi a}{\alpha}} I_{l+1/2}(\alpha)
\]

We expand the boundary functions (6.8) - (6.11) in terms of exponential functions and use the Bessel relation (6.14) to compute the coefficients for several wave numbers \( m = 1, 2, ..., 10 \) and \( n = 1, 2, ..., 10 \). We then generate a cloud of \( k \)-values in the complex plane and assemble the pieces: We insert our Legendre coefficients as \( x \) in the linear system, apply matrix \( A \) to \( x \) with appropriate eigenvalue \( \beta_{m,n} \) and see whether or not the computation returns \( b = 0 \), the zero vector. We do this for all \( k \)-values in our \( k \)-cloud and for all wavenumbers discussed above. For each wavenumber, we compute the infinity norm along all \( k \)-values and display that norm. The results are in Table (6.3).
Table 6.3: The max norm over many $k$-values of the result $b$ for the first 100 wavenumbers. Note that all numerical values are multiplied by $10^{-13}$. We see that the linear system returns a good approximation to the zero vector.

<table>
<thead>
<tr>
<th>Wavenumbers</th>
<th>$n = 1$</th>
<th>$n = 2$</th>
<th>$n = 3$</th>
<th>$n = 4$</th>
<th>$n = 5$</th>
<th>$n = 6$</th>
<th>$n = 7$</th>
<th>$n = 8$</th>
<th>$n = 9$</th>
<th>$n = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = 1$</td>
<td>0.0264</td>
<td>0.0241</td>
<td>0.0718</td>
<td>0.0629</td>
<td>0.0430</td>
<td>0.0545</td>
<td>0.1001</td>
<td>0.1299</td>
<td>0.0715</td>
<td>0.1601</td>
</tr>
<tr>
<td>$m = 2$</td>
<td>0.0564</td>
<td>0.0502</td>
<td>0.1129</td>
<td>0.0713</td>
<td>0.1010</td>
<td>0.1295</td>
<td>0.1218</td>
<td>0.2612</td>
<td>0.1945</td>
<td>0.2070</td>
</tr>
<tr>
<td>$m = 3$</td>
<td>0.0456</td>
<td>0.0725</td>
<td>0.1605</td>
<td>0.0866</td>
<td>0.2055</td>
<td>0.1691</td>
<td>0.1907</td>
<td>0.3944</td>
<td>0.2018</td>
<td>0.4446</td>
</tr>
<tr>
<td>$m = 4$</td>
<td>0.0342</td>
<td>0.1240</td>
<td>0.2589</td>
<td>0.2665</td>
<td>0.2164</td>
<td>0.2104</td>
<td>0.2547</td>
<td>0.4241</td>
<td>0.2112</td>
<td>0.4700</td>
</tr>
<tr>
<td>$m = 5$</td>
<td>0.0481</td>
<td>0.2144</td>
<td>0.2004</td>
<td>0.1886</td>
<td>0.2025</td>
<td>0.2104</td>
<td>0.2122</td>
<td>0.5556</td>
<td>0.5831</td>
<td>0.3265</td>
</tr>
<tr>
<td>$m = 6$</td>
<td>0.1258</td>
<td>0.2032</td>
<td>0.2074</td>
<td>0.2589</td>
<td>0.2434</td>
<td>0.2824</td>
<td>2.2785</td>
<td>0.3122</td>
<td>0.2689</td>
<td>0.2522</td>
</tr>
<tr>
<td>$m = 7$</td>
<td>0.1979</td>
<td>0.1944</td>
<td>0.3652</td>
<td>0.2008</td>
<td>0.3276</td>
<td>0.4912</td>
<td>0.4766</td>
<td>0.9037</td>
<td>0.7034</td>
<td>0.4828</td>
</tr>
<tr>
<td>$m = 8$</td>
<td>0.0548</td>
<td>0.0581</td>
<td>0.2731</td>
<td>0.2093</td>
<td>0.1541</td>
<td>0.1612</td>
<td>0.4104</td>
<td>0.7094</td>
<td>0.4138</td>
<td>0.4567</td>
</tr>
<tr>
<td>$m = 9$</td>
<td>0.0925</td>
<td>0.1192</td>
<td>0.3200</td>
<td>0.2221</td>
<td>0.2659</td>
<td>0.1983</td>
<td>0.6675</td>
<td>0.7320</td>
<td>0.8414</td>
<td>0.8237</td>
</tr>
<tr>
<td>$m = 10$</td>
<td>0.0625</td>
<td>0.0934</td>
<td>0.3322</td>
<td>0.2033</td>
<td>0.2666</td>
<td>0.2763</td>
<td>0.2751</td>
<td>0.4837</td>
<td>0.4251</td>
<td>0.8354</td>
</tr>
</tbody>
</table>

From Table (6.3) we see that, when we insert the non-zero eigendata generated from eigenfunctions of the unit square, the method returns the zero vector as it should. That is to say, given boundary data from an eigenfunction with $\beta$ set to be the corresponding eigenvalue, the linear system is able to recover the zero boundary conditions which spanned the eigenfunctions. The reverse is of course not true. We pursued this result in order to feel more confident about the results of the previous sections. This concludes Chapter 6 and our study of eigenmodes and eigenvalues of the Helmholtz equation on various geometries with some results being independent of boundary conditions.
Chapter 7

Corner Correction for the Helmholtz Equation

In this chapter we consider singular expansions for the Helmholtz equation on a trapezoidal domain for use in our numerical implementation of the Dirichlet to Neumann map. The problem of singularities in certain corners for the Laplace equation is well known and numerous studies have dealt with it including [16], [27], [22], [28], [36], [7], and [8]. The case of the Helmholtz equation is studied in [30]. From Grisvard [18], Fokas and Kapaev [15], and the sources aforementioned, we know that it is possible for singularities to exist in the corners of convex polygons. The singularity can be external in that it results from discontinuities in the boundary conditions or their derivatives. The singularity can also be internal in that it depends on the internal angle of a corner and the type of boundary conditions imposed. For internal angles greater than or equal to $\pi/2$, a Neumann condition on one side and a Dirichlet condition on the other will result in a singularity. The problem description and geometry will follow in §7.1. In [16], Fornberg and Flyer investigate the effects of singularities in the corners of a trapezoidal domain for a Dirichlet to Neumann map. The Dirichlet to Neumann map is based on the Fokas boundary integral method but formulated for the Laplace equation. A singular development is made in the corners and used as a corrective term in the Dirichlet to Neumann map. Upon inserting the corrective term, the method is shown to perform much better in the presence of singularities. A similar result is sought here in view of the availability of singular expansions in corners for the Helmholtz equation. Such expansions and more general ones can be readily found using the framework of Martin [30]. Though we present here the framework for introducing corrective terms to the Dirichlet to Neumann map, we note in
§7.3 that the implementation was not successful and is a work in progress.

7.1 Singular Expansion in the Case of a Singular Corner

We start with the Helmholtz equation on trapezoidal domain. In view of expanding the Helmholtz equation in certain corners, we write the Helmholtz equation in cylindrical coordinates in (7.1). The domain, boundary conditions and geometry used are shown in Figure 7.1. The singularity we are trying to treat lies in the 135 degree corner of sides 3 and 4.

\[
\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial q}{\partial r} \right) + \frac{1}{r^2} \frac{\partial q^2}{\partial \theta} + 4 \beta^2 q = 0 \quad (7.1)
\]

Figure 7.1: Trapezoidal domain with boundary conditions. We center our coordinate system at the corner of sides 3 and 4.

We know that there is an internal singularity in the 135 degree corner since the internal angle is greater than \( \pi/2 \) we have specified a Neumann condition on side 3 and a Dirichlet condition on side 4. We can expand the cylindrical Helmholtz equation in that corner in terms of functions
which are singular at that corner and satisfy the Helmholtz equation. The expansion, from Martin [30] is:

\[ Q(r, \theta) = \sum_{j=1}^{\infty} a_j J_{\frac{2}{3}(2j-1)}(2\beta r) \sin \left( \frac{2(2j-1)\theta}{3} \right) \]  

(7.2)

where \( J \) are Bessel functions of the first kind. It can easily be verified that (7.2) does satisfy the Helmholtz equation and the boundary conditions specified. In order to make use of this expansion in our Dirichlet to Neumann map we must do some work. We recall here the global relations (3.13) and (3.14) which are the basis for the Dirichlet to Neumann map. They are re-casted here because we are going to need them once again:

\[ \int_{\partial \Omega} e^{-i\beta(kz+\pi/k)} \left( \frac{dz}{ds} \left| q_n + \beta \left( -\frac{1}{k} \frac{d\pi}{ds} + k \frac{dz}{ds} \right) q \right| ds \right. \]  

(7.3)

and,

\[ \int_{\partial \Omega} e^{i\beta(kz+\pi/k)} \left( \frac{dz}{ds} \left| q_n + \beta \left( -\frac{1}{k} \frac{dz}{ds} + k \frac{d\pi}{ds} \right) q \right| ds \right. \]  

(7.4)

We must insert the singular expansion \( Q \) (7.2) into the global relations in order to use them in our numerical Dirichlet to Neumann map as corrective terms. We will insert the singular expansion on sides 3 and 4, at the 135 degree corner. The relevant equations for sides 3 and 4 are, parameterizing in \( t \) from \((-1, 1)\), the following: **Side 3:**

\[ Q = J_{2/3}(2\beta r) \sin(2\theta/3) = J_{2/3}(\sqrt{2}\beta(1-t)) \]  

(7.5)

\[ Q_n = 0 \]  

(7.6)

The global relation integral on side 3 involves an exponential and a Bessel function of fractional order. We do not know of closed form expressions for these integrals but we could evaluate the integrals using adaptive quadratures. However, using such quadratures returns large numerical errors, most likely due to the highly oscillating integral. Inspired by [16], we seek to use the \( H \) function (7.7) as it is very promising here. In order to do this we expand the Bessel function in a power series and work with the resulting terms to use the \( H \) function continued fraction representation. Details on the continued fraction representation of the \( H \) functions and its numerical
Evaluation to high precision everywhere in the complex plane can be found in Appendix C. With the ability to evaluate the $H$ function with high accuracy in the complex plane, we now seek to use it.

$$H(a, z) = z^{-a} \int_0^z t^{a-1} e^{-t} dt$$  \hspace{1cm} (7.7)

After expanding the Bessel function of (7.2) in a power series and manipulating the equation to form which can be reduced to the $H$ function, we arrive at the following representation of the global relation on Side 3:

$$I_1 = e^{\beta \gamma} \left( k(1 + i) - \frac{1 - i}{k} \right) \sum_{m=0}^{N} \frac{(-1)^{\frac{9m+2}{3}} 2^{m+1/3} i^{\frac{12m+10}{3}}}{m! \Gamma(m + 5/3)} H \left( \frac{6m + 5}{3}, i\beta \left( k(1 + i) + \frac{1 - i}{k} \right) \right)$$  \hspace{1cm} (7.8)

with its Schwartz conjugate

$$I_2 = e^{\beta \gamma} \left( k(1 - i) - \frac{1 + i}{k} \right) \sum_{m=0}^{N} \frac{(-1)^{\frac{9m+2}{3}} 2^{m+1/3} (-i)^{\frac{12m+10}{3}}}{m! \Gamma(m + 5/3)} H \left( \frac{6m + 5}{3}, -i\beta \left( k(1 - i) + \frac{1 + i}{k} \right) \right)$$  \hspace{1cm} (7.9)

with, $\gamma = k - 1/k$, $\epsilon = k + 1/k$, and $N$ is be the cutoff of the power series expansion of the Bessel function. The power series does converge uniformly and very rapidly, allowing us to use small $N$, not that it is needed. Similarly for side 4: **Side 4:**

$$Q = 0$$  \hspace{1cm} (7.10)

$$Q_n = -(2/3)J_{2/3}(\beta(1 + t))$$  \hspace{1cm} (7.11)

In the same way we arrive at the integrals on side 4:

$$J_1 = -\frac{2}{3} e^{\beta \gamma \frac{N}{3}} \sum_{m=0}^{N} \frac{(-1)^{m} \Gamma(\frac{m+5}{3})}{m! \Gamma(m + 5/3)} H \left( \frac{6m + 5}{3}, \beta \gamma \right)$$  \hspace{1cm} (7.12)

with its Schwartz conjugate which happens to be identical:

$$J_2 = J_1$$  \hspace{1cm} (7.13)

The expressions on side 4 are much simpler. With these expressions at hand, in order to perform corrections to our Dirichlet to Neumann maps, we need to numerically evaluate the $H$ function accurately in the complex plane. A brief description of how that is done is left to the next section.
7.2 Singular Terms in the Global Relations

We investigate the possible numerical evaluations of the integral (7.7) in the complex plane. As noted in [16], the $H$ function is an entire function. This function can be evaluated analytically to become (7.14).

$$H(a, z) = z^{-a} (\Gamma(a) - \Gamma(a, z))$$

(7.14)

where $\Gamma(a, z)$ is the upper incomplete Gamma function with complex argument $z$. It is noted that Matlab does not have a built in function for evaluating the upper incomplete Gamma function of complex arguments. We will therefore need to find a way to evaluate either the $H$ function directly or the upper incomplete Gamma function in the complex plane. As was noted in [16], the $H$ function has a continued fraction representation that is guaranteed to converge everywhere in the complex plane. The continued fraction is the following:

$$H(a, z) = e^{-z} \frac{1}{a \frac{a-1}{a+1 \frac{a-2}{a+2 \frac{a-3}{a+3 \frac{a-4}{a+4 \frac{a-5}{a+5 \frac{a-6}{a+6 \ldots}}}}}}}$$

(7.15)

A first direct numerical implementation of the continued fraction demonstrates its usefulness and convergence but only works in a disk of limited radius because of numerical cancellation problems. If we are to use the $H$ function far out in the complex plane as is needed for the corner correction of the Helmholtz equation, we need to implement the continued fraction differently. We test three ways of calculating the $H$ function in Appendix C. Details can be found there but the conclusion is that we are able to calculate the $H$ function with high accuracy everywhere in the complex plane via a combination of methods.

7.3 Implementation

With the framework of this Chapter, it should be possible to numerically implement corner corrections to the Dirichlet to Neumann map. The corner correction is implemented very similarly to the work of Fornberg and Flyer [16]. The singular expansion (7.2) is further expanded in a power
series and inserted in the global relations for sides 3 and 4. Being a series of terms involving the \( H \) function, we numerically implement accurate and stable algorithms for computing the \( H \) function. However, we do not have a solution to the Helmholtz equation for the boundary conditions shown in Figure 7.1. For the case of the Laplacian studied by Fornberg and Flyer in [16], the singular expansion itself was used to create a solution on the entire domain by ingenious methods. The same attempt was made here for the Helmholtz equation but the method could not converge past 3 decimals and therefore was not adequate for comparisons. Furthermore, the solution seemed to decay rapidly as the distance from the expansion point increased, making it even harder to have useful solutions on sides 1 and 2. Alternatively, finite differences could be used to find a solution but again there are problems in the 135 degree corner due to the singularity there. We also tried to use a solution calculated using Matlab's PDE Toolbox. However, we are not interested in the solution on the interior of the polygon but rather in the solution and its normal derivatives on the periphery of the domain. We then need extrapolate the solution to the periphery of the trapezoid for sides 1 and 3, and for sides 2 and 4, we need to compute derivatives with one sided stencils. All in all we do not feel like we have an adequately accurate solution on \( \partial \Omega \) to be useful in comparison.

In order to generate a solution against which we can compare, we tackle a different problem to which we do have a solution. We take the singular expansion (7.2) and use it as our solution. Out of that solution we gather known boundary conditions on all sides and perform the numerical Dirichlet to Neumann map. We note that the only change from the previous framework is that now \( q_n^{(1)} \neq 0 \) and \( q_n^{(2)} \neq 0 \) as well. Those two boundary conditions are computed directly from (7.2). The standard Dirichlet to Neumann map without any corner corrections yields the following results:
Figure 7.2: Dirichlet to Neumann map results without the singular correction. We note that sides 1 and 2 perform well but that sides 3 and 4 have large errors near the corner due to the singularity there. 500 Halton nodes with $M = 20$.

From Figure 7.2 we see that without any corner correction, we get adequate error results on all sides except near the junction of sides 3 and 4, where the singularity lies. This is to be expected as the solution we build is singular at the 135 degree corner. With this in mind and with the formulation of this chapter, we wish to introduce the singular corrective term and see how it modifies our results. We now see the effects of the correction:
Figure 7.3: Dirichlet to Neumann map results with the singular correction. We note that sides 1 and 2 perform better but that sides 3 and 4 have even larger errors near the corner due to the singularity there. All the parameters are the same as in Figure 7.2.

From Figure 7.3 we see some intriguing results. The error improved along sides 1 and 2 by two and one orders of magnitude respectively. However the error near the singularity at the corner joining sides 3 and 4 increased and became more oscillatory. Though we appreciate the improvements on the two far sides, we are troubled by the decrease in performance in the singular corner as the method was constructed to improve the error there. The results of [16] indicate that the improvement should be along all sides, including at the singular corner. Though the work is still a work in progress we wish to report here several issues which may be causing this behavior.

- The expression (7.8) and its Schwartz conjugate are rather complicated. There are terms involving increasing fractional powers of $i$. Upon simplifying, we arrive at multivalued ex-
pressions. Perhaps the way in which we choose particular Riemann surfaces is not adequate in our numerical implementation. We do note that the choice of manifold is somehow dependent on the value of $k$. We generally overdetermine the linear system with at least 500 $k$-values to introduce the corner correction. This is a problem since we do not know how to choose the correct manifold. Perhaps expression (7.8) can be recast into a different form not involving multivalued quantities to avoid this problem.

- It is possible that the singular expansion (7.2) does not converge far enough to be useful in our applications. We note that there could be other singularities in other corners of the trapezoid and that the radius of convergence of our singular expansion could be limited to the shortest distance from the 135 degree corner and other corners, namely a distance of 1. We however need an expansion that converges up to a distance of $\sqrt{2}$ for side 3. We are currently performing tests to see if this is indeed the problem.

- The coefficient associated with the corrective term is very small, on the order of $10^{-6}$. This indicates that upon solving the system, this coefficient is more or less ignored as being meaningful in the solution. This could indicate that something is wrong with the formulation or the equations in use, perhaps agreeing with the first bullet point.

We are confident from Appendix C that the problem does not rely in the numerical evaluation of the $H$ function in the complex plane. We consider the fact that the derivation of equations (7.8), its Schwartz conjugate, equation (7.12) and its Schwartz conjugate contains some errors. Even though the method did not perform as expected, actually worsening accuracy in the 135 degree corner, we wish to present the work done here as an attempt at implementing a corner correction method for the Helmholtz equation. Work to try and discern where errors rely is still in progress.
Chapter 8

Numerical Solution to the Modified Helmholtz and the Helmholtz Equations

This chapter is focused on the numerical evaluation of the full solutions of (2.2) on the interior of a trapezoid. The discussion is very brief the work in this area was itself very brief. As mentioned in Chapter 3 in order to compute the solutions to our PDE (2.2) with the Fokas methodology, we must first be able to get all boundary information via Dirichlet to Neumann maps, or other means. Having explored the numerical implementation of Dirichlet to Neumann maps and shown that it is fast and accurate, we are now in a position compute the solutions for the interior of a trapezoid. We note that the solution to the Helmholtz equation was not implemented in due time and shall only be briefly discussed.

8.1 Solution to the Modified Helmholtz Equation

From Chapter 3 we have integral solutions to the modified Helmholtz equation involving certain spectral integrals which themselves are in terms of both Dirichlet and Neumann boundary conditions for all sides of the polygon. Even though we do have an accurate and fast method of computing Dirichlet to Neumann maps which would allow us to tackle a well posed problem, we wish to test the full solution formulation without introducing errors or unwanted effects from our D2N. We therefore proceed from a problem to which we know the solution. The geometry on which we tested the solution is the now very familiar trapezoid. We take for solution,

\[ q = \Im \left[ \exp \left( z + \beta^2 \bar{z} \right) \right] \quad (8.1) \]
From (8.1) we extract explicit expressions for both the Dirichlet and the Neumann boundary conditions. We use these to explicitly resolve the spectral functions (3.3). We note that in a real problem we would not be able to do this and we would require our D2N to perform this step. However, with these spectral functions at hand, we sample nodes on the interior of the trapezoid at which we desire a solution. We are left with computing 4 integrals per node along the rays $l_j$. The four rays $l_j$ for this geometry are depicted in Figure 8.1:

Figure 8.1: The four rays $l_j$, $j = 1, \ldots, 4$ from the origin to $\infty$ along which we numerically integrate to get our solution.

As mentioned and proven in [14], the integrals decay exponentially and are bounded at the origin. The integrals are then computed using adaptive quadratures in Matlab cutting off the integration at a distance of 225 away from the origin. The computed solution on the trapezoid is shown in Figure 8.2:
Figure 8.2: The solution to the modified Helmholtz equation computed via the BI-F formulation on the trapezoid. We note that the function values grow exponentially.

We see from Figure 8.2 that the solution is very smooth and grows exponentially in the positive $x$ direction. We now display the error of the computed solution with respect to the explicit solution (8.1):
From Figure 8.3 we see that the error over the entire domain is bounded above by $10^{-11}$. We see that cutting off the integration at a value of 225 yielded excellent results. We are exalted at the accuracy of the solution. However, this method does have a drawback. For each node at which we desire a solution, we must compute 4 integrals using adaptive quadratures. We further note that in the case of a real well posed problem, the spectral functions (3.3) would also have to be computed at each node on the interior. This test was very brief but shows that the solution to the interior is very reliable. The case of the Helmholtz equation is more complicated as we shall see in §8.2.

8.2 Solution to the Helmholtz Equation

As with the modified Helmholtz equation we have from Chapter 3 an integral representation of the solution on $\Omega$. However, the paths of integrations described by (3.7) are more complicated, performing a semicircle around the origin before going out to infinity. Because of this complication
and because of the fact that the integrand near the origin is highly oscillatory, we were unable to test the solution to the Helmholtz equation in due time for this thesis. We note that the structure of the solution is the same as that of the modified Helmholtz equation in that for each node $z$ at which we desire a solution, 4 integrals need to be computed along with the computation of the spectral functions (3.6).

We leave this chapter with some concluding remarks regarding the form of the integral solution. In some cases this method could be very beneficial. For instance, local refinement is directly applicable. We could also imagine that one could be interested in the solution in a particular region of a domain, near a corner or some aperture type of structure. In that case, the solution to Helmholtz-type PDEs (as well as the Laplacian) is very convenient as it can be computed in that region alone, ignoring the rest of the domain. This concludes our brief exploration of the full solution to the Helmholtz and the modified Helmholtz equations.
Chapter 9

Conclusions

The main focus of this thesis was to show that fast and accurate Dirichlet to Neumann maps are possible in the framework of the Fokas methodology: it was shown in Chapter 4 that the numerical implementation of the Dirichlet to Neumann map for the Helmholtz equation and the modified Helmholtz equation on various geometries is not numerically demanding and is spectrally accurate. Solving linear systems of minimal sizes, unknown boundary conditions were recovered with great accuracy. It was noticed however that the method suffered from the presence of singularities in the corners of our polygons. We attempted in Chapter 7 to improve results by applying a singular correction term to the method. A framework for such a correction following the work of [16] was introduced and implemented. However it was shown that the implementation contained errors which did not improve the accuracy of the method as we would have expected it to do. We note from Chapter 5 that the most influential parameter on the accuracy of the method was not necessarily the type of distribution used to overdetermine our linear systems but rather the Reach of a particular distribution in the complex plane. Halton nodes were selected because of their superior performance when other parameters such as the degree of the Legendre expansions, $M$, or the Helmholtz constant, $\beta$, were modified. The full solution for the case of the modified Helmholtz equation was shown to be accurate but not very efficient as multiple integrals need to be calculated at each desired solution point. One strong aspect of the formulation of the solution however is that one can evaluate the solution anywhere. That is one could compute the solution over a small region with a dense grid if needed. Local refinement is also directly available. In Chapter 6 we
also explored the effects of the eigenvalues of the Helmholtz equation on the Dirichlet to Neumann map. As was expected, the method diverged for $\beta$ an eigenvalue of the Helmholtz equation. The eigenvalues were shown to negatively impact the method independently of boundary conditions by measuring their effects on the condition number of the linear system.

The results of this thesis and of the work by Fornberg and Flyer [16] suggest that the presented implementation of the Dirichlet to Neumann map is spectrally accurate and computationally inexpensive. As such the numerical implementation is presented as being competitive compared to existing boundary integral methods of the Fokas methodology for the Laplacian, the Helmholtz equation and the modified Helmholtz equation. Myriad other boundary integral methods exist and were not reviewed. No comparisons were made to measure the performance displayed in this thesis compared to other existing methods. Future studies are required to properly place this method where it stands.

9.1 Future Work

- The successful implementation of the singular correction for the Helmholtz equation. The treatment of corner singularities of [16] suggests that successful implementation of a singular corrective term could improve accuracy everywhere, especially near the singular corner.

- Singular corrections for the modified Helmholtz equation. It might be possible to also treat corner singularities of the modified Helmholtz equation.

- $k$-cloud distribution optimization for the modified Helmholtz equation. The genetic algorithm study was only performed for the Laplace equation and the Helmholtz equation. The performance of the Dirichlet to Neumann map for the modified Helmholtz equation suggests that similar results may ensue but we would like to complete the study to make sure.

- Implementation of the solution for the Helmholtz equation. The full solution was only implemented for the case of the modified Helmholtz equation. We would like to implement
it as well for the Helmholtz equation. The implementation of the solution could also be analyzed and possibly optimized. The effects of the eigenvalues of the Helmholtz equation could also be studied.

• Exploring possibilities of creating fast algorithms for computing the full solution to the Helmholtz and the modified Helmholtz equations. As they stand, the integral solutions seem to be somewhat computationally heavy. It is possible that fast algorithms could be developed.

• Some connection with the trigonometric solutions of the Laplace equation, the Helmholtz equation and the modified Helmholtz equation on regular domains such as squares, rectangles, and certain triangles could potentially be found. Perhaps an eigenfunction expansion of the Green functions could be employed and carried through to yield relationships with trigonometric solutions.

• Direct comparison with other types of boundary integral methods which allow for the computation of Dirichlet to Neumann maps for Helmholtz-type PDEs must be pursued. Though the method is both fast and accurate, it cannot be brought forward of any other methods so long as we do not compare it with other existing methods.
Bibliography


Appendix A

The Fokas Method via Green Functions

We firstly note that this appendix contains no original work and is mostly based on E. A. Spence’s PhD Thesis [34]. The appendix is present to give some background as to how the Fokas methodology can be derived. Derivation of the solutions to the Helmholtz equation and the modified Helmholtz equation by means of Green functions. We recall that our PDE is of the form:

\[ \Delta q(x) - \lambda q(x) = 0 \]  \hspace{1cm} (A.1)

Writing (A.1) in its divergence form, integrating and applying Green’s theorem in two dimensions, we know that we can express the solution in terms of Green functions:

\[ q(x) = \int_{\partial \Omega} \left[ -q(\xi) \frac{\partial E}{\partial n}(\xi, x) + q(\xi) \frac{\partial E}{\partial n}(\xi, x) \right] dS \]  \hspace{1cm} (A.2)

where \( E \) is the free space Green Function. With the modified Helmholtz equation, \( E = \frac{1}{2\pi} K_0(\sqrt{\lambda} |\xi - x|) \) while with the Helmholtz equation: \( E = \frac{i}{4} H^{(1)}_0(\sqrt{-\lambda} |\xi - x|) \). The Green functions themselves must satisfy:

\[ \left( \Delta \xi - \lambda \right) E(\xi, x) = -\delta(\xi - x) \]  \hspace{1cm} (A.3)

We now take the Fourier transform, remembering that \( \delta(\xi - x) \) is the Fourier Transform of 1 with parameter \( \xi - x \)

\[ \delta(\xi - x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ik_1(\xi - x)} dk_1 \]  \hspace{1cm} (A.4)

Upon reorganizing the transformed derivatives which are simply factors in Fourier space, we arrive at:

\[ E(\xi, \eta, x, y) = \frac{1}{(2\pi)^2} \int_{R^2} \int e^{ik_1(x - \xi) + ik_2(y - \eta)} \frac{1}{k_1^2 + k_2^2 - \lambda} dk_1 dk_2 \]  \hspace{1cm} (A.5)
Depending on the sign of $\lambda$ the poles of the integrand will be in different places. Because of this, the contour of integration must be deformed to avoid the poles. The deformations will depend on the sign of $\lambda$. We shall treat the case of the modified Helmholtz equation as it is simpler and more useful in terms of understanding the derivation. The case of the Helmholtz equation is more complicated because of a pole located on the path of integration and is treated in detail in E. A. Spence’s Thesis [34].

For the modified Helmholtz equation, we choose $\lambda = 4\beta^2 > 0$. Via certain transformations and rotations in the complex plane, one of the exponential terms is shown to decay in the upper half plane, allowing the evaluation of one of the integrals of (A.5) to be done by closing the contour in the upper half plane. With more algebra and variable changes, the double integral then becomes:

$$E(z', z) = \frac{1}{4\pi} \int_0^\infty \frac{dk}{k} \exp\left( \frac{i\beta}{k} (k(z - z')e^{-i\theta} - \frac{1}{k} z - z'e^{i\theta}) \right)$$

where $z' = \xi + i\eta$ and $z = x + iy$. We now have a convenient representation for our Green function. However, we are not done. We make use of our PDE and the PDE satisfied by the Green functions:

$$q_{\xi\xi} + q_{\eta\eta} - \lambda q = 0 \quad (A.7)$$

and,

$$E_{\xi\xi} + E_{\eta\eta} - \lambda E = -\delta(\xi - x)\delta(\eta - y) \quad (A.8)$$

Combining the two in divergence form:

$$(E_{\xi} q_{\xi} - q_{\xi} E_{\xi}) - (q_{\eta} E_{\eta} - E_{\eta} q_{\eta}) = q \delta \quad (A.9)$$

We now apply Green’s theorem in two dimensions to get our result:

$$q = \int_{\partial \Omega} E \left( q_{\xi} d\eta - q_{\eta} d\xi \right) - q \left( E_{\xi} d\eta - E_{\eta} d\xi \right)$$

$$\quad (A.10)$$

Inserting our representation to the Green function for the modified Helmholtz equation finally yields the solution to the modified Helmholtz equation on the interior of convex polygons. Introducing the geometry of our domain as done in §3.1, we can separate the peripheral integral into a sum of
$n$ integrals, one for each side, while the paths of integration can be seen to depend on the angles $\alpha_j$. Several substitutions and simplifications yield:

$$q(z, \bar{z}) = \frac{1}{4\pi i} \sum_{j=1}^{n} \int_{l_j} \exp \left( i\beta \left( k z - \frac{\bar{z}}{k} \right) \right) \hat{q}_j(k) \frac{dk}{k} \quad (A.11)$$

where $k \in \mathbb{C}$. The spectral functions $\hat{q}_j(k)$, $j = 1, \ldots, n$ are defined as

$$\hat{q}_j(k) = \int_{z_j}^{\bar{z}_j+1} \exp \left( -i\beta \left( k z - \frac{\bar{z}}{k} \right) \right) \left[ q_n + \beta \left( \frac{1}{k} \frac{d\bar{z}}{ds} + k \frac{dz}{ds} \right) q \right] ds \quad (A.12)$$

and the rays $l_j$, $j = 1, \ldots, n$, oriented from 0 to $\infty$ are defined by

$$l_j = \{ k \in \mathbb{C} : \arg k = -\alpha_j \} \quad (A.13)$$

The solution for the Helmholtz equation is sought in the same way but there are complications [34]. Nevertheless, the solution is shown in [34] to be:

$$q(z, \bar{z}) = \frac{1}{4\pi i} \sum_{j=1}^{n} \int_{L_{outj}} \exp \left( i\beta \left( k z + \frac{\bar{z}}{k} \right) \right) \hat{q}_j(k) \frac{dk}{k} \quad (A.14)$$

where $k \in \mathbb{C}$. The spectral functions $\hat{q}_j(k)$, $j = 1, \ldots, n$ are defined as

$$\hat{q}_j(k) = \int_{z_j}^{\bar{z}_j+1} \exp \left( -i\beta \left( k z - \frac{\bar{z}}{k} \right) \right) \left[ q_n + \beta \left( \frac{1}{k} \frac{d\bar{z}}{ds} + k \frac{dz}{ds} \right) q \right] ds \quad (A.15)$$

and the rays $L_{outj}$, $j = 1, \ldots, n$, oriented from 0 to $\infty$ are defined by

$$L_{outj} = \{ k \in \mathbb{C} : k = le^{-i\alpha_j} \} \quad (A.16)$$

and $l$ is the contour depicted in Figure A.1. We are essentially rotating the contour A.1 by $\alpha_j$ clockwise. These contours are not as simple as (A.13). This is due to the aforementioned complications which arise because poles are present on the path of integration. Care must be taken as to how to deform the contours around the poles. Appropriate deformations are those specified by (A.16).
Figure A.1: Part of the contour of integration in the $k$ plane for the Helmholtz equation

We note here that the derivations of [34] are more general and more involved, treating the cases of interior and exterior solutions of the inhomogeneous version of the Laplace, Helmholtz, and modified Helmholtz equations. We merely make use of the homogeneous solution to the Helmholtz equation and the modified Helmholtz equation.
Appendix B

Eigen-problem of the Helmholtz equation on the Trapezoid

This appendix is focused on the code by Professor Fornberg which finds the eigenvalues and the eigenfunctions of the Helmholtz equation on the trapezoidal domain of §6.2 shown in Figure 6.9. The code is shown here.

```matlab
% FD2 code for Trapezoidal domain eigenvalue problem
% Written by Bengt Fornberg
% Modified by Christopher Davis to export high resolution images
function g=FD2_Trap_eig(n)

n21 = 2*n-1; nn21 = n*n21; h = 1/(n-1);
A = zeros(n21,n,n21,n);
B = zeros(n21,n,n21,n);
A(:,1,:,1) = eye(n21); % First column
A(n-1,1,n,2) = -1;
for j = 2:n-1 % Columns 2,...,n-1
    for i = 1:n-j-1
        A(i,j,i,j) = 1;
    end
    A(n-j,j,n-j ,j ) = 1; % Normal derivative
    A(n-j,j,n-j+1,j+1) = -1; % BC on sloping top edge
    for i = n-j+1:n21-1
        A(i,j,i-1,j ) = 1/h^2;
        A(i,j,i+1,j ) = 1/h^2;
        A(i,j,i ,j-1) = 1/h^2;
        A(i,j,i ,j+1) = 1/h^2;
        A(i,j,i ,j ) = -4/h^2;
        B(i,j,i ,j ) = 1;
    end
    A(n21,j,n21,j) = 1;
end
A(:,n,:,n) = eye(n21); % Last column
AR = reshape(A,nn21,nn21);
BR = reshape(B,nn21,nn21);
[evect,eval] = eig(AR,BR); % Solve generalized ev problem
```
% Print and display lowest 12 modes
[ev,ind] = sort(abs(diag(eval))); be = 0.5*sqrt(ev)';
[x,y] = meshgrid(linspace(0,1,n),linspace(0,2,n21));
figure1 = figure;
screen_size = get(0,'ScreenSize');
addpath 'D:\Work\Research\Bengt Fornberg\Code\ExportFigures'
for k = 1:12
    subplot(3,4,k)
    P = flipud(reshape(evect(:,ind(k)),n21,n));
    for i = n+1:n21
        for j = 1:i-n
            P(i,j) = NaN;
        end
    end
    mesh(x,y,P); hold on
    title(['\beta = ',num2str(be(k))]); colormap([0 0 0]);
    xlabel('x'); ylabel('y'); view([-60,70]); axis equal
    plot3(x(1,:)',1+x(1,:)',diag(P,-n+1),'k-')
end
set(figure1,'Position', [0 0 1600 1200 ]);
set(figure1,'Color', [1 0.9 1]);
% export_fig('Plots\eigenmodesHighRes', sprintf('-r200'))

We wish to analyze the performance of this code for increasing $n$. Assuming that the algorithm converges, we run a test with $n = 100$. We shall be testing for $n$ values smaller than 100. We assume that the run with $n = 100$ is a perfect result in that it approximates the real eigenvalues to at least 16 digits. Upon making runs with smaller values of $n$, we compare the average of the first 31 eigenvalues to the average computed at $n = 100$. Each eigenvalue was independently tested and results show that each eigenvalue performed the same as the average of all the eigenvalues. In this way we avoid the need to show 31 different plots. We now plot the error at successive $n$ values up to $n = 90$: 
Figure B.1: log-log plot of performance of the the algorithm code against itself. The linear trend suggests that the method is of second order.

From Figure B.1 we can see that the second order finite difference eigensolver on the trapezoid seems to perform a little worst than second order. This could be due to the fact that we are comparing the method to itself.
Appendix C

H Function Implementation

This appendix is focused on the fast, accurate and stable calculation of the \( H \) function (C.1) for the purpose of implementing corrective methods that deal with singularities in the corner of the trapezoidal domain of Chapter 7 shown in Figure 7.1:

\[
H(a, z) = z^{-a} \int_{0}^{z} t^{a-1}e^{-t}dt
\]  
(C.1)

This function can be evaluated analytically to become:

\[
H(a, z) = z^{-a}(\Gamma(a) - \Gamma(a, z))
\]  
(C.2)

where \( \Gamma(a, z) \) is the upper incomplete Gamma function with complex argument \( z \). As was noted in §7.2, Matlab does not have a built in function for evaluating the upper incomplete Gamma function of complex arguments. We shall investigate several methods of calculation. The first two methods rely on the evaluation of the continued fraction representation of the \( H \) function here shown:

\[
H(a, z) = e^{-z} \frac{1}{a - \frac{az}{a+1+ \frac{z}{a+2 - \frac{z}{a+3+ \frac{z}{a+4- \frac{z}{a+5+ \frac{z}{a+6- \ldots}}}}}}}
\]  
(C.3)

The third method relies on equation (C.2). The Gamma function with argument \( a \in \mathbb{R} \) can be evaluated easily with Matlab. The only problem would be to compute the upper incomplete Gamma function \( \Gamma(a, z) \) with complex argument \( z \). We are fortunate enough that the upper incomplete Gamma function has two continued fraction representations of its own. Having tested both we
write here the one which worked best for us:

\[ z^{-a} \Gamma(a, z) = e^{-z} \frac{1}{1 + z - a + \frac{a-1}{3 + z - a + \frac{2(a-2)}{5 + z - a + \frac{3(a-3)}{7 + z - a + \frac{4(a-4)}{9 + z - a + \frac{5(a-5)}{11 + z - a + \cdots}}}}} \]  

We note here that even thought the \( H \) function is entire, the upper incomplete gamma function is not. It needs a branch cut discontinuity in the complex plane from the origin out to infinity. By convention, the branch cut is placed from the origin to \(-\infty\). With this in hand, we could compute the \( H \) function by combining the results equation (C.2) with (C.4). In all three cases we need to evaluate a continued fraction numerically. We could "directly" compute the continued fractions using the method first introduced by Wallis [37] but there are very large and very small quantities to be handled as the fractional depth increases. To avoid overflow or underflow, we could rescale partial fractions at any point in the descent. However, as stated in [32], there are two newer algorithms which avoid dealing with these very large or very small quantities. We shall not explain each algorithm in detail but we shall mention some of their limitations and advantages. The algorithms are: Steeds algorithm and a modified version of Lentz’s algorithm. Both algorithms can be found in many sources including [3], [32], [35], and [2]. Steeds algorithm will only work if the we know a priori that no denominators vanish. We know this to be true for the continued fraction representation of (C.1). The modification of Lentz’s algorithm involves introducing a shift of very small magnitude when a near zero problematic quantity is detected. The modification was introduced by Thompson and Barnett in [35]. We note with the knowledge of our test results and from general empirical tests [32] that Lentz’s algorithm performs better than Steed’s algorithm. We shall test both Steed’s and Lentz’s algorithms on (C.1), and Lentz’s algorithm on (C.4). After applying Lentz’s algorithm on (C.4) we do compute (C.2) to arrive at the \( H \) function again. To initiate our tests, we look at the number of fractions necessary to arrive at a desired tolerance throughout the complex plane. The tolerance measured was a measure of the contribution from the most recent partial fraction to the entire result. These computational depth results are depicted in Figures C.1-C.3
Figure C.1: Computational depth of Steed’s method applied to the $H$ function C.1 up to a reach of 500 in the complex plane. 1000 Halton nodes

We can see that the computational depth for Steed’s method applied to the $H$ function grows rapidly up and down the imaginary axis as well as in the right half plane, up to almost 600 partial fractions necessary to meet our desired tolerance. Near the origin the computational depth is minimal. Going farther out in the left half plane seems to not be too troublesome but we still reach about 100 recursions. We now test Lentz’s algorithm for the same continued fraction:
Figure C.2: Computational depth of Lentz’s method applied to the $H$ function C.1 up to a reach of 500 in the complex plane. 1000 Halton nodes.

For Lentz’s method, the results are very similar than with Steed’s method. The computational depth increases quite a lot on the imaginary axis and a significant amount in the right half plane while the increase in left half plane is not as significant. The differences between Lentz’s and Steed’s algorithm here are almost indistinguishable graphically. The only actual difference is near the origin, where Lentz’s algorithm perform slightly better. We next test Lentz’s algorithm on the upper incomplete Gamma function C.4:
The results for Lentz’s algorithm applied to the upper incomplete gamma function are very interesting. Any points lying too close to the branch cut were removed (about 10 points). We note the dramatic change in recursion depth near the origin. However, the worst the method does is about 40 recursions. Away from the origin the method only reaches 7-10 recursions, much better than the two previous tests. This is not something we expected as the upper incomplete Gamma function is multivalued with a branch point at the origin and a branch cut, whereas the $H$ function is entire. Perhaps continued fractions can more easily represent functions which are singular by nature.

Having noticed these differences, we seek to test the accuracy of these methods in the complex plane. In order to compare our results to something we shall make use of Matlab’s variable precision arithmetic capabilities, or vpa. We are essentially going to compare the three methods against their vpa counterparts. We are confident in doing this for several reasons. First of all, the continued fraction representations we use here are guarantied to converge in the complex plane.
away from singularities as they are special cases of Gauss's continued fraction representation of $0_{1}F_{1}$ and $1_{1}F_{1}$, hyper geometric functions. Secondly, even with high numbers of digits in use with vpa, the algorithms always converge. We proceed in implementing our three methods using Matlab’s vpa.

Though all three methods were implemented to use vpa, to save computational time in our tests, we use the vpa version of Lentz’s algorithm applied to the upper incomplete Gamma function (C.4). We again remove points near the branch cut. To test the actual accuracy of the three methods, we first sample 2000 nodes uniformly in the complex plane. At each node we compute the three methods and compare each to the vpa calculation. In our comparison we assume that the vpa calculation (done with 300 digits) is accurate to at least 17 digits. We then count the number of digits that match for each method compared to the vpa calculation digits. We count the digits retained for both the real and the imaginary part of each result. All of this is displayed in Figures C.4-C.9

Figure C.4: Digits retained for the real part of Steed’s method applied to the $H$ function up to a radius of 700. 2000 Uniform nodes. We can see excellent performance for the left half plane but very poor performance for the right half plane. (The jump in accuracy is not exactly on the imaginary axis but rather at about $z = 30$.):
Figure C.5: Digits retained for the imaginary part of Steed’s method applied to the $H$ function up to a radius of 700. 2000 Uniform nodes. As with the real part in Figure C.4, we see excellent performance for the left half plane and poor performance for the right half. We also note the points on the real axis (with no imaginary component) do well in the right plane.

Figure C.6: Digits retained for the real part of Lentz’s method applied to the $H$ function up to a radius of 700. 2000 Uniform nodes. We can see excellent performance for the left half plane but very poor performance for the right half plane.
Figure C.7: Digits retained for the imaginary part of Lentz’s method applied to the $H$ function up to a radius of 700. 2000 Uniform nodes. As with the real part in Figure C.6, we see excellent performance for the left half plane and poor performance for the right half. We also note the points on the real axis (with no imaginary component) do well in the right plane.

Figure C.8: Digits retained for the real part of Lentz’s method applied to the upper incomplete Gamma function up to a radius of 700. About 2000 Uniform nodes. We see that the method performs best on the right half plane as opposed to the left half plane (Figures C.4 - C.7). The left half plane is not as accurate but we still retain at worst 12 digits.
From Figures C.4-C.9 we can see that Steed’s method and Lentz’s method applied to the continued fraction of the $H$ function (C.3) perform very well left of the line $z = 30$, retaining 14 to 16 digits of accuracy. Those two tests reveal however that to the right of $z = 30$, we get almost no accuracy. The accuracy of the imaginary part of the $H$ function is excellent smack on the real axis. This can be understood from the fact that the $H$ function is purely real on the real axis and that the computations are stable when handling small quantities. On the other hand, Lentz’s method applied to the upper incomplete Gamma function retains 13 to 16 digits throughout the entire complex plane, away from the branch cut. We do note that the accuracy suffers somewhat on the left half plane. In view of our results here, in order to approximate the $H$ function accurately and quickly throughout the complex plane we segment the complex plane into two parts. For $z$ points in the left half plane, we use Lentz’s method applied to the continued fraction representation of the $H$ function (C.3). For the points on the right half plane, we use Lentz’s method applied to the continued fraction representation of the upper incomplete Gamma function (C.4) and then
reconstruct the $H$ function using (C.2). Though we could use the upper incomplete Gamma function in the left plane away from the branch cut for quicker calculations, we would like to preserve the most accuracy possible.
Appendix D

Complex Laplacian

This appendix is focused on the derivation of the complex form of our second order partial differential equation (2.2) from its real counterpart (2.1). The center of the derivation lies around the chain rule and \( \text{d} \bar{\text{b}} \) derivatives. This was originally created to try and understand where the notation used in Athanassios Fokas' book [14] came from. When expressing the Laplace equation, the following notation is often used for convenience:

\[
\tilde{f}_{zz} = 0 \quad (D.1)
\]

The function \( f \) is differentiated with respect to \( z \) and then with respect to \( \bar{z} \). Any function \( f(x, y) \) can be written equivalently as \( \tilde{f}(z, \bar{z}) \), with the conventional complex variable \( z = x + iy \). We use this notation so as to not have to specify whether the function is in the \((x, y)\) variables or in the \((z, \bar{z})\) variables. We shall use the convention that we start with \( f(x, y) \) and that it can be written in terms of the variables \( z \) and \( \bar{z} \) as \( \tilde{f}(z, \bar{z}) \). We are looking for a relationship between the Laplacian of \( f \), \( \Delta f(x, y) \), and its complex counterpart \( \tilde{f}_{zz}(z, \bar{z}) \). We know that,

\[
df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy = \frac{\partial f}{\partial z} dz + \frac{\partial f}{\partial \bar{z}} d\bar{z} \quad (D.2)
\]

By the chain rule we also know that :

\[
\frac{\partial \tilde{f}}{\partial z} = \left( \frac{\partial f}{\partial x} \frac{\partial x}{\partial z} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial z} \right) \quad (D.3)
\]

and,

\[
\frac{\partial \tilde{f}}{\partial \bar{z}} = \left( \frac{\partial f}{\partial x} \frac{\partial x}{\partial \bar{z}} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \bar{z}} \right) \quad (D.4)
\]
We perform the substitutions $x = (z + \overline{z})/2$ and $y = (z - \overline{z})/2i$ and compute the partial derivatives of both $x$ and $y$. Upon calculating and simplifying, we arrive at:

\[
\frac{\partial \tilde{f}}{\partial z} = \frac{1}{2} \left( \frac{\partial f}{\partial x} + \frac{1}{i} \frac{\partial f}{\partial y} \right) \quad \text{(D.5)}
\]

and

\[
\frac{\partial \tilde{f}}{\partial \overline{z}} = \frac{1}{2} \left( \frac{\partial f}{\partial x} - \frac{1}{i} \frac{\partial f}{\partial y} \right) \quad \text{(D.6)}
\]

Assuming that $f$ and $\tilde{f}$, their partial derivatives and their mixed and non-mixed second order partial derivatives all exist and are continuous, we can calculate the complex Laplacian $\tilde{f}_{zz}$:

\[
\frac{\partial^2 \tilde{f}}{\partial z \partial \overline{z}} = \frac{\partial}{\partial \overline{z}} \left[ \frac{1}{2} \left( \frac{\partial f}{\partial x} + \frac{1}{i} \frac{\partial f}{\partial y} \right) \right] = \frac{1}{4} \left( f_{xx} - if_{yx} + if_{xy} + f_{yy} \right) = \frac{\Delta f}{4} \quad \text{(D.7)}
\]

So we can see that

\[
\tilde{f}_{zz} = \frac{\Delta f}{4} \quad \text{(D.8)}
\]

The factor of one fourth is the reason that the Helmholtz constant and the modified Helmholtz constant throughout most of this thesis are $4 \beta^2$ and not just $\beta^2$. 
Appendix E

Minimal Polynomials

In this Appendix we explore the staircase behavior of the convergence of the D2N map for the modified Helmholtz equation on the trapezoid. It was noted in §4.3 that Figure 4.15 exhibited strange behavior. We remind ourselves that $\beta = 2$ in this case. As a reminder we recast the plot here.

Figure E.1: Maximum of the errors along all four sides of the trapezoid plotted against the degree of the Legendre expansion along each side. Note that this is a log-linear plot.
We see that the trend indicates spectral convergence but we are going to investigate the staircase, or jagged behavior. The error measured is the maximum error along all four sides of the polygon. We note that from an odd degree to an even one we gain more that one whole order of magnitude but then from an even degree to an odd one the method stagnates or even looses accuracy. However, the trend seems to get better in higher degree regimes. In order to investigate in more detail we shall look at all four sides separately. Looking at each side separately, we can see from Figure E.2 that side two is the most problematic side in terms of accuracy for a given degree $M$. This is true for all values of $M$. We can conclude that Figure E.1 is actually the performance of the method on side 2. We now try to explain why that is.

**Figure E.2**: Maximum error along each side plotted against the degree $M$ or the Legendre expansion.

**Minimal Polynomials** Since we are representing our unknown functions in polynomial expan-
sions, namely Legendre polynomials, we wish to investigate how well polynomials can approximate all four unknown functions for various degrees. To do this we shall approximate the unknowns on each side of the polygon with polynomials while minimizing the infinity norm. In other words we are trying to find a close approximation to the minimax polynomial that most closely represents each unknowns in the infinity norm. We are using Matlab’s fminimax to find a near minimal polynomial approximation to our functions on each side. Laying out nodes along each sides uniformly, we find near minimax approximations of various degrees to our unknowns. The errors associated with the approximations exhibit the wonderful properties that minimal polynomial have. Namely we are pleased to see that the oscillation theorem of Atkinson [1] is satisfied and that the error is well bounded as one would hope from de la Vallée-Poussin theorem. Such properties can be seen in Figure E.3 in which we approximate all unknown functions with polynomials of degree 9.

Figure E.3: Ninth degree polynomial approximation for all four boundary functions for the modified Helmholtz equation. We appreciate the presence of 11 extrema in the errors and the fact that the extrema are near uniformly bounded.
The properties mentioned above appeal to us as we can safely assume that the polynomial approximations yielded by the fminimax routine are near minimax approximations to our unknowns. That is to say, these approximation as very close to the absolute best in terms of polynomial approximations in the infinity norm. We also note that side 2 is also the most difficult to approximate. What we are really interested in is the behavior of the error as we increase the degree of the polynomial approximations to see if it is similar to what we observe with our Dirichlet to Neumann map. We therefore repeat this process but for various degrees and look at the largest error along each side separately. We can see from Figure E.4 that side 2 is consistently the hardest to approximate.

![Figure E.4: Maximum error along each side separately for varying degree. We note that side 2 is the hardest to approximate and that the jagged behavior is also present here. All plots are on the same scale.](image)

We see that side 2 exhibits the jagged behavior that we observed in our Dirichlet to Neumann map. For one reason or another, side two is just harder to approximate with polynomials than the
other three sides. We note that the staircase behavior matches that of the Dirichlet to Neumann map performance even though the minimax approximation is almost half an order better. With this examination we conclude that the staircase behavior is not due to intrinsic properties of the boundary integral method but rather due to the fact that we expanded our unknowns with Legendre polynomials. A Chebyshev expansion would have been preferable since Chebyshev approximations are near minimax approximations themselves but there are no known integral relationships for integrating a Chebyshev polynomial and an exponential together which would allow us to approach the Dirichlet to Neumann map in the way we do.
Appendix F

Sample Code: Helmholtz Dirichlet to Neumann map for the Unit Square

In this appendix we show a sample code for the numerical implementation of the Dirichlet to Neumann map for the Helmholtz equation on the unit square. The Matlab code is as follows:

```matlab
function squareHelmholtzTrivial(order,distributionName,numberOfNodes)
% Dirichlet to Neumann map for the Helmholtz equation on the Unit Square
% Code based on original implementation by Bengt Fornberg for the Laplacian
% and modified for use with the Helmholtz equation on the unit square
% INPUTS:
% order is the degree of the Legendre expansion
% distributionName is the type of distribution to be used for the k-cloud
% numberOfNodes is the number of k-values to be used in the k-cloud
% OUTPUTS:
% Computed unknowns and errors relative to known explicit solution

B = 1; % Helmholtz constant
M = order; % Use M+1 terms in the Legendre expansion
% Generate the k-cloud
k = kdistri(R,distributionName,numberOfNodes); K = length(k);

% Define a couple of constants
ro = sqrt(pi^2-4*B^2); C=1;
gamma = k-1./k; ep = k+1./k; su = ((1./(2*k))-k/2);

% Define functions (rhs of linear system composed of known boundary
% functions integrated from the global relations)
f = @(k) C*B^4*pi*su.*exp(-1i*B*ep/2).*cos(B*ep/2)./(pi^2-(B*ep).^2);
g = @(k) C*B^4*pi*su.*exp(1i*B*ep/2).*cos(B*ep/2)./(pi^2-(B*ep).^2);
I = @(a,n) sqrt(2*pi.*a)./a.*besseli(n+1/2,a);

% Form the over-determined linear system
A = zeros(2*K,4*(M+1));
for m = 0:M
    % Calculate the right-hand side of the linear system
    r = f(k(m));
    % Compute the adjacent term
    g = g(k(m));
    % Integrate the Bessel relation
    I_term = I(k(m),n);
    % Assemble the right-hand side
    r_right = r + g.*I_term;
    % Assemble the linear system
    A(:,m+1) = r_right;
end
```

This code implements the Dirichlet to Neumann map for the Helmholtz equation on the unit square, using a Legendre expansion and a specific distribution for the k-values.
A( 1:K , (m+1)) = 0.5*exp(-1i*B*ep/2).*I(-1i*B*ep/2,m);
A( 1:K , (M+1)+(m+1)) = 0.5*exp(B*(gamma/2-1i*ep)).*I(B*gamma/2,m);
A( 1:K ,2*(M+1)+(m+1)) = B*su.*exp(B*(gamma+1i*ep/2)).*I(-1i*B*ep/2,m);
A( K+1:2*K, (m+1)) = 0.5*exp(1i*B*ep/2).*I(1i*B*ep/2,m);
A(K+1:2*K, (M+1)+(m+1)) = 0.5*exp(B*(gamma/2+1i*ep)).*I(B*gamma/2,m);
A(K+1:2*K,2*(M+1)+(m+1)) = B*su.*exp(B*(gamma+1i*ep/2)).*I(-1i*B*ep/2,m);
A(K+1:2*K,3*(M+1)+(m+1)) = 0.5*exp(B*gamma/2).*I(-B*gamma/2,m);
end
%
% Form the right hand side
b = [f(k) ; g(k)];
%
% Normalize least squares side system
Am = sum(abs(A),2); A = A./repmat(Am,1,4*(M+1)); b = b./Am;
c = A; % Solve the over-determined linear system
max(abs(imag(c))); % Print max imaginary part - a measure of the error
%
% Get rid of (tiny) imaginary parts in the c-coefficients

pp = 101; % Prepare result for graphical display
s = linspace(-1,1,pp);
y1 = zeros(1,pp); y2 = y1; y3 = y1; y4 = y1;
%
% Contract unknowns from computed Legendre coefficients
for m = 0:M
    leg = legendre(m,s);
    y1 = y1 + c( (m+1)) * leg(1,:); % Obtain computed function qn_1(x)
    y2 = y2 + c( (M+1)+(m+1)) * leg(1,:); % Obtain computed function qn_2(x)
    y3 = y3 + c(2*(M+1)+(m+1)) * leg(1,:); % Obtain computed function q_3(x)
    y4 = y4 + c(3*(M+1)+(m+1)) * leg(1,:); % Obtain computed function qn_4(x)
end
%
% Take care of parametrization direction
y3 = flipdim(y3,2);
y4 = flipdim(y4,2);
%
% Plot "Approximations"
subplot(2,4,1);
plot(t,C*ro*sin(pi*t)*tanh(ro),'k:',t,y1,'k--');
title('Approximation to qn_1(s)'); xlabel('s');
legend('Analytic','least square')
subplot(2,4,2);
plot(t,-C*pi*cosh(ro*(t-1))*sech(ro),'k:',t,y2,'k--');
title('Approximation to qn_2(s)'); xlabel('s');
legend('Analytic','least square')
subplot(2,4,3);
plot(t,C*sin(pi*t)*sech(ro),'k:',t,y3,'k--');
title('Approximation to q_3(s)'); xlabel('s');
legend('Analytic','least square')
subplot(2,4,4);
plot(t,-C*pi*cosh(ro*(t-1))*sech(ro),'k:',t,y4,'k--');
title('Approximation to qn_4(s)'); xlabel('s');
legend('Analytic','least square')

% Plot "Errors"
subplot(2,4,5);
plot(s,C*ro*sin(pi*t)*tanh(ro)-y1,'k'); title('Error for qn_1(s)'); xlabel('s');
subplot(2,4,6);
plot(s,-C*pi*cosh(ro*(t-1))*sech(ro)-y2,'k'); title('Error for qn_2(s)'); xlabel('s');
subplot(2,4,7);
plot(s,C*sin(pi*t)*sech(ro)-y3,'k'); title('Error for q_3(s)'); xlabel('s');
subplot(2,4,8);
plot(s,-C*pi*cosh(ro*(t-1))*sech(ro)-y4,'k'); title('Error for qn_4(s)'); xlabel('s');