Loughborough University

PhD thesis

Computation of Scattering Matrices and Resonances for Waveguides

Greg Roddick

Supervised by
Dr. Alexander Strohmaier

April 26, 2016

Contents

1 Introduction **5**

2 Background and definitions **7**

2.1 Basic analysis background . . . 7

2.1.1 Banach and Hilbert spaces . . . 7

2.1.2 Bounded and compact operators . . . 8

2.2 Fourier transforms and distributions . . . 9

2.2.1 Fourier transforms . . . 9

2.2.2 Distribution theory . . . 10

2.3 Sobolev spaces . . . 12

2.4 Unbounded self-adjoint operators on Hilbert spaces . . . 14

2.4.1 A recap of the bounded case . . . 14

2.4.2 Symmetric operators . . . 14

2.5 Friedrich's extension and quadratic forms . . . 15

2.6 Resolvents and spectra of unbounded operators . . . 17

2.7 Spectral theory for compact operators . . . 19

2.8 Elliptic differential operators . . . 19

2.9 The Neumann Laplace operator . . . 20

2.9.1 The Neumann Laplacian and its resolvent . . . 20

2.9.2 Inhomogeneous boundary conditions . . . 21

2.10 Resolvent kernels of $(\Delta - \lambda)$ in $\mathbb{R}$ . . . 21

2.10.1 The Schwartz kernel theorem . . . 22

2.10.2 Calculating resolvent kernels in $\mathbb{R}$ . . . 22

2.11 The Scattering matrix in a general setting . . . 24

3 Thesis background theory **25**

3.1 Waveguides . . . 25

3.1.1 A description of the domain of our resolvent function . . . 27

3.2 The continuation of the resolvent . . . 30

3.3 Generalised Eigenfunctions and the Scattering Matrix . . . 36

3.4 The Neumann to Dirichlet map . . . 40

3.5 Calculating the Neumann to Dirichlet map . . . 40

3.5.1 The Levitin-Marletta method for indirect calculation of the Neumann to Dirichlet map . . . 43

3.5.2 Improving the rate of convergence . . . 43

3.6 An outline of the finite element method . . . 44

3.6.1 Eigenvalue problems . . . 46

3.6.2 The FreeFem++ package and some references to the algorithms . . . 47

4 Calculating scattering matrices and their derivatives 48

4.1 From the Neumann to Dirichlet map to the scattering matrix and its derivatives . . . 48

4.1.1 Calculating the S matrix . . . 48

4.2 Derivatives of the S matrix . . . 50

4.2.1 A Neumann to Dirichlet map for the system on the external domain 50

4.2.2 A Neumann to Dirichlet map for the system on the internal domain 53

4.2.3 Extracting $S^{(n)}(\lambda)$ . . . 54

4.2.4 A few words about computational cost . . . 55

4.3 Embedded eigenvalues and resonances . . . 55

4.3.1 Embedded eigenvalues . . . 55

4.3.2 Resonances and their calculation . . . 56

5 Numerical computations on various domains using the scattering matrix 58

5.1 Cylinders with a Circular Obstacle . . . 58

5.1.1 Some notes on mesh refinement . . . 59

5.1.2 Some notes on the number of eigenvalues . . . 60

5.1.3 Observing a resonance transition from one sheet to another . . . 61

5.2 Some more numerical results . . . 63

5.3 Varying the widths of ends . . . 64

5.3.1 The singe ended case . . . 64

5.3.2 The two ended case . . . 67

5.3.3 The three ended case . . . 69

5.4 Varying the size of an obstacle of fixed position . . . 71

5.4.1 The single ended case . . . 71

5.4.2 The two ended case . . . 72

5.4.3 Three ended case . . . 75

5.5 Varying the position of an obstacle of radius 0.5 . . . 76

5.5.1 The single ended case . . . 77

5.5.2 The two ended case . . . 82

5.5.3 The three ended case . . . 85

5.6 Time delay and scattering length . . . 90

6 Documentation of code 92

6.1 Documentation Freefem++ of code . . . 92

6.1.1 File structure . . . 92

6.1.2 Generating domains . . . 92

6.1.3 Generating a directly computed Neumann to Dirichlet map . . . 94

6.1.4 Generating eigendata . . . 95

6.1.5 A shell script to automate the production of, Neumann to Dirichlet maps and eigendata . . . 95

6.2 Documentation Mathematica code . . . 96

6.2.1 Reading the eigendata . . . 96

6.2.2 Calculating scattering matrices and their derivatives . . . 97

6.2.3 Calculating resonances . . . 100

6.2.4 A sample of the implementation of this code for a search area . . . 106

6.2.5 A visual summary of all of this . . . 108

6.3 3d Domains . . . 109

6.3.1 Cuboidal ends . . . 109

6.3.2 Cylindrical, circular ends . . . 109

6.3.3 Cylindrical ends with arbitrary cross section . . . 109

7 Concluding remarks **110**
7.1 Future directions 110
7.2 Acknowledgements 110

Chapter 1

Introduction

The scattering matrix has its origins in physics, both theoretical and experimental, where it is used to describe the outcome of a scattering event; the scattering of a wave packet originating at infinity. In any such event, a proportion of an incoming wave packet will be transmitted and a proportion reflected, the coefficients of the scattering matrix encode this information. The scattering matrix is a significant component of the branch of mathematical physics know as scattering theory. There is a vast amount of literature, particularly in physics, on potential scattering. For example Reed-Simon's third text in the methods of modern mathematical physics series gives a good introduction to the mathematical background of potential scattering [1]. Mathematicians have also considered more geometric cases, such as the case of manifolds with hyperbolic ends, asymptotically flat manifolds, obstacle scattering in $\mathbb{R}^n$ (see [2], [3], [4] [5],[6]) and, in our case, waveguides in Euclidian space.

Waveguides in Euclidian space are piecewise path connected subsets of $\mathbb{R}^n$ that can be written as the union of a compact domain with boundary and their cylindrical ends. The compact and non-compact parts share a common boundary. This boundary is assumed to be Lipschitz, piecewise smooth and piecewise path connected. The ends can be thought of as the cartesian product of the boundary with the positive real half-line. A notable feature of Euclidian waveguides is that the scattering matrix admits a meromorphic continuation to a certain Riemann surface with a countably infinite number of leaves [2], which we will describe in detail and deal with. In order to construct this meromorphic continuation, one usually first constructs a meromorphic continuation of the resolvent for the Laplace operator. In order to do this, we will use a well known glueing construction (see for example [5]), which we adapt to waveguides. The construction makes use of the meromorphic Fredholm theorem and the fact that the resolvent for the Neumann Laplace operator on the ends of the waveguide can be easily computed as an integral kernel. The resolvent can then be used to construct generalised eigenfunctions and, from them, the scattering matrix.

This thesis is concerned with the computation of the scattering matrix and its derivatives from the Neumann to Dirichlet map on the boundary of the compact domain. This relies on an idea of Levitin and Marletta [7], who use an expression for the Neumann to Dirichlet map in terms of Dirichlet data of Neumann eigenvalues in order to compute embedded eigenvalues and locate resonances. The main advantage of this method is that the computation of the Dirichlet data of Neumann eigenvalues has to be performed only once. This is the most computationally costly step. We show here that the scattering matrix, as well as its derivatives can be obtained from this data directly, by means of linear algebra. Levitin and Strohmaier have used this technique to obtain the scattering matrix on finite volume, non-compact hyperbolic surfaces [3]. Due to the more complicated nature of the

Riemann surface and the fact that the rank of the scattering matrix jumps each time the spectrum is crossed, the problem determining the scattering matrix for waveguides is more complex. We present here an algorithm capable of numerically computing the scattering matrix as well as its derivative at a point on an arbitrary sheet of the Riemann surface.

Being in possession of the scattering matrix allows us to calculate resonances; poles of the scattering matrix. We are able to do this using a combination of numerical contour integration and Newton's method, but for this we require not just the scattering matrix, but its first derivative.

With all of this, we have created code for computing the scattering matrix and its derivatives, with a combination of the FreeFem++ [8] and Mathematica languages. This uses Levitin and Marletta's method and our own extension of it. This code is both fast and flexible, and automated to some extent; the FreeFem++ code generates eigendata from the internal domains, and the Mathematica code uses this to produce the scattering matrix, automatically detecting the number of ends, their widths etc. We have also created a fairly sophisticated script to use the scattering matrix to locate resonances within a given search area.

Using this code, we have been able to replicate results from others, namely Levitin and Marletta [7] and Aslanyan, Parnovski and Vassiliev [9], and compare their results to ours, before presenting some results of our own on a selection of domains. We have also computed the scattering length and time delay for a sample of these domains, and in so doing, verified some formulas from Müller and Strohmaier's paper [10].

The thesis is organised as follows: In the first two chapters, we will outline the necessary background theory needed to properly understand the material. The first chapter covers the basic background theory culminating in the spectral theorem for self-adjoint operators on Hilbert spaces, the Schwartz kernel theorem, Friedrichs extension and existence and regularity for elliptic differential operators. The second chapter deals with background theory specific to this thesis, including a detailed description of waveguides and the Riemann surface that the resolvent operator inhabits, the Neumann to Dirichlet map in abstract then concrete terms, generalised eigenfunctions and defining the scattering matrix and an outline of the finite element method, which has been pivotal to our calculations. The fourth chapter gives a description of the construction of scattering matrices and their derivatives, and the fifth chapter presents some computational results. We will finish off by outlining the code that was used to produce these results, giving full instructions on how to make use of it, for the benefit of anyone who may wish to.

Chapter 2

Background and definitions

2.1 Basic analysis background

2.1.1 Banach and Hilbert spaces

It is assumed that the reader is familiar with the basic definitions and properties of normed vector spaces, Banach spaces, and Hilbert spaces up to the level of an undergraduate analysis course. We will remind the reader of some, but not all, of them here, introducing new material as is necessary for the presentation of the results gathered.

Definition 1. ***Hilbert space***
A Hilbert space $\mathcal{H}$ is an inner product space that is complete with respect to its induced norm $\|v\| = \langle v, v\rangle^{\frac{1}{2}}$.

As the inner product induces a norm, every Hilbert space is also a Banach space [11, Page 8].

Definition 2. ***L^p spaces***
Let (Ω, μ) be a measure space. For any $1 \leq p < \infty$, we define $L^p(\Omega, d\mu)$ to be the set of equivalence classes of measurable functions on Ω such that

$$\|f\|^p_{L^p(\Omega,d\mu)} = \int_\Omega |f(x)|^p d\mu(x), \tag{2.1}$$

is finite.

Two L^p functions are said to be equivalent if they only differ on a set of measure zero. Reed-Simon cover measure theory in their book [11, Page 26], but, for the most part, we will simply be interested in the basic Lebesgue measure.

We can check the norm axioms and see that when $f \in L^p(\Omega, d\mu)$, the $\|f\|_{L^p(\Omega,d\mu)}$ define norms. It can be proved that L^p spaces are, in fact, Banach spaces. For the special case of $p = 2$, we may define an inner product for $f, g \in L^2(\Omega, d\mu)$, by

$$\langle f, g\rangle_{L^p(\Omega,d\mu)} = \int_\Omega f(x)\overline{g(x)} d\mu(x),$$

and obtain a Hilbert space.

Another useful family of spaces, which we shall define here are Frechet spaces. For these we will need to introduce several definitions and concepts before being able to define them. The first is that of a seminorm.

Definition 3. ***Seminorm***
A seminorm is a map $s : V \longrightarrow \mathbb{R}_+$, obeying the following rules:

1. $s(u+v) \geq s(u) + s(v)$
2. $s(au) = |a|.s(u)$

for all $u, v \in V,\ a \in \mathbb{C}$.

This is very similar to the definition of a norm, but with the zero condition dropped; nonzero elements may be mapped to zero by the seminorm.

Definition 4. *Let $\{s_\alpha\}_{\alpha \in A}$ be a family of seminorms, where A is a, not necessarily countable, indexing set. We say that such a family of seminorms* ***separates points****,*
if for all $v \in V \backslash \{0\}$, there exists $\alpha \in A$ such that $s_\alpha(v) \neq 0$.

Definition 5. ***Fréchet space***
A Fréchet space is a vector space V equipped with a countable family of seminorms, $\{s_\alpha\}_{\alpha \in A}$, that separate points.

This next lemma is crucial, as it means that a Frechet space can be thought of as a metric space, and all of the useful properties of metric spaces apply to them.

Lemma 6. *Let V, be a Frechet space, with enumerated seminorms $\{s_j\}_{j=1}^{\infty}$, then V has the structure of a metric space by defining, for $u, v \in V$, the metric d to be*

$$d(u,v) = \sum_{j=0}^{\infty} 2^{-j} \frac{s_j(u-v)}{1+s_j(u-v)}. \tag{2.2}$$

Proof. This is contained in the Appendix in Taylor [12, Page 480], or discussed in detail in a section in Reed-Simon [11, Page131]. □

2.1.2 Bounded and compact operators

A class of very useful operators, which we shall commonly encounter in this thesis, are compact operators. Compact operators have a spectral theorem that greatly simplifies our understanding of them. This section will give an overview of them. We assume knowledge of the various definitions and implications of compactness in a point-set topological context. We will start by stating of some definitions and properties of compact operators.

We call a linear map between two normed spaces X and Y :

$$T : X \longrightarrow Y, \tag{2.3}$$

bounded, if and only if

$$\sup_{x \in X/\{0\}} \frac{\|Tx\|_Y}{\|x\|_X} < \infty.$$

We will denote the vector space of bounded operators from spaces Y to X by $\mathcal{B}(Y,X)$. A stronger property than boundedness is compactness, which we shall now introduce.

Definition 7. ***Compact operator***
Let $T : X \to Y$ be a map between two Banach spaces X and Y.
We say that T is compact if for all bounded subsets of X, their image under T has compact closure.
Or equivalently, for any bounded sequence in X, its image under T has a Cauchy subsequence in Y.

Compact operators have a number of useful properties. The sum of two compact operators is itself compact, the composition of a compact operator with any other operator is compact (they form an ideal) and the limit of a sequence of operators compact in the norm topology is compact.

2.2 Fourier transforms and distributions

Distributions, sometimes called generalised functions, generalise our definitions of functions and of differentiation to a larger category of objects and make various familiar spaces closed. This section gives a brief description of them.

2.2.1 Fourier transforms

Fourier transforms and their inverses are important because they reduce differential operators to multiplication operators when such an operator is conjugated with the Fourier transform. We will briefly outline the basic definitions and state, without proof, the theorems we intend to use throughout this thesis. We will begin by defining the Schwartz space, which will be our first example of a Fréchet space.

Definition 8. ***Schwartz space***
Define seminorms

$$s_j(f) = \sum_{|\alpha|\leq k} \sup \langle x\rangle^j |D^\alpha f(x)| \ j \in \mathbb{N},\ \alpha \in \mathbb{N}^n.$$

We say that a function f is in Schwartz space, or $f \in \mathcal{S}(\mathbb{R}^n)$, if $\forall j \in \mathbb{N}$ there is an $N_j > 0$ such that $s_j(f) < N_j$,.

Definition 9. ***The Fourier transform and its inverse***

For a function $f \in \mathcal{S}(\mathbb{R}^n)$ we define maps $\mathcal{F}, \mathcal{F}^{-1} : \mathcal{S}(\mathbb{R}^n) :\rightarrow \mathcal{S}(\mathbb{R}^n)$ called the Fourier transform and the Fourier inverse respectively given on functions by:

$$(\mathcal{F}f)(\xi) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} e^{-ix\xi} f(x)dx, \quad (\mathcal{F}^{-1}f)(\xi) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} e^{ix\xi} \hat{f}(\xi)d\xi.$$

It can be proved with an elementary calculation that, for functions in Schwartz space, the Fourier transform and its inverse are bone fide maps and that the Fourier inverse is indeed the inverse map of the Fourier transform. This means that the Fourier transform is a bijection from the Schwartz space to itself. What's more, as the Schwartz space is contained in $L^2(\mathbb{R}^n)$, we can look at the L^2 norm and get the following theorem.

Theorem 10. *For functions $f, g \in \mathcal{S}(\mathbb{R}^n)$, $\langle \mathcal{F}f, \mathcal{F}g\rangle = \langle f, g\rangle$, and also $\|\mathcal{F}f\| = \|f\|$.*

The theorem is referred to as Plancherel's theorem or Plancherel's formula. The proof is a series of fairly straightforward, but longwinded calculation that we shall omit here. Although $\mathcal{S}$ is a complete metric space, (the metric is given by equation 2.2), it is not a Hilbert space and therefore too restrictive to be of any use to us. Fortunately, due to $\mathcal{S}(\mathbb{R}^n)$ being dense in $L^2(\mathbb{R}^n)$, we can extend the Fourier transform to a unitary map on $L^2(\mathbb{R}^n)$, but within the inner product only, by thinking of elements in $L^2(\mathbb{R}^n)$ to be L^2 limits of Schwartz functions. i.e. when

$$f, g \in L^2(\mathbb{R}^n),\ \langle \mathcal{F}f, \mathcal{F}g\rangle = \langle f, g\rangle, \tag{2.4}$$

where, in the case of f and g not being in the Schwartz space, they are thought of as L^2 limits of functions that are.

In L^2, we encounter problems when we attempt to compute the Fourier transforms explicitly. For example, if we were to take $f(x) = \frac{1}{1+|x|}$, then $(\mathcal{F}f)$ is not even a function; look at $(\mathcal{F}f)(0)$. Constructing a sequence of Schwartz functions tending towards a given L^2 function is also difficult. One way of overcoming this difficulty is to invoke the Riemann-Lebesgue Lemma:

Lemma 11. *For all $f \in L^1(\mathbb{R}^n)$,*

$$(\mathcal{F}f)(\xi) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} e^{-ix\xi} f(x)dx$$

is well defined, continuous, and $(\mathcal{F}f)(\xi) \to 0$, as $\xi \to \infty$.

A consequence of the Riemann-Lebesgue lemma is that for functions; $f \in L^1(\mathbb{R}^n) \cap L^2(\mathbb{R}^n)$, we are able to explicitly calculate the Fourier transform and can prove that it is the same map as in equation 2.4; the Fourier transform defined on Schwartz space, then extended to L^2. We also note that $L^1(\mathbb{R}^n) \cap L^2(\mathbb{R}^n)$ is dense in $L^2(\mathbb{R}^n)$, indeed, take the truncation. Thus Fourier transforms for L^2 functions can be thought of as limits of the Fourier transforms of their truncations at N as $N \to \infty$.

Another way to approach this problem will lead us into the next section.

2.2.2 Distribution theory

In the last section, when we attempted to compute the Fourier transform of $f(x) = \frac{1}{1+|x|}$, we ran into problems: As a function, it was undefined. In actual fact, it is a Schwartz distribution; a generalisation of $L^2(\mathbb{R}^n)$. The aim of this section is to give an overview of such objects. In order to do so we will need to introduce the weak* topology. Before we do so, we shall recap some of the absolute basics from point-set topology.

A function f mapping one topological space to another is continuous if and only if for all open sets U in the range, their pre-image in the domain, $f^{-1}(U)$ is open. It stands to reason that notions of continuity are heavily dependent on the choice of topology. The same is true for convergence. In a purely topological sense a sequence converges to a point if given a neighbourhood of that point, all elements of the sequence will be contained in that neighbourhood, for sufficiently large N. Here, neighbourhood of a point is defined to be an open set containing that point. Now if we have an underlying set X, and two topologies, $\mathcal{F}_1, \mathcal{F}_2$. If $\mathcal{F}_1 \subset \mathcal{F}_2$, then we say that $\mathcal{F}_1$ is weaker than $\mathcal{F}_2$. This is due to the fact that convergence of sequences in $\mathcal{F}_1$ is a weaker property than $\mathcal{F}_2$: more neighbourhoods implies more opportunities for convergence to fail. Let us now state a very basic definition.

Definition 12. ***Bounded linear functional***
Let $\mathcal{H}$ be a Hilbert space $\mathbb{C}$, a bounded linear map

$$T : \mathcal{H} \longrightarrow \mathbb{C}$$

we call a ***bounded linear functional.***

We will now state, without proof, a very important theorem:

Theorem 13. ***Riesz representation theorem***
Let $\mathcal{H}$ be a Hilbert space and let $T \in \mathcal{H}^*$, a bounded linear map from $\mathcal{H}$ to $\mathbb{C}$. Then there exists a unique $g \in \mathcal{H}$ such that $T(f) = \langle f, g \rangle$, and also $\|T\|_{\mathcal{H}^*} = \|g\|_{\mathcal{H}}$.

This theorem is significant because it says that Hilbert spaces are self-dual.

Definition 14. ***Weak topology***
Let $\mathcal{H}$ be a Hilbert space. The weak topology on $\mathcal{H}$ is the weakest topology such that every functional, as defined in the Riesz representation theorem, is continuous.

When $f_n \in \mathcal{H}$ is a sequence. We say that f_n converges to f in the weak topology, or f_n weakly converges to f. If for all $T \in \mathcal{H}^*$ then $T(f_n) \to T(f)$.

Definition 15. ***Weak* topology***
Let X be a Banach space, the weak topology on X^* is the weakest topology such that, for all $T \in X^*$ and $x \in X$, the functions $T \mapsto T(x)$ are continuous.*

A sequence of functionals $T_n \in X^*$ converges in the weak* topology to $T \in X^*$ if for all $x \in X$, $T_n(x)$ converges to $T(x)$ point-wise.

Definition 16. ***Tempered Distribution***
We say that the space of tempered distributions $\mathcal{S}'(\mathbb{R}^n)$, acting on Schwartz functions, is the set made up of continuous linear functionals

$$T : \mathcal{S}(\mathbb{R}^n) :\longrightarrow \mathbb{C}$$

equipped with the weak topology.*

It ought to be mentioned that $\mathcal{S}$ is a Fréchet space and not a Banach space, which makes $\mathcal{S}'$ somewhat out of place in this section but, as we shall see in due course, it allows us to introduce the distributional derivative in the most natural way, so we will overlook this.

We should observe here that $L^2(\mathbb{R}^n)$ can be included in the space of Schwartz distributions as, for some $f \in L^2(\mathbb{R}^n)$, $\langle \cdot, f \rangle$ defines a continuous linear functional acting on $\mathcal{S}(\mathbb{R}^n)$. Distributions allow us to extend our notion of differentiation:

Definition 17. ***Distributional derivative***
Let $f \in \mathcal{S}'(\mathbb{R}^n)$ and $g \in \mathcal{S}(\mathbb{R}^n)$. We define

$$(D^\alpha f)(g) = (-1)^{|\alpha|} \int_{\mathbb{R}^n} f(x)(D^\alpha g)(x)dx$$

where D^α is the partial differential operator of order $|a|$, which is given component-wise by $\alpha = (\alpha_1, \ldots, \alpha_n)$, thus

$$D^\alpha = (-i)^{|\alpha|} \frac{\partial^{\alpha_1}}{\partial x_1} \cdots \frac{\partial^{\alpha_n}}{\partial x_n}.$$

If $D^\alpha f$ is indeed well defined, as a function, then integration by parts shows us that $D^\alpha f$ as a distributional derivative $(-1)^{|\alpha|}(f, D^\alpha \cdot)$ is the same as the functional $(\cdot, D^\alpha f)$, otherwise, the fact that g is a Schwartz function, means that $D^\alpha f$ is always exists as a distribution.

When attempting to apply the Fourier transform to $L^2(\mathbb{R}^n)$ in the last section, we encountered the problem that for an $f \in L^2(\mathbb{R}^n)$, $(\mathcal{F}f)$ may not exist as a function; it does, however always exist as a Schwartz distribution. We can see this by explicitly defining it, for $f \in L^2(\mathbb{R}^n)$: $(\mathcal{F}f) = (f, \mathcal{F}\cdot)$. This always exists by Theorem 10, and, in the case of $(\mathcal{F}f)$ existing as a function; we see that the distribution associated to it will be $(\mathcal{F}f) = ((\mathcal{F}f), \cdot) = (\mathcal{F}f) = (f, \mathcal{F}\cdot)$, by Theorem 10. Thus we can finally state that the Fourier transform and its inverse extend uniquely to Schwartz distributions, and are continuous in the weak* topology.

We will end this section by defining some other distributions, which will be of use to us later on. A general distribution $\mathcal{D}'(\mathbb{R}^n)$ acts on $C_0^\infty(\mathbb{R})^n$, the space of **test functions** on $\mathbb{R}^n$.

Definition 18. ***General Distribution***
A general distribution on $\mathbb{R}^n$ *is a map*

$$T : C_0^\infty(\mathbb{R}^n) \longrightarrow \mathbb{C}$$

that is both linear and continuous in the topology of uniform convergence. We denote the space of all such distributions on $\mathbb{R}^n$ *as* $\mathcal{D}'(\mathbb{R}^n)$.

Definition 19. ***Compactly supported distribution***
We say that a distribution $T \in \mathcal{D}'(\mathbb{R}^n)$ *is compactly supported if there is a closed, bounded set* K*, such that when* T *is paired with a test function* g *that vanishes in a neighbourhood of* K*, then* $T(g) = 0$*. We denote the space of compactly supported distributions by* $\mathcal{E}'(\mathbb{R}^n)$*. We call the smallest such set the support of* T.

2.3 Sobolev spaces

The properties of Sobolev spaces play a very important, as we shall see in the next section; they are the natural domains for various differential operators. A crucial property of Sobolev spaces is that, given certain criteria on their domain and order, that compactly embed into each other. Later on this will be a vital tool in proving that certain operators are compact, with all of the consequences that ensue from this. With this in mind we begin.

Definition 20. ***Sobolev spaces on*** $\mathbb{R}^n$
Define

$$H^k(\mathbb{R}^n) = \{f \in L^2(\mathbb{R}^n) | D^\alpha f \in L^2(\mathbb{R}^n),\ \forall |\alpha| \leq k\}.$$

Where $k \in \mathbb{N}$, $\alpha \in \mathbb{N}^n$ is a multi-index, and the differential operator D^α, considered to be a distributional derivative acts on $L^2(\mathbb{R}^n)$, though we consider $D^\alpha f$ as a Schwartz distribution. [12, Page 204]

We can extend this definition using the Fourier transform so that we can have Sobolev spaces of real order, not just positive integers.

Definition 21.

$$H^s(\mathbb{R}^n) = \{f \in \mathcal{S}'(\mathbb{R}^n) | \langle \xi \rangle^s (\mathcal{F} f) \in L^2(\mathbb{R}^n)\}.$$

Where $\langle \xi \rangle = (1 + |\xi|^2)^{\frac{1}{2}}$.

Simply observing the Fourier transform's action on $D^\alpha f$ and vice versa ought to be enough to convince us that when $s \in \mathbb{N}$, these definitions are equivalent. Sobolev spaces become Hilbert spaces, if we define the Soblolev inner product to be

$$\langle f, g \rangle_{H^s(\mathbb{R}^n)} = \langle \mathcal{F}^{-1}(\langle \xi \rangle^s \mathcal{F} f), \mathcal{F}^{-1}(\langle \xi \rangle^s \mathcal{F} g) \rangle_{L^2(\mathbb{R}^n)}.$$

Using the Riesz representation theorem, we can see that $H^s(\mathbb{R}^n)^* = H^{-s}(\mathbb{R}^n)$.

Sobolev spaces on compact bounded domains $X \subset \mathbb{R}^n$ are defined in exactly the same was as in Definition 20, and subject to certain conditions for the smoothness of the boundary, lead to Rellich's theorem. The main condition is that X must be a Lipschitz domain, which we shall define here.

Definition 22. ***Lipschitz domain***
Let $X \subset \mathbb{R}^n$*, we say that* X *is a* ***Lipschitz domain*** *if for all points* x *on the boundary* ∂X*, there is a neighbourhood of* x *on which* ∂X *is the graph of a Lipschitz function.*

Definition 23. ***Compact embedding***
Let, Banach spaces, $V \subset W$ be an embedding, we say such an embedding is compact if the identity operator

$$I : V \to W$$

is a compact operator.

We would ultimately like to have a coherent definition of Sobolev spaces on compact manifolds with boundary. We would also like to prove that the embeddings of higher order Sobolev spaces into lower order Sobolev spaces are compact: a version of the Relich-Kondrachov theorem. Taylor devotes a section to this in his book to prove this [12, Page 283]. We intend not to fully reproduce his proof here, but to give a very rough commentary of how this was achieved. Let us define:

Definition 24. ***Sobolev spaces on compact manifolds***
Let M be a compact manifold and $U \subset M$ be a co-ordinate chart. Define:

$$H^s(M) = \{f \in \mathcal{D}'(M) : \ \forall U \ \textit{and} \ g \in C_0^\infty(U), \ f.g \in H^s(U)\},$$

where we obtain $H^s(U)$ by identifying U, with the image of its co-ordinate maps in $\mathbb{R}^n$.

It must be proved that definition 24 is not dependent on the choice of co-ordinates for M. This is achieved by showing that, if $T : \mathbb{R}^n \to \mathbb{R}^n$ is a linear map, which, in the case of manifolds, could be our change of co-ordinates, then

$$T^* : H^s(\mathbb{R}^n) \to H^s(\mathbb{R}^n).$$

Where $T^*f(x) = f(T(x))$. For integer valued s, this can be achieved by the chain rule, and extends to manifolds by considering its actions on the coordinates. To prove this for non-integer s, complex interpolation must be used, which Taylor explains in detail [12, Page 284].

On the n-Torus, the Fourier transform becomes discrete (Fourier series), thus the operator of $\mathcal{F}^{-1}\langle\xi\rangle^{-s}\mathcal{F}$ will be limit of a sequence of finite rank (compact) operators, and thus compact. This is used to show that the embedding $H^s(\mathbb{T}^n) \subset H^t(\mathbb{T}^n)$ is compact for $s > t$. We can then use the fact that X is compact to pick a co-ordinate cover that maps onto $\mathbb{T}^n$ for each chart and construct $H^s(X)$ by using a partition of unity subordinate to our choice of co-ordinates. This is used to prove that the embedding $H^s(\mathbb{T}^n) \subset H^t(\mathbb{T}^n)$ is also compact.

When X is a compact manifold with piecewise smooth Lipschitz boundary ∂X, we can extend X along a tubular neighbourhood [13, Page 255] about ∂X. This can in turn be smoothed at its boundary, then identified with a copy of itself to produce M; a compact manifold (without boundary). The boundary, MUST, be piecewise smooth and Lipschitz in order for us to be able to do this. We can now simply define $H^s(X)$ using equivalence classes as follows:

Definition 25. ***Sobolev spaces on compact manifolds with boundary***

$$H^s(X) \cong H^s(M)/\{f \in H^s(M)| \quad f|_X = 0\}.$$

The embedding $H^s(X) \subset H^t(X)$ is proved to be compact by proving that the map,

$$E : H^s(X) \to H^s(M)$$

is well defined and

$$R : H^s(X) \to H^s(M)$$

is onto, before using a commutative diagram argument, together with the fact that compositions of compact operators are compact to finally prove the result.

2.4 Unbounded self-adjoint operators on Hilbert spaces

We will assume here that the reader is familiar with the basics of point-set topology; the notion of compactness/sequential compactness, boundedness, completeness, open/closedness, density and duality. Otherwise see [11, Page 90]. Those who are familiar with the notion of self-adjointness in the case of bounded operators will appreciate its significance for the study of those objects: It makes possible the spectral theorem; an infinite dimensional analogue of the diagonalisation of matrices. This is, without doubt, a highly desirable property to have in an operator one wishes to study. In the unbounded case, things become more complicated, but nevertheless, self-adjointness is still definable and still leads to a spectral theorem. Will will outline this theory here, starting with a recap of the bounded case.

2.4.1 A recap of the bounded case

The Riesz representation theorem is significant here; because of the self-duality of Hilbert spaces, it leads to a natural definition of the adjoint operator in the case of bounded operators. For an operator $T : \mathcal{H} \to \mathcal{H}$, we can simply define T^* to be the unique operator such that $\langle Tf, g \rangle = \langle f, T^*g \rangle$, $\forall f, g \in \mathcal{H}$. To prove existence and uniqueness of this operator, we can simply apply the Riesz Representation theorem. We say that T is self-adjoint if $T^* = T$. Of course all of this assumes boundedness. Once this condition is no longer satisfied, we encounter problems. Overcoming these to obtain a working notion of self-adjointness will occupy the rest of the section.

2.4.2 Symmetric operators

We will now move on and outline some background theory relating to unbounded linear operators on Hilbert spaces. Differential operators are the prime example of such objects. As a map between normed spaces, the operator's continuity in the operator norm is equivalent to its boundedness. Boundedness is a condition for many of the standard theorems one may encounter when studying operators and their spectra; it is a necessary condition of compactness, which we shall discuss in the next section, and it also alleviates, to some extent, the issue of domains, which we must deal with. For reasons that will become clear later on, we will not attempt to define an unbounded operator (T say) on the whole of our Hilbert space $\mathcal{H}$ but instead on some subspace $D(T)$, which we will call the **domain** of T, so that we can write our, now unbounded T as

$$T : D(T) \to \mathcal{H}.$$

If $D(T)$ is dense in $\mathcal{H}$, as is normally the case, then we say that T is densely defined.

Definition 26. ***Graph and Graph norm***
*The **graph** of a linear operator T is the set*

$$\{(f, Tf) \ : \ f \in D(T)\} \subset \mathcal{H} \times \mathcal{H}.$$

This is often denoted $\Gamma(T)$ [11, Page 250]. We can define an inner product, and consequently a norm, on such a set as follows:

$$\langle (f_1, \psi_1), (f_2, g_2) \rangle = \langle f_1, f_2 \rangle + \langle g_1, g_2 \rangle.$$

We say that an operator T is **closed** if its graph is a closed subset of $\mathcal{H} \times \mathcal{H}$.

Definition 27. *T' is an* ***extension*** *of operator T on $\mathcal{H}$ if, $\Gamma(T) \subset \Gamma(T')$, and is also a bone-fide operator on $\mathcal{H}$ itself.*

An operator is **closable** if it admits a closed extension. We are now ready to introduce the notion of the adjoint of an unbounded operator.

Definition 28. ***The adjoint of an unbounded operator***
Let $T : D(T) \to \mathcal{H}$ be a densely defined operator on a Hilbert space $\mathcal{H}$
Let $D(T^*) \subset \mathcal{H}$ be the set of g such that there exists $\eta \in \mathcal{H}$ and

$$\langle Tf, g \rangle = \langle f, \eta \rangle \quad \forall f \in D(T).$$

We define $T^*\varphi$, the adjoint of T by $T^*\varphi = \eta$.

An important property of the adjoint is that it is always closed [11, Page 253]. If we happen to apply definition 2.4.2 to the case of a bounded operator, we very quickly see that $D(T) = D(T^*)$ and we have no issues. This is because for every $g \in \mathcal{H}$, the functional $\langle T\cdot, g \rangle : \mathcal{H} \to \mathbb{C}$ is always a bounded linear functional; the boundedness of T guarantees it, so by the Riesz representation theorem, there will be a $\eta = T^*\varphi$ always. When T is unbounded, this may or may not be so, thus we need the following:

Definition 29. ***Symmetric operator***
Let T be densely defined on $\mathcal{H}$ We say that T is ***symmetric*** *if and only if $D(T) \subset D(T^*)$ and*

$$\langle Tf, g \rangle = \langle f, Tg \rangle, \quad \forall f, g \in D(T).$$

We say that T is ***self-adjoint*** *if, in addition to the above, $D(T) = D(T^*)$.*

We can see that a symmetric operator is closable as $D(T) \subset D(T^*)$, and T^* is a closed extension of T, but even a closed symmetric operator may still fail to be self-adjoint. The next theorem, can be thought of a a corollary of the closed graph theorem [11, Page 84] it will cause major problems for us, as we shall now see.

Theorem 30. ***Hellinger-Toeplitz theorem***
An everywhere defined ($D(T) = \mathcal{H}$) symmetric operator T is necessarily bounded.

This means that to make sense of symmetric operators on Hilbert spaces, we can't think of them as being defined on the whole space, but must instead think of them as extensions of symmetric operators, defined on a dense subspace of our Hilbert space called a **core**. These extensions might not exist, and are by no means unique. In the case of a symmetric operator having a unique self-adjoint extension we call it **essentially self-adjoint**.

2.5 Friedrich's extension and quadratic forms

We left the last section with the theory of self-adjoint operators. We showed that unbounded self-adjoint operators make sense only as extensions of densely defined, symmetric operators, and that given a densely defined symmetric operator, self adjoint extensions may not be unique, or even exist. We will now present a construction that, in the case of our operator in question being positive, gives a self-adjoint extension with a number of interesting properties.

Definition 31. ***Quadratic form***
Let $Q(q)$ be a dense vector subspace of our Hilbert space $\mathcal{H}$. A map

$$q : Q(q) \times Q(q) \longrightarrow \mathbb{C},$$

that is sesquilinear in the same order as the inner product, we define to be a ***quadratic form****, and we call $Q(q)$ its* ***form domain****.*

We call a quadratic form **symmetric** if $q(\phi, \varphi) = \overline{q(\varphi, \phi)}$ for all $\phi, \varphi \in Q(q)$ and **positive** if $q(\varphi, \varphi) \geq 0$ for all $\varphi \in Q(q)$. We could further generalise and introduce the concept of semi-boundedness for quadratic forms, as Reed and Simon do, but positivity alone will suffice for what we wish to do with them. [11, Pagr 276]

Of particular interest to us will be the fact that if T is a positive symmetric operator on $\mathcal{H}$, we can obtain a quadratic form q by setting

$$q(\phi, \varphi) = \langle T\phi, \varphi \rangle,$$

with form domain $D(T)$. We have already encountered closed operators in 2.4.2, whose graph norm is a closed subset of $\mathcal{H} \times \mathcal{H}$. We will now introduce the analogue concept for a quadratic form and associated norm.

Definition 32. ***Closed and closable quadratic forms***
Let q be a positive quadratic, we say that it is ***closed****, if the form domain $Q(q)$ is complete in the norm*

$$\|\varphi\|_{+1} = \sqrt{q(\varphi, \varphi) + \langle \varphi, \varphi \rangle}.$$

We say that q is ***closable*** *if for all sequences $\varphi_n \in Q(q)$, such that $\varphi_n \to 0$ in $\mathcal{H}$ as $n \to \infty$ and $q(\varphi_n - \varphi_m, \varphi_n - \varphi_m) \to 0,\ n, m \to \infty$ in $\|\cdot\|_{+1}$, then $\varphi_n \to 0, n \to \infty$ in $\|\cdot\|_{+1}$.*

This next theorem is provides justification as to why it is beneficial to look at quadratic forms and not simply look at the operators as before.

Theorem 33. *Any positive, closed, symmetric quadratic form is the quadratic form associated to a unique self-adjoint operator.*

This is property is especially pleasant for us as it means that, quadratic forms need only be closed, bounded and symmetric to be self-adjoint, and leads us naturally to the main theorem we wish to present in this section.

Theorem 34. ***Friedrichs extension***
Let T be a positive symmetric operator and let

$$q(\phi, \varphi) = \langle T\phi, \varphi \rangle,\ \phi,\ \varphi \in D(T). \tag{2.5}$$

Then q is a closable quadratic form, whose closure $\hat{q}$ is the quadratic form of a unique self-adjoint operator $\hat{T}$.

$\hat{T}$ ***is defined to be the Friedrichs extension of*** *T. $\hat{T}$ is a positive extension of T and the lower bound of its spectrum is the lower bound of q. Also, $\hat{T}$ is the only self-adjoint extension of T whose domain is contained in the form domain of $\hat{q}$. [11, Page177]*

Example 35. ***The Dirichlet and Neumann Operators on an interval*** *For simplicity, we will perform this calculation in one dimension, although it extends to higher dimensions.*

Here we will let $q(\phi,\varphi) = \langle \frac{d\phi}{dx}, \frac{d\varphi}{dx} \rangle$*, with the form-domain of* q *being* $C^\infty[0,1] \subset L^2[0,1]$.

We will first show that this form is closable, then work out the form domain of its closure and the self-adjoint operator associated to it.

Going on our definition of closability, we will observe the fact that a sequence $\varphi_n \in C^\infty[0,1]$ *such that* $\varphi_n \to 0,\ n \to \infty$ *in* $L^2[0,1]$ *and* $q(\varphi_n - \varphi_m, \varphi_n - \varphi_m) \to 0,\ n, m \to \infty$ *in* $\|\cdot\|_{+1}$, *then* φ_n' *converges to an element* $\psi \in L^2[0,1]$ *in the* L^2 *norm, so we must now show that it is* 0. *Let* $\hat{\psi} = \int_{x_0}^1 \psi$, *for any* $x_0 < 1$*; we claim that* $\varphi_n \to \hat{\psi},\ n \to \infty$. *Now*

$$\|\varphi_n - \hat{\psi}\| = \|\varphi_n - \int_{x_0}^{x} \psi\| = \|\int_{x_0}^{x} (\varphi_n' - \psi) + \varphi_n(x_0)\| \leq \int_{x_0}^{x} \|\varphi_n' - \psi\| + \|\varphi_n(x_0)\|,$$

by the triangle inequality. Since $\varphi_n \to 0,\ n \to \infty$ *in the* L^2 *norm, we know that a subsequence of* φ_n *tends to* 0 *point-wise outside a null set (we refer the reader to Reed-Simon's section on this for discussion and definition of null sets). Thus we can pick an* x_0 *where* $\varphi_{n_j} \to 0,\ j \to \infty$, *and see that* $\hat{\psi}$ *is indeed the* L^2 *limit of* φ_n, *which we have previously assumed to be* 0. *This means that* $\hat{\psi}$ *is* 0 *in the* L^2 *norm and so is* ψ. *Therefore q is closable.*

We now need to know the form domain for $\hat{q}$ *the closure of* q. *A good candidate will be* $H^1[0,1]$. *Any* $\varphi \in Q(\hat{q})$ *is the limit of a sequence of* C^∞ *functions* φ *such that both* $\varphi_n \to \varphi$ *and* $\varphi_n' \to \varphi'$ *in the* L^2 *norm as* $n \to \infty$*; their limit will be in* $H^1[0,1]$. *Conversely, any* $\varphi \in H^1[0,1]$ *will be the limit of a sequence of* C^∞ *functions in the* $\|\cdot\|_{+1}$ *norm.*

Theorem 33 tells us that $\hat{q}$ *is the quadratic form of a unique self-adjoint operator, such that* $q(\phi,\varphi) = \langle T\phi, \varphi\rangle$. *We can see that for* $\phi, \in C^\infty[0,1],\ \varphi \in H^1[0,1]$,

$$\hat{q}(\phi,\varphi) = \langle \frac{d\phi}{dx}, \frac{d\varphi}{dx} \rangle = \langle -\frac{d^2\phi}{dx^2}, \varphi \rangle + \frac{d\phi}{dx}\overline{\varphi}(1) - \phi\frac{d\phi}{dx}\overline{\varphi}(0).$$

This means that $\frac{d\phi}{dx}(1) = \frac{d\phi}{dx}(0) = 0$, *and the Neumann Laplacian is the self-adjoint operator associated to* $\hat{q}$ *whose domain* $\{\phi \in H^1[0,1] : \frac{d\phi}{dx}(1) = \frac{d\phi}{dx}(0) = 0\}$ *is contained (strictly in this case) in that of* $\hat{q}$.

If instead of using $C^\infty[0,1]$ *for our domain, we used* $C_0^\infty(0,1)$, *smooth functions on* $(0,1)$ *with compact support, then the Friedrich's extension would be the Dirichlet Laplacian.*

2.6 Resolvents and spectra of unbounded operators

Having introduced self-adjoint operators, either in the simple case of bounded operators, or the more involved, unbounded case, we are now in a position to make use of this property; the fact that the spectral theorem is applicable to them. In this section we will adapt some of the more basic definitions before concluding the section with the spectral theorem for self-adjoint operators on Hilbert spaces. The absolute basics to begin with:

Definition 36. ***Resolvent set***
Let T *be a, not necessarily bounded, closed operator on a Hilbert space* $\mathcal{H}$. *We define the* ***resolvent set*** $\rho(T)$, *to be the* $\lambda \in \mathbb{C}$ *such that the operator*

$$(T - \lambda) : D(T) \longrightarrow \mathcal{H}$$

is a bijection of $D(T)$ into its image in $\mathcal{H}$, with bounded inverse. [11, Page 253]

Obviously if T is a bounded operator, one may not worry so much about the issue of domains and this reduces down to the same definition as earlier. The inverse being bounded, in any case, is a consequence of the bounded inverse theorem [11, Page 83].

Definition 37. ***Resolvent function***
We call the operator valued function

$$R_\lambda(T) : \mathbb{C} \longrightarrow \mathcal{B}(\mathcal{H}, \mathcal{H}) \text{ where } R_\lambda(T) = (T - \lambda)^{-1}$$

the ***resolvent function****, or sometimes just the resolvent.*

This function is holomorphic (see [14]) on the resolvent set. It has some noteworthy properties, which we shall now outline, before introducing the spectrum.

Lemma 38. ***The resolvent equation***
Let $\lambda, \lambda_0 \in \rho(T)$ then

$$R_\lambda(T) - R_{\lambda_0}(T) = (\lambda - \lambda_0)R_\lambda(T)R_{\lambda_0}(T). \tag{2.6}$$

Proposition 39. *Let $\lambda, \lambda_0 \in \rho(T)$ and $|\lambda - \lambda_0| < \|R_{\lambda_0}(T)\|^{-1}$, then*

$$R_\lambda(T) = \sum_{n=0}^{\infty}(\lambda - \lambda_0)^n R_{\lambda_0}(T)^{n+1}. \tag{2.7}$$

In order to have a spectral theorem, we must define the spectrum.

Definition 40. ***Spectrum***
The complement of the resolvent set $\mathbb{C} - \rho(\lambda)$, we call the ***Spectrum*** *of T, denoted $\sigma(T)$. This can be decomposed as follows:*

- ***Point spectrum*** *or set of* ***eigenvalues*** *consists of the set of $\lambda \in \mathbb{C}$, for which there exists non-zero $\varphi \in X$, such that $T\varphi = \lambda\varphi$. We call such φ* ***eigenfunctions****.*
- ***Residual spectrum*** *comprises of the union spectrum, with the complement of the set of eigenvalues.*

Finally we have arrived at a point where we may present the spectral theorem for unbounded self-adjoint operators. We will present the version derived by E.B. Davies in his text, Spectral theory and Differential Operators; it being the most elegant. [15, Page 37]

Theorem 41. ***The Spectral Representation***
Let T be a self-adjoint operator on $\mathcal{H}$, a Hilbert space. Then there exists a finite measure on $\sigma(T) \times \mathbb{N}$ on and a unitary operator

$$U : \mathcal{H} \longrightarrow L^2(\sigma(T) \times \mathbb{N}, d\mu)$$

so that

- *If the function $h : \sigma(t) \times \mathbb{N} \to \mathbb{R}$ is defined as $h(\lambda, n) = \lambda$, then an $f \in \mathcal{H}$ is in $Dom(T)$ if and only if $h.U(f) \in L^2$.*
- *$(UTU^{-1}f) = hf$ for all $f \in U(D(T))$*
- *$U\varphi(T)U^{-1}f = \varphi(h)f$, forall $\varphi \in C_0(\mathbb{R})$, $f \in L^2(\sigma(T) \times \mathbb{N}, d\mu)$.*

This can be uniquely extended from $C_0(\mathbb{R})$ to $\mathcal{B}(\mathbb{R})$, the space of bounded Borel functions on $\mathbb{R}$

2.7 Spectral theory for compact operators

We defined compact operators in the first section but, up until now, have not made much use of them. Having defined the spectrum and presented a spectral theorem for self-adjoint operators in the previous section, we must now do the same for compact operators. Compact operators, being bounded, relieve us of the burden of having to define them on a dense domain of our space; we may simply define them on the whole space. This greatly simplifies the definition of the resolvent set and resolvent. It turns out that the spectrum of compact operators has a number of remarkable properties:

Let $\mathcal{H}$ be a Hilbert space and $T : \mathcal{H} \longrightarrow \mathcal{H}$ be compact, then:

- $\sigma(T)$ is a discrete, at most countable set, with at most one accumulation point at 0.
- $\forall\lambda \in \sigma(T)$ λ is also an eigenvalue of T with, at most, finite multiplicity.

We can now state the spectral theorem for compact operators on a Hilbert Space.

Theorem 42. ***Spectral theorem for compact operators***
Let T be a compact operator, on a separable Hilbert space space $\mathcal{H}$. Then there exists a finite or countable set of orthonormal eigenfunctions $\{f_n\}_{n=1}^N \in \mathcal{H}$, and associated eigenvalues $\{\lambda_n\}_{n=1}^N \in \mathbb{C}$, where $N \in \mathbb{N}$ or N"="∞, such that for some $\varphi \in \mathcal{H}$,

$$Tf = \sum_{n=1}^{N} \lambda_n \langle f, f_n \rangle f.$$

2.8 Elliptic differential operators

We will now digress slightly in order to introduce an important theorem in the theory of differential operators, which we will use several times throughout this thesis. Before giving the elliptic regularity theorem, it is necessary to define an elliptic partial differential operator.

Definition 43. ***Elliptic operator***
A partial differential operator $P(D)$ of order m,

$$P(x, D) = \sum_{|\alpha| \leq m} a_\alpha(x) D^\alpha,$$

is said to be elliptic provided $|P(x, \xi)| \geq C|\xi|^m$, for large $|\xi|$. [12, Page 245]

With this defined, we may state the theorem.

Theorem 44. ***Elliptic regularity theorem***
*Let $P(x, D)$ be an elliptic partial differential operator, and take $\varphi \in \mathcal{D}'(\mathbb{R})^n$.
If $P(x, D)\varphi = f$, for some $f \in \mathcal{D}'(\mathbb{R}^n)$, then*

$$\text{sing supp } \varphi = \text{sing supp } f.$$

[12, Page 247]

This is important but inadequate for what we intend to do. We require boundary regularity on piecewise smooth, convex Lipschitz domains (see definition 22). Let us now define an elliptic boundary problem on them.

Definition 45. ***Elliptic boundary value problem***
Let P be an elliptic differential operator, on a Lipschitz domain X, of order m and let B_j be a family of differential operators defined in a neighbourhood of ∂X of order strictly less than m. An elliptic boundary value problem is an equation of the form

$$P(D)u = f \text{ on } X, \quad B_j u = g_j \text{ on } \partial X.$$

Theorem 46. ***Elliptic boundary regularity***
If $P(D)$ is elliptic on $\overline{X}$ and the boundary value problem satisfies the hypothesis of regularity, when the coefficients of the operator are fixed, then given $u \in H^m(X)$, if $Pu \in H^k(X)$, for some $k \in \mathbb{N}$ and $B_j u \in H^{m+k-m_j-1/2}(\partial X)$, then $u \in H^{m+k}(X)$ and

$$\|u\|^2_{H^{m+k}(X)} \leq C \left(\|Pu\|^2_{H^k(X)} + \sum_j \|B_j u\|^2_{H^{m+k-m_j-1/2}(\partial X)} + \|u\|^2_{H^{m+k-1}(X)} \right).$$

(see [12, Page 383])

2.9 The Neumann Laplace operator

In cartesian co-ordinates, we will no doubt, be aware the Laplace operator's action on functions is to sum their second derivatives with respect to each co-ordinate, multiplied by minus 1. The latter to ensure the operator is positive, (see section 2.5). In this section, we will give a more, in depth overview of the Neumann Laplace operator, which we first encountered in Example 35. We focus on the the Neumann Laplace operator here as this will feature heavily in the rest of the thesis, although, the material in this section, applies to the Dirichlet Laplace operator with minimal modification.

2.9.1 The Neumann Laplacian and its resolvent

When X is a connected compact manifold, with nonempty smooth boundary, we define the Neumann Laplacian to be Friedrichs extension of the Laplace operator on $C^\infty(X)$. For $u, v \in C^\infty(X)$, we observe that

$$(\Delta\varphi, \psi) = \langle d\varphi, d\psi\rangle - \int_{\partial X} \frac{\partial \varphi}{\partial n}\overline{\psi} dS.$$

However, due to the Neumann boundary conditions (necessary for $(\Delta\varphi, \psi)$ to be a closed quadratic form), $\frac{\partial\varphi}{\partial n}$ vanishes on ∂X, and thus the integral vanishes. The domain of the operator is $H^2(X)$ and $H^1(X)$ for its associated quadratic form $\langle d\varphi, d\psi\rangle$.

We introduce the operator

$$Q_{\mathcal{N}} : H^1(X) \to H^1(X)^*,$$

where $H^1(X)^*$ is the dual space to $H^1(X)$, which acts on it as follows:

$$u \mapsto \langle d\,\cdot\,, du\rangle \in H^1(X)^* \text{ for some } u \in H^1(X).$$

It is proved that the operator

$$Q_{\mathcal{N}} + 1 : H^1(X) \to H^1(X)^*,$$

is bijective, whose inverse, when restricted to $L^2(X)$ is compact by Rellich's theorem and self-adjoint: Symmetric and self-adjointness are equivalent in the bounded case. So then there will be an orthonormal basis of $L^2(X)$, consisting of eigenfunctions of the inverse: T_N, whose discrete set of associated, real eigenvalues $\{\alpha_j\}$, has only one limit point, namely 0. Thus, when arranged in descending order, they form a non increasing sequence $m_j \to 0$ as $j \to \infty$. We can now see that $Q_{\mathcal{N}}$ has non-zero eigenvalues $\mu_j = \frac{1}{\alpha_j} - 1$, tending to infinity.

We can see, by inspection, that all of the above remains true true for $Q_{\mathcal{N}} - \lambda$, for any such λ for which $Q_{\mathcal{N}} - \lambda$ is bijective.

We have thus constructed the resolvent of the associated quadratic form to $(\Delta - \lambda)$ and shown the existence of the resolvent of the operator $R_\lambda(\Delta)$. If we wish to do the same for the Dirichlet Laplacian, we can do so in exactly the same way, but we must use appropriate domains for such an operator. It should also be noted that as an operator valued function, $R_\lambda(\Delta)$ is analytic (as will be the resolvent for any operator) in its resolvent set, which in this case will be the complex place minus the discrete set of Neumann eigenvalues μ_j.

2.9.2 Inhomogeneous boundary conditions

In the previous section, we introduced the resolvent for the Neumann Laplacian, on a compact manifold, with piecewise-smooth Lipschitz boundary. In this section, we will use it to show existence and uniqueness of the problem

Lemma 47. *Let φ be such that, for some non-zero $g \in C^\infty(\partial X)$,*

$$(\Delta - \lambda)\varphi = 0, \ \frac{\partial \varphi}{\partial n}|_{\partial X} = g, \ \varphi \in L^2(X), \tag{2.8}$$

where $\lambda \in \mathbb{C}$ is not a Neumann eigenvalue. Such a function exists, and is unique.

The proof comes from Taylor [12, Page 308].

Proof. Let $F \in C^\infty(X)$ be such that $\frac{\partial F}{\partial n}|_{\partial X} = g$.
Then the φ in equation 2.8 is equivalent to $\varphi = F + \psi$, where ψ is a solution to

$$(\Delta - \lambda)\psi = (\lambda - \Delta)F, \quad \frac{\partial \psi}{\partial n}|_{\partial X} = 0.$$

Thus we can see that $\psi = (Q_{\mathcal{N}} - \lambda)^{-1}(\lambda - \Delta)F$ and thus $\psi \in H^1(X)$. We can now construct our φ as

$$F + (Q_{\mathcal{N}} - \lambda)^{-1}(\lambda - \Delta)F. \tag{2.9}$$

□

This method is very similar for Dirichlet boundary conditions.

2.10 Resolvent kernels of $(\Delta - \lambda)$ in $\mathbb{R}$

When our domain is no longer compact, the Neumann Laplace operator no longer has a compact resolvent. We can say some things about it though, namely that as an operator valued function on $\mathbb{C}$, it admits a meromorphic extension to the whole complex plane. In this section we will prove this. We have followed the the work of Richard Melrose, and these results, but not the calculation, are included in his book; Geometric Scattering Theory [5]. Central to this is the theory in integral kernels, which we now introduce.

2.10.1 The Schwartz kernel theorem

This next theorem is important. We can simply observe that for two open sets X_1 and $X_2 \subset \mathbb{R}^n$, a function $K \in C^0(X_1, X_2)$ defines a linear operator ; an integral operator

$$T : C_0(X_2) :\to C(X_1),$$

by

$$(K\varphi)(x_1) = \int_{X_2} K(x_1, x_2)\varphi(x_2)$$

for $\varphi \in C_0(X_2),\ x_1 \in X_1$ [16, Page128]. The Schwartz kernel theorem allows us to extend this to distributions, where T now must act on $C_0^\infty(X_2)$, we now have

$$T : C_0^\infty(X_2) :\to \mathcal{D}'(X_1), \tag{2.10}$$

For the $\varphi \in C_0^\infty(X_2)$ and $\psi \in C_0^\infty(X_1)$, we can define the tensor product $\varphi \otimes \psi \in C^\infty(X_1 \times X_2)$, by $(\varphi \otimes \psi)(x_1, x_2) = \varphi(x_1)\psi(x_2),\ x_1 \in X, x_2 \in X_2$. We can now state that for $K \in C(X_1 \times X_2)$,

$$\langle T\varphi, \psi\rangle = K(\psi \otimes \varphi). \tag{2.11}$$

We are now ready for the following:

Theorem 48. ***Schwartz Kernel theorem***
For all $K \in \mathcal{D}'(X_1 \times X_2)$ defines a linear map as in (2.10), by way of equation 2.11, where

$$\varphi_n \to 0 \Rightarrow T\varphi_n \to 0,\ \text{as } n \to \infty,$$

Conversely, for all

$$T : C_0^\infty(X_2) \to \mathcal{D}'(X_1),$$

there is a unique, square integrable $K \in \mathcal{D}'(X_1 \times X_2)$ such that (2.11) holds.

2.10.2 Calculating resolvent kernels in $\mathbb{R}$

With the Schwartz kernel theorem proving existence and uniqueness of the kernel for our resolvent, we will go about calculating it in this section. In $\mathbb{R}$, we see that by solving the equation

$$(\Delta - \lambda)k(x, y) = \delta(x - y), \tag{2.12}$$

gives a resolvent kernel if k defines a bounded linear operator. We should remind the reader that f is some function in the domain of $(\Delta - \lambda)$, y is a parameter and δ is the Dirac delta distribution.

This is because when we apply $(\Delta - \lambda)R(\lambda)g$, for some g in the domain of the resolvent, we get

$$(\Delta - \lambda)\int_{\mathbb{R}} k(x, y)g(y)dy = \int_{\mathbb{R}} \delta(x - y)g(y)dy = g(x).$$

A general solution to equation 2.12, will be of the form,

$$A(x)e^{-i\sqrt{\lambda}(x-y)} + B(x)e^{i\sqrt{\lambda}(x-y)}.$$

Applying Δ to this gives us

$$(\lambda A + 2i\sqrt{\lambda}A' - A'')e^{-i\sqrt{\lambda}(x-y)}$$
$$(+\lambda B - 2i\sqrt{\lambda}B' - B'')e^{i\sqrt{\lambda}(x-y)}.$$

The requirement that our kernel be square integrable for λ in the upper half-plane, implies that B vanishes for negative values of $x - y$, and A for positive. Due to this we try, $A(x) = H(y - x),\ B(x) = H(x - y)$, we get

$$\begin{aligned}&\Big(\lambda H(y-x) - 2i\sqrt{\lambda}\delta(x-y) + \delta(x-y)\Big)\, e^{i\sqrt{\lambda}(x-y)}\\ +&\Big(\lambda H(x-y) - 2i\sqrt{\lambda}\delta(x-y) - \delta(x-y)\Big)\, e^{-i\sqrt{\lambda}(x-y)}.\end{aligned}$$

Meaning that the solution to $(\Delta - \lambda)k(x, y) = \delta(x - y)$ is

$$\frac{-e^{i\sqrt{\lambda}|x-y|}}{2i\sqrt{\lambda}}. \tag{2.13}$$

Note 49. *The requirement that the kernel be square integrable for* λ *in the upper half-plane was entirely arbitrary and depends on the branch choice of the square root. We could have instead chosen the lower half-plane also, to give* $\frac{-e^{-i\sqrt{\lambda}|x-y|}}{2i\sqrt{\lambda}}$ *as the kernel.*

Though unused here, it is worth mentioning that this result generalises to higher dimensions by using the same method we have employed here. On $\mathbb{R}^n$ the integral kernel will be

$$\frac{\pi}{2} i^{\frac{n-2}{2}} \lambda^{\frac{n-2}{2}} H^{(1)}_{\frac{n-2}{2}}(\lambda|x-y|),$$

where $H^{(1)}_{\frac{n-2}{2}}$ is an $\frac{n-2}{2}$ order Hankel function of the first kind [5]. The NIST digital library or handbook of mathematical functions gives information on Hankel functions [17] [18], and in Melrose's text [5], they appear as kernels of resolvents.

Example 50. *Neumann Resolvent on a half-line*
Let our domain be $\mathbb{R}_+$ *with Neumann boundary conditions at* 0*. By looking at equation 2.13, we see that the kernel of our resolvent is now*

$$\frac{-e^{i|x+y|\sqrt{\lambda}}}{4i\sqrt{\lambda}} + \frac{-e^{i|x-y|\sqrt{\lambda}}}{4i\sqrt{\lambda}}. \tag{2.14}$$

We should take note of the fact that when $x \neq y$, this kernel is holomorphic, when defined as a function of λ, in the branch of the square root where $\mathrm{Im}(\sqrt{\lambda}) > 0$. The anti-diagonal doesn't cause problems, because x and y are non-negative.

We can see, by inspection, that $(\Delta - \lambda)$ is elliptic for all $\mathbb{C}$. Theorem 44 also tells us that for λ in the resolvent set for Δ,

$$\text{sing supp } (\Delta - \lambda)^{-1} f = \text{sing supp } f.$$

Lemma 51. *The integral kernel:* $k(x, y)$ *of the Neumann Laplace operator is smooth, off the diagonal, that is when* $x \neq y$*.*

Proof. Let us apply the Laplace operator to the first and second variables of its resolvent integral kernel respectively:
We note that $(\Delta_x + \Delta_y - 2\lambda)$ is an elliptic differential operator on $M \times M$.
Now

$$(\Delta_x + \Delta_y - 2\lambda)k(x, y) = \delta(x - y),$$

which gives

$$\text{sing supp } k(x, y) = \text{sing supp } \delta(x - y);\ \text{ as a function of } x \text{ and } y.$$

This proves the lemma. □

2.11 The Scattering matrix in a general setting

The culmination of the next chapter will be a rigorous mathematical definition of the scattering matrix. All of the material beforehand will be necessary to achieve this end. Beforehand though, we will give an overview of this object.

In Reed and Simon's book on the subject, the scattering matrix appears in the context of dynamic scattering theory. In a dynamical system there are said to be given dynamics and free dynamics [1, Page 1]. This is most often a Hamiltonian with and without potential or a domain or metric with or without perturbation. From these it is possible to define positive and negative wave operators, which act on the phase space of the dynamical system, mapping a given position and velocity to the solution of the given dynamical system at time $t = 0$ which is asymptotic to the free system at $t = -\infty$ for the positive operator or $t = +\infty$ in the negative case. The scattering matrix in this context is a composition of the adjoint of the negative wave operator with the positive wave operators [1, Page8,73]. In their paper, Müller and Strohmaier have presented a proof, in English, that the scattering matrices arising from dynamic and stationary scattering theory are the same [4], a fact originally proved by Guillopé [22]. We will be dealing only with stationary scattering theory in this thesis and will introduce the scattering matrix in this context.The next section will cover in detail the stationary scattering matrix as we have chosen to define it.

At the most basic level, the scattering matrix for a waveguide describes the outcome of a scattering event in which a wave packet originating from one of the ends undergoes scattering. The aftermath of such a scattering event will result in some transmission of a proportion of the wave packet to the other ends and reflection back down the end it originated. Each such end will correspond to a column of the scattering matrix where in turn each element will correspond to an end and denote the proportion of the wave packet transmitted or reflected along it, subject to an appropriate choice of basis. We can say that the diagonal elements are the reflection coefficients and the off-diagonal elements are the transmission coefficients. As will become clear in the next section, this is greatly over simplified. In the case of Neumann waveguides, which we will be focussing on, the scattering matrix can be said to act on selected Neumann eigenfunctions of the cross sectional area of the ends. The selection of Neumann eigenfunctions will depend on the sheet of Z, meaning that the rank of the scattering matrix will also vary with this choice as opposed to just λ and the geometry of the domain.

In the case of waveguides, the scattering matrix, is indeed a true matrix. In the context of obstacle or potential scattering in $\mathbb{R}^n$ the scattering matrix becomes an operator acting on functions defined on the unit sphere.

Chapter 3

Thesis background theory

3.1 Waveguides

From now on let M be a waveguide in n dimensional Euclidian space with K cylindrical ends, a piecewise path connected subset of $\mathbb{R}^n$ that can be written as $M = E \cup X$. Here X is a compact, piecewise connected manifold, with piecewise smooth Lipschitz boundary (a Lipschitz domain) and

$$E = \Gamma \times \mathbb{R}_+ = \bigcup_{k=1}^{K} [\Gamma_k \times \mathbb{R}_+] = \bigcup_{k=1}^{K} E_k.$$

$\Gamma_k \subset \mathbb{R}^{n-1}$ is compact and connected domain, with smooth boundary, and for any $i \neq j$,

$$[\Gamma_i \times \mathbb{R}_+] \cap [\Gamma_j \times \mathbb{R}_+] = \emptyset.$$

Define

$$E_k = \Gamma_k \times \mathbb{R}_+, \quad \text{and} \quad \Gamma = E \cap X = \{0\} \times \Gamma = \bigcup_{k=1}^{K} [\{0\} \times \Gamma_k].$$

Lastly, we will call the boundary of M, Σ.

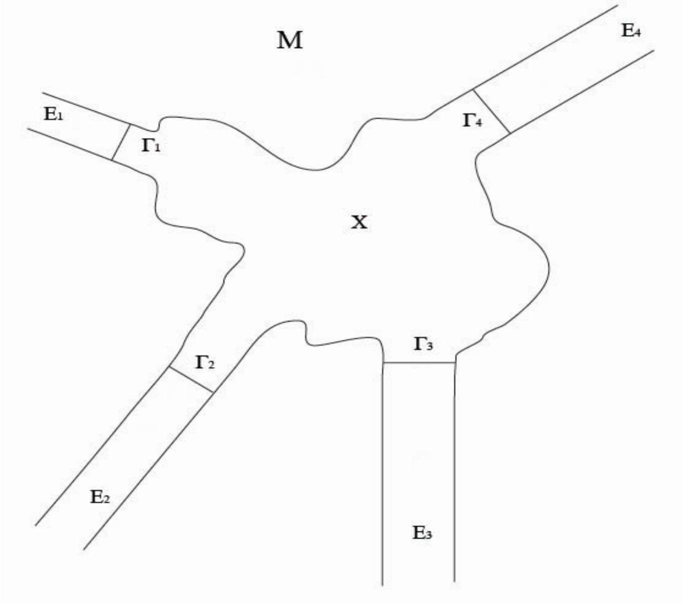


Figure 3.1: Waveguide

The following theorem is taken directly from Reed and Simon [11, p. 52,II.10]

Theorem 52. *Let $\langle M_1, \mu_1\rangle$ and $\langle M_2, \mu_2\rangle$ be measure spaces so that $L^2(M_1, d\mu_1)$ and $L^2(M_2, d\mu_2)$ are separable.*

Then there is a unique isomorphism from $L^2(M_1, d\mu_1)\hat{\otimes}L^2(M_2, d\mu_2)$ to $L^2(M_1 \times M_2, d\mu_1 \otimes d\mu_2)$, where the $\hat{\otimes}$, denotes the completion of the tensor product, so that $f \otimes g \mapsto fg$.

Corollary 53. *With E defined as above,*

$$L^2(E) \cong L^2(\mathbb{R}_+)\hat{\otimes}L^2(\Gamma).$$

Applying theorem 52 to E gives us:

Corollary 54. *With $L^2(E)$, defined as in Corollary 53, Δ_{E_j} can be written in the form*

$$\Delta_{E_j} = -\frac{\partial^2}{\partial x^2} \otimes 1 - 1 \otimes \Delta_\Gamma, \tag{3.1}$$

where Δ_Γ is the Laplace operator on $\Gamma \subset \mathbb{R}^{n-1}$.

We know that the Neumann subspace of $L^2(\Gamma)$ has an orthonormal basis consisting of Neumann eigenfuctions of $(\Delta_\Gamma - \lambda)^{-1}$, thus we can say

$$L^2(\Gamma) \cong \bigoplus_{j=0}^{\infty} \mathbb{C} = l^2.$$

We can now conclude that

$$L^2(E) \cong L^2(\mathbb{R}_+) \otimes L^2(\Gamma) \cong L^2(\mathbb{R}_+) \otimes l^2 \cong \bigoplus_{j=0}^{\infty} L^2(\mathbb{R}_+), \tag{3.2}$$

where $(\Delta_{E_j} - \lambda)$ acts on each summand by $-\frac{\partial^2}{\partial x^2} - \lambda + \mu_j$, where the μ_j are the, not necessarily distinct, Neumann eigenvalues of Δ_Γ, enumerated in ascending order and repeated with multiplicity taken into account, as needs be. We shall henceforth refer to each of the summands as **modes**. $\Delta_{E_j} - \lambda$ acts on each mode as multiplication by $\xi^2 - \lambda + \mu_j$ in the spectral representation.

The resolvent, for Δ_{E_j} written as it is in (3.1), can now be written as

$$R_0(\lambda) = \bigoplus_j r^j(\lambda), \tag{3.3}$$

where $r^j(\lambda) = \frac{1}{\xi^2 + \mu_j - \lambda}$ in the spectral representation. This means we must slightly modify our integral kernel, that we derived in the last section (equation 2.13) to get

$$r^j(\lambda) = \frac{-e^{i|x+y|\sqrt{\lambda-\mu_j}}}{4i\sqrt{\lambda-\mu_j}} + \frac{-e^{i|x-y|\sqrt{\lambda-\mu_j}}}{4i\sqrt{\lambda-\mu_j}}. \tag{3.4}$$

The existence of square roots, and their branches, adds extra complexity to this resolvent kernel. Rather than talking about these square roots as being defined on $\mathbb{C}$, we must, instead talk about them being defined on a complicated Riemann surface Z on which it is single valued function of λ. Of course $R_0(\lambda)$ only exists as a resolvent function for λ in the resolvent set, this will occur when the the $r^j(\lambda)$ are square integrable. In the next section will will describe in detail this surface in order that we may merromorphically continue or resolvent, from the resolvent set, to the rest of Z.

3.1.1 A description of the domain of our resolvent function

Up until now we have enumerated the Neumann eigenvalues $\{\mu_j\}$ of Γ in ascending order with multiplicity taken into account. It will be notationally convenient in this section, to refer to the same roots as $\{\eta_j\}$ enumerated in ascending order, but where $i \neq j \Rightarrow \eta_i \neq \eta_j$, as per Christiansen [2, Page5]. The reason for this is that for each $j \in \mathbb{N}$ corresponds to a branch point for $\sqrt{\lambda - \eta_j}$. A complete description of this has already appeared in Guillopé's paper [22], we will, nevertheless, present our interpretation here.

Before we begin our description of the domain we wish to extend our resolvent to, we will describe a far simpler one; that of the single square root function $\sqrt{\lambda}$. We can observe that for the case of a single square root function $\lambda \mapsto \sqrt{\lambda}$, if we naively plot the Imaginary component of this function, we observe a branch cut along the negative real axis; a jump discontinuity, as seen from Figure 2.

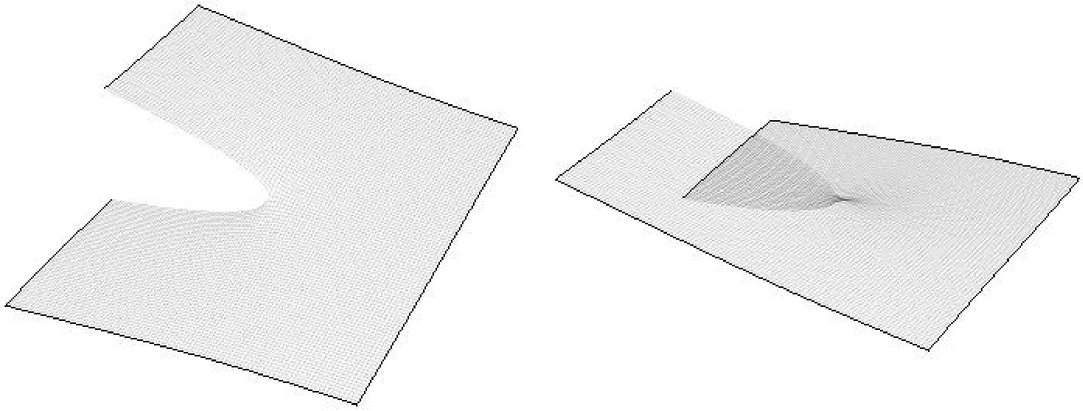


Figure 3.2: The imaginary part of the positive and negative branches of $\sqrt{\lambda}$

This is due to the fact that, for $\lambda \in \mathbb{C} \setminus \mathbb{R}$ there are two possible values for $\sqrt{\lambda}$, or branches of the function $\lambda \mapsto \sqrt{\lambda}$; the principal branch, which we have plotted in the figure above, and the negative branch, which is simply the positive branch with the sign reversed. We may observe from Figure 2 that when "glued" together, the square root will become continuous, not on $\mathbb{C}$, but on a **Riemann surface**; a path-connected, one dimensional complex manifold, that is also a covering space for the complex plane. The graph of the real component of these "glued" branches is the same as for the imaginary component, rotated a half-turn about the point 0. We can view this surface as a union of two "sheets", each identified with $\mathbb{C} \setminus \mathbb{R}_+$, where the imaginary part of $\sqrt{\lambda}$ is always positive and negative respectively. In the sheet where $\sqrt{\lambda}$ has positive real components, this is made up from the positive branch of the square root, when λ is in the upper half-plane and the negative branch when λ is in the lower.

If we were to look at the image of a point on Z under the covering projection onto $\mathbb{C} \setminus \{\eta_j\}$, we would see that when joined to another point by a path, this path and its two endpoints lift into Z. More importantly; any point in Z can be described by a point in $\mathbb{C} \setminus \{\eta_j\}$ and an equivalence class of paths to it from the base point. We will now describe these equivalence relations. Each sheet can obviously be identified with the complex plane, minus a half line, but it is the path that determines which sheet of the Riemann surface our point in $\mathbb{C} \setminus \{\eta_j\}$ lifts to. For such a simple Riemann surface this is unnecessary, but for the much more complicated Riemann surface that forms the domain of our resolvent function it provides a clear way to comprehend such a structure and this will be how we shall define it.

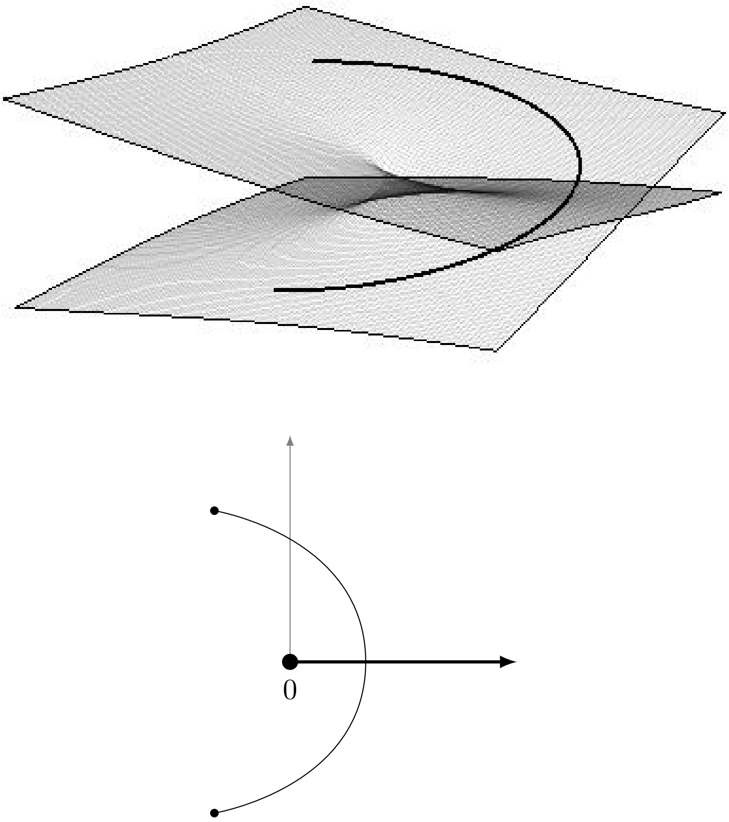


Figure 3.3: A path in our Riemann surface crossing the boundary between two sheets and its covering projection crossing the real axis

As we have seen in the last section, the resolvent is made up of the direct sum of the $r^j(\lambda)$, acting on the direct sum of L^2 spaces; equation 3.4. The Riemann surface for each individual summand that makes up this resolvent will have a branch at each of the η_j, for $j \in \mathbb{N}$, behaving similarly to the ordinary square root function. We define the **physical sheet** of Z to be the sheet of the surface, which can be identified with $\mathbb{C}\setminus\mathbb{R}_+$, for which all the $\sqrt{\lambda - \eta_j}$, have positive imaginary part and identify it with $\mathbb{C}\setminus\mathbb{R}_+$. The whole surface Z is made up of uncountably many "sheets" of this nature, each of which represents a choice as to whether each $\sqrt{\lambda - \eta_j}$ has a strictly positive imaginary part or not. To each sheet, we can associate an infinite sequence of elements in $\mathbb{Z}_2$, where the j^{th} element will be 0 if $\sqrt{\lambda - \eta_j}$ has a strictly positive imaginary part and 1 otherwise. We can see that the physical sheet will be represented by a sequence of zeros.

A path in Z will remain a path in $\mathbb{C} \setminus \{\eta_j\}$ under the covering projection p. Similarly, a path in $\mathbb{C}\setminus\{\eta_j\}$, lifts to a path in Z if the location of the pre-image of one of the endpoints is known (or given). We note for paths that paths crossing of the intervals (η_j, η_{j+1}), $j \in \mathbb{N}$ on the real line in $\mathbb{C} \setminus \{\eta_j\}$, correspond with crossings of the boundaries between sheets in Z. We aim to describe this correspondence here.

Every path of finite length, starting in the physical sheet of Z, $\gamma : I \longrightarrow Z$, and crossing the boundary between sheets at least once, can decomposed into K sub-paths γ_k, such that $\gamma_k(0) = \gamma_{k+1}(0)$, where each sub-path makes only one crossing between sheets. Under the covering projection, each $p \circ \gamma_k$ make only one crossing of the real axis. $p \circ \gamma_k$ will cross an interval (η_j, η_{j+1}) say. The consequence of this is that whatever the indexing sequence for $p \circ \gamma_k(0)$, the indexing sequence for $p \circ \gamma_k(1)$ (and consequently $p \circ \gamma_{k+1}(0)$) will have all entries up to their j^{th} entry changed (1s become 0s and vice versa). The sub-paths generate a sequence of K such objects, whose K^{th} entry describes the sheet that the endpoint of γ; $\gamma(1)$ resides in. The other entries $k = 1 \dots K$ describe the sheets of Z that γ passes through on its 'journey'. Given the correspondence between paths from the physical sheet in Z and paths in the complex plane, and our complete description of how they interact, from now on, we will refer to paths in $\mathbb{C} \setminus \{\eta_j\}$ and paths in Z interchangeably.

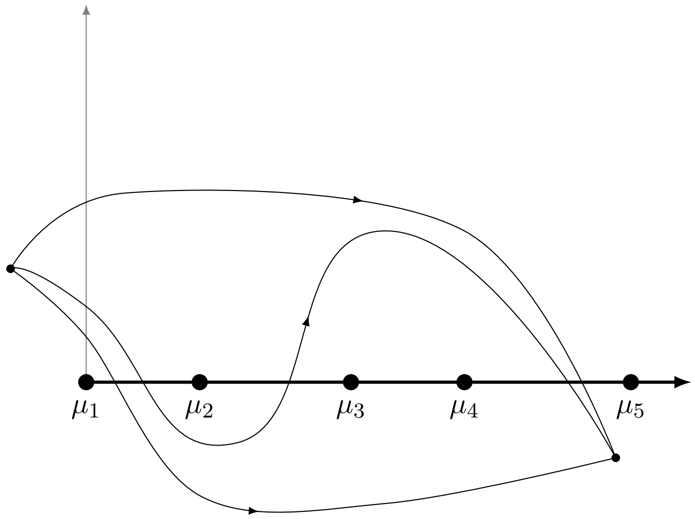


Figure 3.4: Inequivalent paths to the same point in $\mathbb{C}$

Here we see that, whilst the endpoints for all three paths are the same, all three paths lift into different sheets of Z and intersect different subsurfaces. The journey of the paths between sheets, from top to bottom, can be described by the following sequence of indexing sequences:

$$(0, \dots) \mapsto (1, 1, 1, 1, 0, \dots)$$
$$(0, \dots) \mapsto (1, 0, \dots) \mapsto (0, 1, 0, \dots) \mapsto (1, 0, 1, 1, 0, \dots)$$
$$(0, \dots) \mapsto (1, 1, 1, 1, 0, \dots)$$

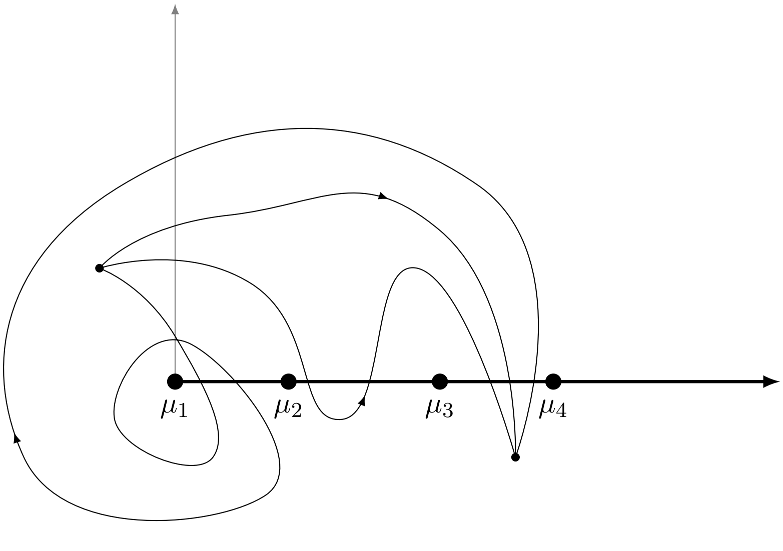


Figure 3.5: Paths that lift to the same point in Z

The paths above all map to the same point in Z, but have different indexing sequences; and we can describe their journey below:

$$(0,\ldots) \mapsto (1,1,1,0,\ldots)$$
$$(0,\ldots) \mapsto (1,1,0,\ldots) \mapsto (0,\ldots) \mapsto (1,1,1,0,\ldots)$$
$$(0,\ldots) \mapsto (1,0,\ldots) \mapsto (0,\ldots) \mapsto (1,1,1,0,\ldots)$$

We must however note that the "looped" path is not homotopic to the others, this will be something to consider in the in the next section, where we will give an overview of the monodromy theorem.

3.2 The continuation of the resolvent

The aim of this section is to prove, that for any sheet of Z, the full resolvent $R(\lambda)$ admits a meromorphic continuation from the physical sheet onto it. To do this, we must construct meromorphic continuations of the resolvent on E, $R_0(\lambda)$ beyond the physical sheet, then use "glueing constructions" and the meromorphic Fredholm theorem to obtain this same result for the resolvent $R(\lambda)$. The main techniques we shall use to prove this appear in Melrose's "Geometric Scattering theory" [5] and were used on the resolvent on a domain isometric to $\mathbb{R}^n$ outside of some compact manifold. They can be adapted to waveguides with with minimal modification, as we shall demonstrate shortly. Construction of meromorphic continuations for resolvents, for the purpose of scattering theory, has a rich history in the literature [23] [10] [5] [24].

We saw in the previous section, that Z is the surface on which the infinite direct sum of resolvent kernels for each mode $r^j(\lambda)$, is single valued. The resolvent is analytic on the physical sheet; to construct a meromorphic continuation, we must connect the physical sheet to our desired sheet with a path of finite length. Melrose and others have already demonstrated the existence of analytic continuations for the individual $r^j(\lambda)$ [5, Page 11].

We will now state, without proof, the monodromy theorem. It is not our intention to introduce the geometric topological definitions that we will occasionally make use of here or prove the complex analytic theorems that are pre-requisites to the monodromy theorem. If the reader requires such, they can be found in [25, Page 123] Jones and Singerman's text, along with more detailed discussions of Riemann surfaces and meromorphic continuations than are present here.

Lemma 55. ***Uniqueness of the direct meromorphic continuation***
Let f_1 be a meromorphic function defined on some region D_1. Let D_2 be another region, such that $D_1 \cap D_2 \neq \emptyset$, then a meromorphic f_2, defined on D_2 such that $f_1 = f_2$ on $D_1 \cap D_2$ is unique, if it exists.

The f_2, referred to in the lemma will be called a **direct meromorphic continuation of** f_1 [25, Page124].

Definition 56. ***Meromorphic continuation along a path***
Let $\gamma :\to M$ be a path in some manifold M, and f me a meromorphic function on a disc D, around $\gamma(0)$, we define the meromorphic continuation along a path to be a finite sequence of direct meromorphic continuations

$$(D, f) \sim (D_1, f_1) \sim \ldots (D_K, f_K)$$

such that each

- *each D_k is an open disc in M and $\gamma(0) \in D_1 \subseteq D$.*
- *there is a partition of the interval, $I_1, \dots, I_K$ such that $\gamma([I_k, I_{k+1}]) \subset D_k$, obviously $\gamma(1) \in D_K$.*

Theorem 57. ***The monodromy theorem***
Let γ_1, γ_2 be two homotopy-equivalent paths in Z from the physical sheet to to another, specified non-physical sheet. Also let
$\gamma_1(0) = \gamma_2(0),\ \gamma_1(1) = \gamma_2(1).$

If f is a meromorphic function in a neighbourhood of $\gamma_1(0)/\gamma_2(0)$, then, if it exists, the meromorphic continuation of f along paths γ_1 and γ_2 will be identical in a neighbourhood of $\gamma_1(1)/\gamma_2(1)$.

Since each sheet of Z is simply connected, once we have defined a path from the physical sheet to our chosen sheet and constructed a meromorphic continuation, to a neighbourhood around a point in this sheet, we can canonically extend this continuation to the whole sheet unambiguously.

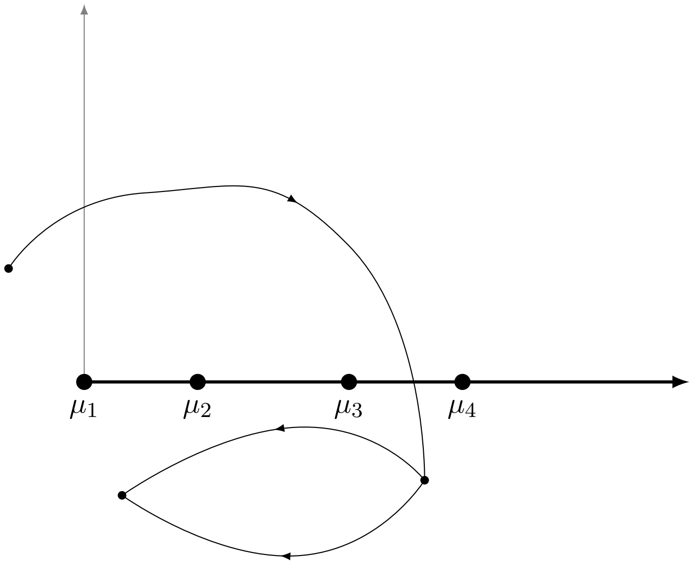


Figure 3.6: Meromorphic continuations along paths to points in the same sheet are unique.

Lemma 58. *Let $\gamma : I \to Z$ be a path of finite length from some point on the physical sheet to another point on an arbitrarily chosen non-physical sheet. The resolvent $R_0(\lambda)$ admits a meromorphic continuation along γ to a neighbourhood of $\gamma(1)$, then subsequently to the whole sheet containing it.*

Lemma 59. *Let $R_0(\lambda)$ be the meromorphic continuation of the resolvent of the Laplace operator on E along some path γ. Let us define weighted L^2 spaces,*

$$\mathcal{H}^{\pm}(\mathbb{R}_+) = L^2(\mathbb{R}_+, e^{\pm\frac{x^2}{2}} dx). \tag{3.5}$$

Then $R_0(\lambda)$ as an operator from $\mathcal{H}^+ \otimes l^2$ to $\mathcal{H}^- \otimes l^2$, is bounded.

Proof. We can see here that the meromorphic continuation of each summand that makes up R_0 is achieved canonically in this weighted space just from the definitions given in 2.14.

We can now calculate the operator norm of $R_0(\lambda)$:

$$\begin{aligned}
\|R_0(\lambda)\| =& \sup_{f\neq 0}\left(\frac{\|R_0(\lambda)f\|_{\mathcal{H}^-\otimes l^2}}{\|f\|_{\mathcal{H}^+\otimes l^2}}\right) \\
=& \sup_{f\neq 0}\left(\sup_j\left(\int_{\mathbb{R}_+}|R_0(\lambda)f_j(x)|^2 e^{\frac{-x^2}{2}}dx.\|f\|^{-1}_{\mathcal{H}^+\otimes l^2}\right)\right) \\
=& \sup_{g\neq 0}\left(\sup_j\int_{\mathbb{R}_+}\left|\int_{\mathbb{R}_+}\frac{-e^{i|x+y|\sqrt{\lambda-\mu_j}}-e^{i|x-y|\sqrt{\lambda-\mu_j}}}{4i\sqrt{\lambda-\mu_j}}g_j(y)e^{\frac{-y^2}{2}}dy\right|^2 e^{\frac{-x^2}{2}}dx\|g\|^{-1}_{L^2\otimes l^2}\right) \\
<& \sup_j\left(\int_{\mathbb{R}_+}\int_{\mathbb{R}_+}\left|\frac{-e^{i|x+y|\sqrt{\lambda-\mu_j}}-e^{i|x-y|\sqrt{\lambda-\mu_j}}}{e}^{\frac{-y^2}{2}} e^{\frac{-x^2}{2}}\right|^2 dydx\right).
\end{aligned}$$

We can see that this is finite all of the time, no matter what sheet of Z we are on, and no matter what value of j, due to the extreme weighting of these spaces. □

The cornerstone of the proof of resolvent meromorphicity is the analytic Fredholm theorem. We shall state it below bore fore attempting to show the main result.

Theorem 60. ***Meromorphic Fredholm Theorem***

Let Ω be a connected open patch of some complex manifold and $A(z)$ an operator valued (operators from a Hilbert space to itself) meromorphic function of z. In addition if:

1. *$A(z)$ is compact if $z \in \Omega \backslash D$, where D is a discrete set, outside of which A is analytic.*
2. *Coefficients $A_{-k}\cdots A_{-1}$ of the Laurent series are finite rank operators.*

Then either:

- *$1 - A(z)$ is not invertible for any z*
  or
- *$\exists D' \subset \Omega$, a discrete set such that $(1 - A(z))^{-1}$ exists if $z \notin D \cup D'$ and extends to a function analytic in $\Omega\backslash D'$, meromorphic in Ω.*

This is taken from Reed, Simon IV [26, p. 107].

We now must extend our internal domain slightly, along a tubular neighbourhood of Γ. Here we define

$$X_L = X \cup ([0, 1] \times \Gamma).$$

The requirement at the start for Γ to be piecewise smooth, path connected, and Lipschitz guarantees that this construction is possible. Let us define

$$(\Delta_{X_L} - \lambda)^{-1},$$

the inverse operator for $(\Delta_{X_L} - \lambda)$, where Δ_{X_L} is the Neumann Laplacian restricted to X_L . All of the results that apply for the resolvent of X, still apply for X_L; namely that the resolvent is a compact, self adjoint operator.

With this done, we can now move straight to the main result for this section.

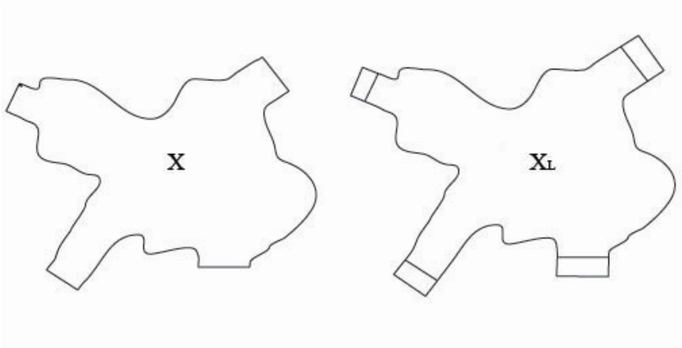


Figure 3.7: Extending the centre along the cylindrical ends by an interval.

Theorem 61. *Let $R(\lambda)$ be the resolvent for the Neumann Laplace operator on the waveguide M defined on the upper half-place of the physical sheet of Z. Let $\gamma : I \to Z$ be a path, of finite length, from the physical sheet of Z into another, arbitrarily chosen sheet. Then $R(\lambda)$ is analytic on the physical sheet of Z, and admits a meromorphic continuation, along γ to the sheet containing its endpoint on which, it may have poles that are, at most, finite order.*

Proof. Let Δ_E be the Neumann Laplacian on E, and we will assume any λ to be on the physical sheet of Z until further notice, identified with $\mathbb{C} \setminus \mathbb{R}_+$, as described in the last section.

Let us now define $\rho(a, b) \in C^\infty(\mathbb{R})$ to be a non-decreasing function with

$$\rho(a,b)(x) = \begin{cases} 1, & x \le a \\ 0, & x \ge b. \end{cases}$$

Now we can define our cut-off functions on $[0,1] \times \Gamma$, where $t \in [0,1]$ and $y \in \Gamma$:

$$\begin{aligned} \psi_1(y,t) &= \rho\left(\frac{1}{2}, \frac{3}{4}\right)(t), \quad & \phi_1(y,t) &= \rho\left(\frac{7}{9}, \frac{8}{9}\right)(t) \\ \psi_2(y,t) &= 1 - \rho\left(\frac{1}{2}, \frac{3}{4}\right)(t), \quad & \phi_2(y,t) &= 1 - \rho\left(\frac{1}{9}, \frac{2}{9}\right)(t). \end{aligned} \tag{3.6}$$

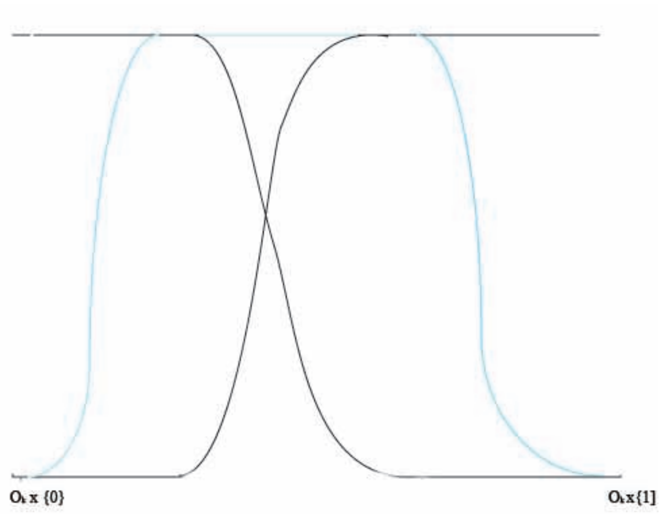


Figure 3.8: Cut-off functions

We have chosen the function ρ and its parameters in such a way that

$$\begin{aligned} \psi_1 + \psi_2 &= 1, \\ \psi_1 \phi_1 &= \psi_1, \\ \psi_2 \phi_2 &= \psi_2, \\ \operatorname{supp} \phi_k' \cap \operatorname{supp} \psi_k &= \emptyset. \end{aligned} \tag{3.7}$$

Distance is simply the length of an interval in $\mathbb{R}$ in this case. We can canonically extend these functions to the rest of M so they're constant on $X \backslash ([0,1] \times \Gamma)$; just define them to be 1 or 0 there depending on their value on the. Now define

$$Q(\lambda) = \psi_1 (\Delta_{X_L} - \lambda)^{-1} \phi_1 + \psi_2 (\Delta_E - \lambda)^{-1} \phi_2. \tag{3.8}$$

Here it it will sometimes be notationally convenient to write $(\Delta_E - \lambda)^{-1} = R(\lambda)$, the meromorphic continuation along γ of the resolvent for Δ on E.

It is at this stage where we observe that, in coordinates, with $f \in H^1(M)$, $\phi \in C^\infty(M)$ and, when defined locally, $\phi f \in H^1(M)$:

$$\Delta(\phi f) = \phi \Delta f + (\Delta \phi) f - 2 \nabla f . \nabla \phi. \tag{3.9}$$

Applying Q to $(\Delta - \lambda) f$, for some arbitrary $f \in H^1(X)$, gives:

$$\begin{aligned} & Q(\lambda)(\Delta - \lambda) f = \\ & \psi_1 (\Delta_{X_L} - \lambda)^{-1} \phi_1 (\Delta - \lambda) f \qquad (3.10) \\ & + \psi_2 (\Delta_E - \lambda)^{-1} \phi_2 (\Delta - \lambda) f. \qquad (3.11) \end{aligned}$$

In both equations 3.10 and 3.11, we see that multiplying $(\Delta - \lambda) f$ by ϕ_i maps this into the domains of the respective resolvents.

$$\begin{aligned} (3.10) &= \psi_1 (\Delta_{X_L} - \lambda)^{-1} \phi_0 \Delta f - \psi_1 (\Delta_{X_L} - \lambda)^{-1} \lambda \phi_1 f \\ &= \psi_1 (\Delta_{X_L} - \lambda)^{-1} (\Delta - \lambda)(\phi_1 f) - \psi_1 (\Delta_{X_L} - \lambda)^{-1} \left[(\Delta \phi_1) f + 2 \nabla f . \nabla \phi_1 \right]. \\ (3.11) &= \psi_2 (\Delta_E - \lambda)^{-1} \phi_2 \Delta f - \psi_2 (\Delta_E - \lambda)^{-1} \lambda \phi_2 f \\ &= \psi_2 (\Delta_E - \lambda)^{-1} (\Delta - \lambda)(\phi_2 f) - \psi_2 (\Delta_E - \lambda)^{-1} \left[(\Delta \phi_1) f + 2 (\nabla f . \nabla \phi_2) \right]. \end{aligned}$$

Adding these up, and taking note of equation 3.7 gives us:

$$\begin{aligned} & f - \psi_1 (\Delta_{X_L} - \lambda)^{-1} \left[(\Delta \phi_1) f + 2 (\nabla f . \nabla \phi_1) \right] \\ & - \psi_2 (\Delta_E - \lambda)^{-1} \left[(\Delta \phi_2) f + 2 (\nabla f . \nabla \phi_2) \right]. \end{aligned}$$

So we conclude that

$$Q(\lambda)(\Delta - \lambda) = \mathrm{Id} - J(\lambda), \tag{3.12}$$

where we have defined

$$J(\lambda) = \psi_1 (\Delta_{X_L} - \lambda)^{-1} \left[(\Delta \phi_1) + 2 (\nabla \phi_1 . \nabla) \right] + \psi_2 (\Delta_E - \lambda)^{-1} \left[(\Delta \phi_2) + 2 (\nabla \phi_2 . \nabla) \right]. \tag{3.13}$$

We will call the first and second summands in (3.13) J_1, and J_2 respectively. Since J and Q are made up of resolvents, which admit meromorphic continuations along γ, Q and J inherit these properties and also admit meromorphic continuations along γ. Of course for λ in a non-physical sheet equation 3.12 will fail, but the continuations of Q and J nevertheless exist.

Observe that when $\psi_2(\Delta_E - \lambda)^{-1}\left[(\Delta\phi_2) + 2(\nabla\phi_2.\nabla)\right]$, acts on a function f in its domain, it will be of the form

$$\psi_2(x)\int_{\mathbb{R}_+} k(x,y)\left[(\Delta\phi_2) + 2(\nabla\phi_2.\nabla)\right] f(y)dy,$$

$$= \int_{\mathbb{R}_+} \psi_2(x)k(x,y)\left[(\Delta\phi_2(y)) + 2(\nabla\phi_2(y).\nabla f(y))\right] dy,$$

$$= \int_{\mathbb{R}_+} \psi_2(x)k(x,y)(\Delta\phi_2(y))f(y))dy + \nabla.\left[2\psi_2(x)k(x,y)(\nabla\phi_2(y)\right] f(y))dy$$

by using integration by parts. Now, the only time that $k(x,y)$ may fail to be smooth is when $y = x$, by elliptic boundary regularity. But the ψ_2 and ϕ_2 have been specifically constructed so that the supports of ψ_2 and ϕ_2' are disjoint, so in this case, the integrand vanishes and we have a smoothing operator. This means that the left hand side of the above equation is well defined and still smooth.

We have thus constructed a smooth integral kernel $\tilde{k}(x,y)$, for

$$J_2 = \psi_2(\Delta_E - \lambda)^{-1}\left[(\Delta\phi_2) + 2(\nabla\phi_2.\nabla)\right],$$

given by

$$\tilde{k}(x,y) = \psi_2(x)k(x,y)(\Delta\phi_2(y))f(y))dy + \nabla.\left[2\psi_2(x)k(x,y)(\nabla\phi_2(y)\right],$$

which is compactly supported in the y variable and where it's support in the x and y variables is disjoint, by choice of cut-off functions: see equation 3.7. Now since the integral Kernel of the adjoint is compactly supported in the first variable, the adjoint to $\psi_2(\Delta_E - \lambda)^{-1}\left[(\Delta\phi_2) + 2(\nabla\phi_k.\nabla)\right]$ is compact, we see that the operator itself is compact. Thus

$$\psi_2(\Delta_E - \lambda)^{-1}\left[(\Delta\phi_2) + 2(\nabla\phi_k.\nabla)\right],$$

is compact.

We observe that

$$\left(\psi_1(\Delta_{X_L} - \lambda)^{-1}\left[(\Delta\phi_1) + 2(\nabla\phi_1.\nabla)\right]\right)^* : L(X_L)^2 \longrightarrow H(X_L)^2 \longrightarrow H(X_L)^1,$$

and is likewise compact by the Rellich-Kondrachov theorem, meaning that J_1 is also compact. We can now apply the meromorphic Fredholm theorem [26, Page 107], to $(\mathrm{Id}+J(\lambda))$, but we must make sure that the first case of the meromorphic Fredholm theorem applies. The first case being that $(\mathrm{Id} + J(\lambda))$ is invertible outside of some discrete set, not that it isn't invertible anywhere.

Now $J(\lambda) : \mathcal{H}^- \longrightarrow \mathcal{H}^-$, because $\left[(\Delta\phi_1) + 2(\nabla\phi_1.\nabla)\right] : \mathcal{H}^- \longrightarrow \mathcal{H}^+$.

Now we can see that

$$\|J(\lambda)\|_{\mathcal{H}_-} = C_1\|J(\lambda)\|_{L_2},$$

for some $C_1 > 0$. This is due to the fact that $J(\lambda)$ maps to functions with compact support; everything is multiplied by either $\nabla\phi_1, \nabla\phi_2$, $\Delta\phi_1$ or $\Delta\phi_2$, all of which have compact support. Now

$$\|J(\lambda)\|_{L_2} \leq \sum_{i=1}^{2} \|J_i(\lambda)\|_{L^2},$$

by the triangle inequality.

We must now deal with J_1 and J_2 separately again. Let $P_i = [(\Delta\phi_i) + 2(\nabla\phi_i.\nabla)]$; we can see that this is a differential operator of order one. For J_1 we can now rewrite it as

$$J_1(\lambda) = \psi_1(\Delta_{X_L} - \lambda)^{-1}(\Delta_{X_L}+1)^{1/2}(\Delta_{X_L}+1)^{-1/2}P_1.$$

and now we can apply the triangle inequality once again to see that

$$\|J_1(\lambda)\| \leq \|\psi_1(\Delta_{X_L} - \lambda)^{-1}(\Delta_{X_L}+1)^{1/2}\|.\|(\Delta_{X_L}+1)^{-1/2}P_1\|.$$

Now the action of $(\Delta_{X_L} - \lambda)^{-1}(\Delta_{X_L}+1)^{1/2}$ on a basis of eigenfunctions will be that of multiplication by

$$\frac{\sqrt{\lambda_n^2+1}}{\lambda_n^2 - \lambda}, \tag{3.14}$$

where the $\lambda_n \in \mathbb{R}_+$ are the eigenvalues of their respective eigenfunctions for $n = 1, \ldots, \infty$. We can now write $\lambda = -R^2,\ R > 0$, meaning that $\sqrt{\lambda} = iR$. So equation3.14 can now be rewritten as

$$\frac{\sqrt{\lambda_n+1}}{(\lambda_n^2 - iR)(\lambda_n^2 + iR)} = \frac{\sqrt{\lambda_n+1}}{(\lambda_n^2+R^2)^{1/2}(\lambda_n^2+R^2)^{1/2}} \leq \frac{\sqrt{\lambda_n+1}}{(\lambda_n^2+R^2)^{1/2}(\lambda_n^2+R^2)^{1/2}} \leq \frac{1}{R}.$$

The same argument holds true for $(\Delta_E - \lambda)^{-1}(\Delta_E+1)^{1/2}$ using the spectral resolution of the operator Δ_E.

This means that we can pick an appropriate value of λ to give $(\mathrm{Id} + J(\lambda))$ a Neumann series. Consequently $(\mathrm{Id} + J(\lambda))$ is invertible outside of some discrete set and,

$$(\mathrm{Id} + J(\lambda))^{-1}Q(\lambda)(\Delta - \lambda) = \mathrm{Id},$$

so

$$(\mathrm{Id} + J(\lambda))^{-1}Q(\lambda)$$

is a resolvent for $(\Delta - \lambda)$, when λ is in the physical sheet and has a meromorphic continuation along γ. □

3.3 Generalised Eigenfunctions and the Scattering Matrix

Let χ be a function on M with support on E that equals 1 outside a compact set and fix an orthonormal basis of Neumann eigenfunctions of Δ_Γ, $\{\nu_j(y)\}$. Let γ be a path from the physical sheet to another, non-physical sheet in Z of our choice. To describe this sheet we must introduce $J \subset \mathbb{N}$ given by:

$$J = \{j \in \mathbb{N} : \mathrm{Im}(\sqrt{\lambda - \mu_j}) \leq 0\}. \tag{3.15}$$

We refer the reader back to section 3.1, where we equated sheets of Z to equivalence classes of paths. We can consider J to be an indexing set for the sequences defined there. For example the J for the three examples defined in figure 3.5 would be $\{1, 2, 3\}$. This notation, whilst differing slightly from that introduced before, is ideally suited for use later on.

As was said before, we may identify this sheet with $\mathbb{C} \setminus \min_{j\in J}[\mu_j, \infty)$. We will generally be working with either the physical sheet of Z, or the sheet defined by J, which we shall refer to as the non-physical sheet from now on. When identified in this way, every λ in the non-physical sheet of Z has its counterpart in the physical sheet which is identical as a complex number. As each sheet is simply connected, we may extend γ to λ in only one homotopy-equivalent way, meaning that once the resolvent is continued, meromorphically along γ, we can uniquely continue it to a neighbourhood of any point on the same sheet, without ambiguity. Thus, once we fix a sheet, denoted by equation 3.15, and extend our resolvent along a path to it from the physical sheet, we can assume that the resolvent is meromorphic on the whole sheet.

We should make the reader aware of the fact that, from now on, we will be enumerating the transverse Neumann eigenvalues, with multiplicity taken into account; μ_j, not to be confused with η_j from before when multiplicity was ignored.

We are now ready to to introduce:

$$\varphi(\lambda, x, y) = \sum_j \varphi^j(\lambda, x, y) = \sum_j \chi e^{-i\sqrt{\lambda-\mu_j}x}\nu_j(y) - R(\lambda)\left[(\Delta-\lambda)\left(\chi e^{-i\sqrt{\lambda-\mu_j}x}\nu_j(y)\right)\right]. \tag{3.16}$$

This is a **Generalised Eigenfunction**, as it is a solution of $(\Delta - \lambda)\varphi(\lambda, x, y) = 0$ (we can verify this, just by applying $(\Delta - \lambda)$ to it) but not an L^2 solution when λ is in the physical sheet. It should be noted that the above equation is not already zero. The reason for this is that $R(\lambda)$, unless λ is on the physical sheet, is a meromorphic continuation of the resolvent and not the resolvent itself. When λ is in the physical sheet, the resolvent is only an inverse of $(\Delta - \lambda)$ for L^2 functions. These functions have a number of properties:

Proposition 62. *1. $\varphi(\lambda, x, y)$ is a meromorphic function of λ for any λ in the non physical sheet indexed by J and holomorphic if λ is in the physical sheet.*

2. For $j \in J$ and λ in the physical sheet of Z;

$$\varphi^j(\lambda, x, y) - \chi e^{-i\sqrt{\lambda-\mu_j}x}\nu_j(y) \in L^2(\mathbb{R}_+) \tag{3.17}$$

3. There exists a unique meromorphic $S_{j,k}(\lambda)$, such that on E and with λ in the physical sheet

$$\begin{aligned}\varphi(\lambda, x, y) = &\sum_{j\in J}\left(e^{-i\sqrt{\lambda-\mu_j}x}\nu_j(y) + \sum_{k\in J} S_{j,k}(\lambda)e^{i\sqrt{\lambda-\mu_k}x}\nu_k(y)\right) \\ &+\sum_{j\notin J} T_j(\lambda)e^{i\sqrt{\lambda-\mu_j}x}\nu_j(y),\end{aligned} \tag{3.18}$$

and

$$\begin{aligned}\varphi(\lambda, x, y) = &\sum_{j\in J}\left(e^{i\sqrt{\lambda-\mu_j}x}\nu_j(y) + \sum_{k\in J} S_{j,k}(\lambda)e^{-i\sqrt{\lambda-\mu_k}x}\nu_k(y)\right) \\ &+\sum_{j\notin J} T_j(\lambda)e^{i\sqrt{\lambda-\mu_j}x}\nu_j(y),\end{aligned} \tag{3.19}$$

for λ in the non-physical sheet. In the case where λ is in the physical sheet, the $S_{j,k}(\lambda)$ are holomorphic.

Proof.

1. Obvious: It follows from the meromorphicity of the various functions whose products make up $\varphi(\lambda, x, y)$.
2. This is due to the square integrability of the resolvent kernels in the physical sheet.
3. Observe that for any $j \in J$, the $\varphi^j(\lambda, x, y)$ is a solution to $(\Delta - \lambda)\varphi^j(\lambda, x, y) = 0$. Using a simple separation of variables on E, we see that any solution to $(\Delta - \lambda)F(\lambda, x, y) = 0$, including the one we have, will be of the form,

$$\varphi^j(\lambda, x, y) = \sum_{k=1}^{\infty} \left(A_{j,k}(\lambda) e^{-i\sqrt{\lambda-\mu_k}x} + B_{j,k}(\lambda) e^{+i\sqrt{\lambda-\mu_k}x} \right) \nu_k(y). \tag{3.20}$$

We can see that in part 2) of this proposition, the requirement that $\varphi^j(\lambda, x, y) - \chi e^{-i\sqrt{\lambda-\mu_j}x}\nu_j(y) \in L^2(\mathbb{R}_+)$, means that $A_{j,k} = \delta_{j,k}$.

In order to reconcile equation 3.20 with equation 3.16, we call the $B_{j,k}(\lambda) = S_{j,k}(\lambda)$ for $k \in J$. For $k \notin J$ we see that the remaining terms;

$$B_{j,k}(\lambda) e^{+i\sqrt{\lambda-\mu_k}x} \tag{3.21}$$

are all square integrable as $k \notin J \Rightarrow \mathrm{Im}(\sqrt{\lambda - \mu_k}) > 0$.

When we sum the $\varphi^j(\lambda, x, y)$ to get $\varphi(\lambda, x, y)$, as described in equation 3.16, we get

$$\sum_{j\in J} \left(e^{-i\sqrt{\lambda-\mu_j}x}\nu_j(y) + \sum_{k\in J} S_{j,k}(\lambda) e^{i\sqrt{\lambda-\mu_k}x}\nu_k(y) \right) + \sum_{k\notin J}\sum_{j\in J} B_{j,k}(\lambda) e^{i\sqrt{\lambda-\mu_k}x}\nu_k(y),$$

where we can define, for $j \notin J$, $T_j(\lambda) = \sum_{k\in J} B_{k,j}(\lambda)$ to finish. The result for the non-physical sheet is due to meromorphic continuation.

□

We will now call $S(\lambda)$, made up from the $S_{j,k}(\lambda)$, the **scattering matrix**. As has been mentioned in section 2.11: Many papers have been written which define and make use of the scattering matrix, and our definition is by no means unique [2, Page 899] [19] [20, Page 31] [21, Page 248].

We will now prove some useful properties about the generalised eigenfuctions and scattering matrix.

Proposition 63. ***Uniqueness of the Generalised Eigenfunctions***
Any $\varphi(\lambda, x, y)$, abbreviated as φ_λ, where $(\Delta - \lambda)\varphi_\lambda = 0$ and that obeys 1) and 2) of Proposition 62 is unique in $L^2(\mathbb{R}_+)$.

Proof. We have already defined φ_λ, so let $\widetilde{\varphi}_\lambda$ be some other generalised eigenfunction of $(\Delta-\lambda)$ on M, obeying 1 and 2 of Proposition 62 and let us start on the physical sheet of Z.

We now know that $(\Delta - \lambda)(\varphi_\lambda - \widetilde{\varphi}_\lambda) = 0$.

Given that $\widetilde{\varphi}_\lambda$ and φ_λ both obey 1) and 2) we can see that $(\varphi^j_\lambda - \widetilde{\varphi}^j_\lambda) \in L^2(\mathbb{R}_+)$, $\forall j \in \mathbb{N}$.

This means that λ is an eigenvalue and $(\varphi_\lambda - \widetilde{\varphi}_\lambda) \in L^2(\mathbb{R})$ is an eigenfunction, not a generalised eigenfunction. However Δ is a positive self adjoint operator, so all of its eigenvalues must be positive and real. Thus contradicts with λ being in the physical sheet, which was identified with $\mathbb{C}\backslash\mathbb{R}_+$ in section 8. This implies that on the physical sheet, $(\varphi_\lambda - \widetilde{\varphi}_\lambda) = 0$, then by continuation, it is 0 on the rest of Z.

□

Corollary 64. *$S(\lambda)$ is also uniquely determined by the geometry of M the choice of basis for the J Neumann eigenfunctions and J.*

Let λ be in the non-physical sheet of Z, given by J and identified with a suitable subset of the complex plane, and let λ^* be its counterpart in the physical sheet.

Theorem 65. $S(\lambda^*) = \overline{S^{-1}(\lambda)}$

Proof. We can see that

$$\langle \Delta\varphi(\lambda, x, y), \varphi(\overline{\lambda}, x, y)\rangle - \langle \varphi, \Delta\varphi\rangle = (\overline{\lambda} - \overline{\lambda})\langle \varphi, \varphi\rangle = 0.$$

The right hand side of the equation above, namely the $(\overline{\lambda} - \overline{\lambda})$ is not an issue, as λ is simply a number in this context and the $\overline{\lambda}$ will be in the same sheet of Z as λ.

Green's second identity can now be invoked to give that

$$\int_\Gamma \left(\frac{\partial\varphi(\lambda^*, x, y)}{\partial n} \overline{\varphi(\lambda, x, y)} - \varphi(\lambda^*, x, y) \frac{\partial \overline{\varphi(\lambda, x, y)}}{\partial n} \right) = 0.$$

In particular, this means that

$$\begin{aligned}
&\sum_{j,k\in J} i\sqrt{\lambda^* - \mu_k}(\delta_{k,j} - S_{j,k}(\lambda^*)) \sum_{l,m\in J} \left(\delta_{l,m} + \overline{S_{m,l}(\overline{\lambda})}\right) \int_\Gamma \nu_k(y)\nu_l(y) \\
&- \sum_{j,k\in J} (\delta_{k,j} + S_{j,k}(\lambda^*)) \sum_{l,m\in J} -i\sqrt{\lambda - \mu_l} \left(\overline{S_{m,l}(\lambda)} - \delta_{l,m}\right) \int_\Gamma \nu_k(y)\nu_l(y) = \\
&\sum_{j,k\in J} i\sqrt{\lambda^* - \mu_k}(\delta_{k,j} - S_{j,k}(\lambda^*)) \sum_{l,m\in J} \left(\delta_{l,m} + \overline{S_{m,l}(\overline{\lambda})}\right) \int_\Gamma \nu_k(y)\nu_l(y) \\
&- \sum_{j,k\in J} (\delta_{k,j} + S_{j,k}(\lambda^*)) \sum_{l,m\in J} i\sqrt{\lambda^* - \mu_l} \left(\overline{S_{m,l}}(\lambda) - \delta_{l,m}\right) \int_\Gamma \nu_k(y)\nu_l(y) = 0
\end{aligned}$$

This implies that

$$\sum_{k,j,m\in J} \Big((S_{j,k}(\lambda^*) - \delta_{k,j})(\delta_{k,m} + \overline{S_{m,k}(\overline{\lambda})}) + (\delta_{k,j} + S_{j,k}(\lambda^*))(\overline{S_{m,k}(\overline{\lambda})} - \delta_{k,m}) \Big) = 0$$

due to the orthonormality of the $\{\nu_j\}$ and the way in which we have identified our sheet of Z with a $\mathbb{C}$. Multiplying these out and simplifying shows that, for fixed $j, m \in J$, we get

$$\sum_{k\in J} S_{j,k}(\lambda^*)\overline{S_{m,k}(\overline{\lambda})} = \delta_{j,m}.$$

Since $S(\lambda)$ and $S(\overline{\lambda})$ are, by construction, the same when λ and $\overline{\lambda}$ are in the same sheet of Z the result follows.

□

3.4 The Neumann to Dirichlet map

As usual, let X be a Lipschitz domain in $\mathbb{R}^n$, $n \geq 2$.
We define the map

$$N : H^{-1/2}(\partial X) \to H^{1/2}(\partial X)$$

acting on $g \in H^{-1/2}(\partial X)$ by

$$\mathcal{D}g = \varphi_{|\partial X},$$

where φ is the solution to the Neumann problem, with g as the boundary derivative. This is the inverse of the Dirichlet to Neumann map

$$\mathcal{D} : H^{1/2}(X) \to H^{-1/2}(X),$$

whose action on $f \in H^{1/2}$ is $\mathcal{D}f = \frac{\partial}{\partial n}(HF)$, where H is an extension of f to a solution of $(\Delta - \lambda)(Hf) = 0$ on X. [27]

3.5 Calculating the Neumann to Dirichlet map

The Neumann to Dirichlet map will be a vital intermediate step between the resolvents, which we have extensively covered, and the scattering matrix. In our case it is significant, because it can be easily numerically computed using finite element techniques on the internal domain, and is known for the "ends".

It is known that the Neumann eigenvectors of $(\Delta - \lambda)$ on Γ form an orthonormal basis of $L^2(\Gamma)$, with Neumann eigenvalues μ_j, $j \in \mathbb{N}$. [12] Given a basis of $L^2(\Gamma)$, we may compute the Neumann to Dirichlet map in matrix form. This is the first step towards viewing that Neumann to Dirichlet map as a concrete, computable object. Given the correspondence between the Neumann to Dirichlet map, the scattering matrix and the resolvent, computing the Neumann to Dirichlet map in this way allows us to realise these other objects in a similar manner.

Definition 66. ***Neumann to Dirichlet map, associated to $(\Delta - \lambda)$ on a basis***
Let us consider an ordered orthonormal basis of the Neumann subspace of $L^2(\Gamma)$, $\{\nu_j\}_{j=0}^{\infty}$, and φ_k, such that

$$(\Delta - \lambda)\varphi_k = 0, \ \frac{\partial \varphi_k}{\partial n}|_{\partial X \setminus \Gamma} = 0, \ \frac{\partial \varphi_k}{\partial n}|_{\Gamma} = \nu_k. \tag{3.22}$$

Then the k, l^{th} element of the Neumann to Dirichlet map, in matrix form, with respect to basis $\{\nu_k\}$, will be given by

$$\langle \varphi|_{\Gamma}, \nu_l \rangle_{L^2(\Gamma)}.$$

Obviously, when doing this calculation practically, we must truncate this after a finite number of entries; say $M \in \mathbb{N}$, to get an $M \times M$ matrix, giving the first M rows and M columns of the infinite dimensional matrix.

We will now present some examples of Neumann to Dirichlet maps calculated, explicitly and manually for very simple internal domains. As well as giving the reader a clearer picture of the nature of this map, they are an important tool to check for errors and accuracy of any algorithm able to calculate these for more complicated internal domains.

Example 67. ***A rectangle in $\mathbb{R}^2$ with two ends***
Let the width of the two ends be denoted by w and the length of the domain l. The ends Γ_1 and Γ_2 will be the sets $\{l\} \times [0, w]$ and $\{0\} \times [0, w]$ respectively. An orthonormal basis of Neumann eigenfunctions for Γ will be or the form $\{cos(\frac{k\pi}{w}y)\}$, $k = 1 \ldots \infty$ on each

connected component of Γ *(or end of* X*) in turn (it will be* 0 *on the others). Without loss of generality, we will perform this calculation for* $\frac{\partial\varphi}{\partial n}|_{\Gamma_1} = cos(\frac{k\pi}{w}y)$*, some* $k \in \mathbb{N}$ *: it will be the same the other way around anyway.*
Solutions to equation 3.22 will now be linear combinations of

$$\cos\left(\sqrt{\lambda - \left(\frac{k\pi}{w}\right)^2}x\right)\cos\left(\frac{k\pi}{w}y\right). \tag{3.23}$$

Due to our choice of Neumann boundary condition for Γ_1*, our solution restricted to there becomes*

$$\frac{\cos\left(\sqrt{\lambda - \left(\frac{k\pi}{w}\right)^2}l\right)}{\sqrt{\lambda^2 - \left(\frac{k\pi}{w}\right)^2}\sin\left(\sqrt{\lambda^2 - \left(\frac{k\pi}{w}\right)^2}l\right)}\cos\left(\frac{k\pi}{w}y\right), \tag{3.24}$$

but on Γ_2 *it is*

$$\frac{1}{\sqrt{\lambda - \left(\frac{k\pi}{w}\right)^2}\sin\left(\sqrt{\lambda^2 - \left(\frac{k\pi}{w}\right)^2}l\right)}\cos\left(\frac{k\pi}{w}y\right). \tag{3.25}$$

As we will be truncating after M *Neumann eigenvalues,or modes for each end, we can see here that for this very simple domain, the Neumann to Dirichlet map in matrix form will be a bi-diagonal* $2M \times 2M$ *matrix whose diagonal entries will be that of the coefficients from equation 3.24, the half-diagonal entries (*$k+M^{th}$ *row ,*k^{th} *column or* $k-M^{th}$ *row* k^{th} *column; whichever is defined) will be made up of the coefficients from 3.24. This ordering of elements has come about due to it being the most natural way to calculate the Neumann to Dirichlet map in FreeFem.*

Example 68. ***A cuboid in*** $\mathbb{R}^3$ ***with two identical rectangular ends***
This is practically the same as in the previous example, though we will provide it here as it is useful for us when testing out concrete computations and it also provides the enumeration of the basis elements which will be useful later on. Let us define our cuboid to be of length l *width* w *and height* h*, with homogeneous Neumann boundary conditions on the for outside surfaces*
$\{0,y,z\},\{w,y,z\},\{x,0,z\},\{x,h,z\}$*, where* x,y,z *run from 0 to* $w,h,$ *and* l *respectively. Our ends* Γ_1 *and* Γ_2 *will be the two rectangles* $\{x,y,l\}$ *and* $\{x,y,0\}$*. We first impose homogeneous Neumann boundary conditions on the boundary of the domain, excluding* Γ_1 *and* Γ_2*.*

To compute this map; truncated and in matrix form, we must first define a basis of eigenfunctions, which span the space of functions satisfying Neumann boundary conditions on the boundary of the two 'end' rectangles. The obvious choice for this is $\{\cos(\frac{j\pi}{w}x)\cos(\frac{k\pi}{h}y)\}$*,* $j,k \in \{0,1,2\ldots\}$ *on* Γ_1 *and* 0 *on* Γ_2 *then vice vera. Once again; for this calculation, it will be sufficient to only consider inhomogeneous boundary conditions on* Γ_1 *for the same reasons as described in the previous example. We will truncate after* $\frac{N(N+1)}{2}$ *elements, and enumerate in a diagonal pattern as follows:*

$$\begin{aligned}&1, \cos(\frac{\pi}{w}x), \cos(\frac{\pi}{h}y), \cos(\frac{2\pi}{w}x), \cos(\frac{\pi}{w}x)\cos(\frac{\pi}{h}y), \cos(\frac{\pi}{h}y), \ldots\\ &\ldots, \cos(\frac{\pi}{w}x)\cos(\frac{(N-1)\pi}{h}y), \cos(\frac{N\pi}{h}y).\end{aligned}$$

We set $\frac{\partial\varphi}{\partial n}|_{\Gamma_1} = \cos(\frac{j\pi}{w}x)\cos(\frac{k\pi}{h}y)$*, for some* $j, k \in \{0, \ldots, N\}$*.*

Solutions to equation 3.22 will now be linear combinations of

$$\cos\left(\sqrt{\lambda - \left(\frac{j\pi}{w}\right)^2 - \left(\frac{j\pi}{h}\right)^2} z\right)\cos\left(\frac{j\pi}{w}x\right)\cos\left(\frac{k\pi}{h}y\right). \tag{3.26}$$

Due to our choice of Neumann boundary condition for Γ_1*, our solution restricted to there becomes*

$$\frac{\cos\left(\sqrt{\lambda - \left(\frac{j\pi}{w}\right)^2 - \left(\frac{j\pi}{h}\right)^2} l\right)}{\sqrt{\lambda^2 - \left(\frac{j\pi}{w}\right)^2 - \left(\frac{j\pi}{h}\right)^2}\sin\left(\sqrt{\lambda^2 - \left(\frac{j\pi}{w}\right)^2 - \left(\frac{j\pi}{h}\right)^2} l\right)}\cos\left(\frac{j\pi}{w}x\right)\cos\left(\frac{k\pi}{h}y\right), \tag{3.27}$$

but on Γ_2 *it is*

$$\frac{\cos\left(\frac{j\pi}{w}x\right)\cos\left(\frac{k\pi}{h}y\right)}{\sqrt{\lambda - \left(\frac{j\pi}{w}\right)^2 - \left(\frac{j\pi}{h}\right)^2}\sin\left(\sqrt{\lambda^2 - \left(\frac{j\pi}{w}\right)^2 - \left(\frac{j\pi}{h}\right)^2} l\right)}. \tag{3.28}$$

As we have been careful to choose the the basis elements of our boundaries to be orthogonal to each other, (except in the case of the "same" basis elements on the opposite boundary). We can see that, in matrix form, with said basis, the column of our Neumann to Dirichlet map corresponding to basis element $\cos((\frac{j\pi}{w})x)\cos((\frac{j\pi}{h})y)$ *will be* 0*, except on the diagonal, where it will be the coefficient of (3.27) and the half-diagonal where it will be the coefficient of (3.28), mapping to the same basis element, on the opposite end in exactly the same way as in the previous example.*

Example 69. ***A cylinder in*** $\mathbb{R}^3$ ***with two identical circular ends***
To start off, we will let the length of our cylinder be l*, and the radius of the circles* R*. This calculation is similar for the cylinder, except we will be working in spherical polar co-ordinates. We must choose a (preferably orthonormal) basis as before, truncate then enumerate its elements. We have our change of co-ordinates*

$$\begin{aligned} x &\mapsto r\cos(\theta) \\ y &\mapsto r\sin(\theta) \\ z &\mapsto z. \end{aligned}$$

Our Laplacian is now $-\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial}{\partial r}) - \frac{1}{r^2}\frac{\partial^2}{\partial\theta^2} - \frac{\partial\varphi}{\partial z^2} - \lambda^2$*. On the two end discs, an orthonormal basis of Neumann eigenfunctions will take the form*

$$J_n\left(\frac{Z'_{n,m}}{R}r\right)\sin(n\theta),\quad J_n\left(\frac{Z'_{n,m}}{R}r\right)\sin(n\theta),$$

where $n, m \in \mathbb{N}, r \in \mathbb{R}$*. Here we can see that the* J_n *are Bessel functions of the first kind and the* $Z'_{n,m}$ *are the* m^th *roots of their first derivatives with respect to* r*. As in the previous example, we must truncate this infinite series. For some* $M \in \mathbb{N}$*, we choose let* m, n *run between* 0 *and* M *giving us a square matrix where each row and column has* $\#ends.(2M^2 - M)$ *elements. This is because we get a cos eigenfunction for every* n*, but a sin function only for non-zero* n*, and this is repeated for all roots* m*. To calculate this, we must pick a basis element, for some* m, n*, an end* Γ_1 *say, where* $z = L$*, then solve the pde. on* X*, with the homogeneous Neumann boundary conditions as before. Now observe that our eigenfunctions will be as follows:*

$$\cos\left(\sqrt{\lambda - \left(\frac{Z_{n,m}}{R}\right)^2} z\right) J_n\left(\frac{Z_{n,m}}{R}r\right)\cos(n\theta)/\sin(n\theta)$$

Setting the Neumann boundary condition at $z = L$ to be

$$j_n\left(\frac{Z_{n,m}}{R}r\right)\cos(n\theta)/\sin(n\theta)$$

yields

$$\sqrt{\lambda - \left(\frac{Z_{n,m}}{R}\right)^2}\cot\left(\sqrt{\lambda^2 - \left(\frac{Z_{n,m}}{R}\right)^2}L\right)J_n\left(\frac{Z_{n,m}}{R}r\right)\cos(n\theta)/\sin(n\theta)$$

as Dirichlet data.

3.5.1 The Levitin-Marletta method for indirect calculation of the Neumann to Dirichlet map

This has the advantage in that there is only one calculation (the division) involving λ, which ought to produce faster computation of the Neumann to Dirichlet map, and subsequent scattering matrix, for any given λ. All of the other data can be pre-calculated and "used again" for any λ. The following formula for the k, l^{th} entry of N acting on basis of the Neumann subspace of $L^2(\Gamma)$, $\{\nu_i\}_{i=1}^{\infty}$, is taken from the paper by Levitin and Marletta, where it is calculated. [7, Page 11] The $\{\nu_i\}_{i=1}^{\infty}$ remain the same as in the examples above and in definition 66; trigonometric functions in the two dimensional case, products of trigonometric functions in higher dimensional rectangular cases, Bessel and trig functions in truly cylindrical cases and for ends of arbitrary cross section, they could themselves be approximated by the finite element method.

$$N_{k,l}(\lambda) = \sum_{m=1}^{\infty}\frac{1}{\lambda - \upsilon_m}\langle\nu_k, U_m|_\Gamma\rangle_{L^2(\Gamma)}.\langle U_m|_\Gamma, \nu_l\rangle_{L^2(\Gamma)}. \tag{3.29}$$

Where U_m and υ_m are the eigenfunctions and eigenvalues of the homogeneous Neumann problem on X,

$$(\Delta - \upsilon_m)U_m = 0, \quad \frac{\partial U_m}{\partial n}|_{\partial X} = 0.$$

3.5.2 Improving the rate of convergence

Equation 3.29 gives us a method to compute $N_{k,l}$. It is a vast improvement over direct calculation, for practical purposes, as once we have obtained our as many eigenvalues and Fourier coefficients of their associated eigenfunctions, computing $N(\lambda)$ for any λ we wish, now only involves matrix multiplication, and not applications of the finite element method. Levitin and Marletta devised a simple trick/method to further improve the rate of convergence, or accuracy given a fixed number of eigenvalues and eigenfunctions. We should observe that for $\lambda, \tilde{\lambda}$, calculating N for both and using equation 3.29, noting that the only part of this equation to have any dependence on the parameter is the fraction to the right of the sum, gives:

$$N_{k,l}(\lambda) - N_{k,l}(\tilde{\lambda}) = \sum_{m=1}^{\infty}\frac{\tilde{\lambda} - \lambda}{\upsilon_m^2 - \lambda\upsilon_m - \tilde{\lambda}\upsilon_m + \lambda\tilde{\lambda}}\langle\nu_k, U_m|_\Gamma\rangle_\Gamma.\langle U_m|_\Gamma, \nu_l\rangle_\Gamma.$$

$N_{k,l}(\tilde{\lambda})$ can be calculated beforehand for fixed $\tilde{\lambda}$ as in Definition 66, meaning.

$$N_{k,l}(\lambda) = N_{k,l}(\tilde{\lambda}) + \sum_{m=1}^{\infty}\frac{\tilde{\lambda} - \lambda}{\upsilon_m^2 - \lambda\upsilon_m - \tilde{\lambda}\upsilon_m + \lambda\tilde{\lambda}}\langle\nu_k, U_m|_\Gamma\rangle_\Gamma.\langle U_m|_\Gamma, \nu_l\rangle_\Gamma. \tag{3.30}$$

This now gives, quadratic, as opposed to linear convergence. This process can be repeated as many times as one desires to further increase the speed of convergence. In practice however, it is unnecessary to take more than one extra point and doing so would greatly complicate our implementation of the algorithm we have devised to find derivatives of the scattering matrix, which will be explained in section 4.1.

3.6 An outline of the finite element method

The finite element method is a numerical method to compute the solutions of linear PDE.s and has been of such vital importance to our computations that it deserves its own section. The finite element method is, of course, a massive topic, and we are only able to give it the most cursory treatment possible here. A more thorough overview of finite element analysis can be found in the literature [28] [29]. Nevertheless we will begin:

On a compact domain $X \subset \mathbb{R}^2$, with polygonal boundary, let

$$\Delta\varphi = f, \quad \frac{\partial\varphi}{\partial n}|_{\partial X} = g. \tag{3.31}$$

We know that a weak solution to equation 3.31 is a $\varphi \in \mathcal{D}'(X)$ such that for any test function ψ

$$\langle\Delta\varphi, \psi\rangle = \langle\nabla\varphi, \nabla\psi\rangle - \int_{\partial X} \frac{\partial\varphi}{\partial n}\overline{\psi} = \langle\nabla\varphi, \nabla\psi\rangle - \int_{\partial X} g\overline{\psi} = \langle f, \psi\rangle. \tag{3.32}$$

Re-arranging we see that a weak solution to the problem is akin to finding a unique φ such that for all test functions ψ

$$\langle\nabla\varphi, \nabla\psi\rangle = \int_{\partial X} g\overline{\psi} + \langle f, \psi\rangle. \tag{3.33}$$

At this point we can introduce the Lax Milgram theorem, which can be thought of as a corollary of the [28, Page 18] Riesz Representation theorem, theorem 13. We should take care to observe that the functionals involved here are real.

Theorem 70. ***Lax-Milgram theorem***
Let $\mathcal{H}$ be a real Hilbert space and let q be a bilinear functional on $\mathcal{H}\times\mathcal{H}$ where the following hold:

- *$q(\varphi, \varphi)$ is bounded from below in the Hilbert space norm, i.e. $\exists C_1 > 0$ such that for all $\varphi \in \mathbb{H}$, $q(\varphi, \varphi) \geq C_1\|\varphi\|^2$.*
- *$\exists C_2 > 0$ such that $q(\phi, \varphi) \leq C_2\|\phi\|\|\varphi\|$, forall $\phi, \varphi \in \mathbb{H}$*

If, in addition, we let $l(\cdot)$ be a bounded linear functional on $\mathcal{H}$, then there exists a unique $\varphi \in \mathcal{H}$ for all $\phi \in \mathcal{H}$,

$$q(\varphi, \phi) = l(\phi).$$

With a little extra calculation, we should be able to see that the weak solution to equation 3.31 satisfies the prerequisites of the Lax-Milgram theorem with the first Sobolev inner product, and so a unique solution is guaranteed. Of course a classical solution is necessarily a weak solution.

We must triangulate the domain X i.e. write X as the union of a finite number of triangles, whose sides may only intersect other sides and corners only intersect other corners (see figure below).

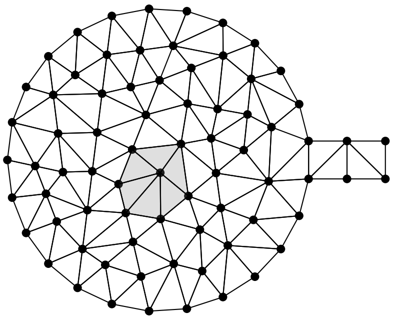


Figure 3.9: A triangulated domain

Our triangulated domain will have N vertices (in figure 3.9, $N = 74$). For each vertex we can constrict a basis function $\nu_k(x, y)$, such that on the k^{th} vertex ν_k will be 1, but 0 on all of the other vertices. Its support will be contained in the triangles having k as a vertex, and on this support it will also be a linear, function of x and y. We call the vector space spanned by these basis functions, the **finite element space** V_N.

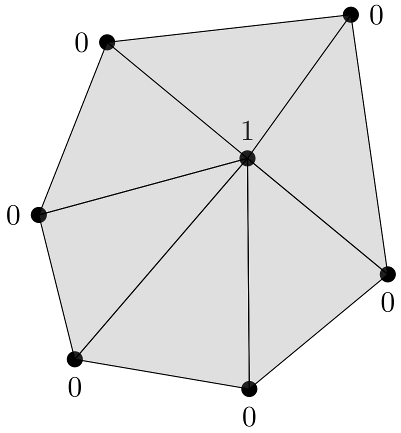


Figure 3.10: The shaded area from Figure 3.9 enlarged

We can approximate φ with linear combinations of N of these functions,
$\nu_1, \ldots \nu_N \in H^1(X)$, where our approximation

$$\varphi_N = \sum_{k=1}^{N} \mathrm{x}_k \nu_k$$

and $\mathrm{x}_1, \ldots \mathrm{x}_N \in \mathbb{C}$ are coefficients to be calculated.

If equation 3.32 is valid for any test function, then it is certainly valid for the basis functions $\nu_1, \ldots, \nu_N$. Thus if we take φ_N to be the "solution" to equation 3.31 then for each ν_l with $1 \leq l \leq N$, the equation 3.32 becomes

$$\langle \nabla \varphi_N, \nabla \nu_l \rangle - \int_{\partial X} g\overline{\nu_l} = \sum_{k=1}^{N} \mathrm{x}_k \langle \nabla \nu_k, \nabla \nu_l \rangle - \int_{\partial X} g\overline{\nu_l} = \langle f, \nu_l \rangle. \tag{3.34}$$

We can now see that the calculation of the x_k has been reduced to a system of N linear equations in N unknowns if we were to write $A\mathrm{x} = \mathrm{b}$, where for k and l;

$$A_{kl} = \langle \nabla \nu_k, \nabla \nu_l \rangle \quad \text{and} \quad \mathrm{b}_l = \int_{\partial X} g\overline{\nu_l} + \langle f, \nu_l \rangle.$$

All of the components that make up A and b are known. A is commonly referred to as the **stiffness matrix.**

Lemma 71. ***Cea's lemma***

$$\|\varphi - \varphi_N\|_{H^1(X)} = \min_{\nu \in V_N} \|\varphi - \nu\|_{H^1(X)}$$

Proof. Let φ_N be the weak solution to equation 3.34 in the finite element space. This means that $\forall \nu \in V_N$

$$q(\varphi_N, \nu) = l(\nu). \tag{3.35}$$

Now let $\varphi \in H^1(X)$ be the unique solution, to equation 3.32, whose existence is guaranteed by the Lax-Milgram theorem. For any $\phi \in H^1(X)$, φ will satisfy

$$q(\varphi, \phi) = l(\phi), \tag{3.36}$$

but since $V_N \subset H^1(X)$, φ also satisfies equation 3.35. Subtracting equation 3.35 from equation 3.36 gives us, for all $\nu \in V_N$,

$$q(\varphi - \varphi_N, \nu) = 0.$$

This phenomena is referred to as **Galerkin orthogonality**. From the two assumptions made in the statement of the Lax-Milgram theorem and using the H^1 norm, we can see that

$$q(\varphi-\varphi_N, \varphi-\varphi_N) \geq C_1\|\varphi-\varphi_N\|^2_{H^1(X)} \quad \text{and also} \quad q(\varphi-\varphi_N, \varphi-\varphi_N) \leq C_2\|\varphi-\varphi_N\|^2_{H^1(X)}.$$

This implies that $\|\varphi - \varphi_N\|^2_{H^1(X)} \leq \frac{C_2}{C_1}\|\varphi - \varphi_N\|^2_{H^1(X)}$, and the result follows. □

This lemma states that the approximation to φ given by the finite element method is the best possible approximation we can have from the finite element space . To compute error estimates, we can construct functions, that are in the space, whose difference with φ in the Sobolev norm can be explicitly bounded; the bound being greater or equal to that achieved with φ_N.

3.6.1 Eigenvalue problems

There are a great many numerical eigenvalue algorithms. In this subsection, we will outline the Arnoldi method, which has made the biggest contribution to the results that we shall present later on in this thesis. The finite element method can also be used for eigenvalue problems. We are looking for a φ and a $\lambda \in \mathbb{R}$ such that $(\Delta - \lambda)\varphi = 0$. In the weak sense, with homogeneous Neumann boundary conditions,this will be

$$\langle \Delta\varphi, \phi \rangle = \langle \nabla\varphi, \nabla\phi \rangle = \lambda\langle \varphi, \phi \rangle$$

and in the finite element space this will be

$$\langle \Delta\varphi_N, \nu_l \rangle = \sum_{k=1}^{N} \mathrm{x}_k \langle \nabla\nu_k, \nabla\nu_l \rangle = \lambda \sum_{k=1}^{N} \mathrm{x}_k \langle \nu_k, \nu_l \rangle$$

Thus we now have two matrices; A, the stiffness matrix, which we have encountered before and $B = \{\langle \nu_k, \nu_l \rangle\}$, and the eigenvalue problem can be re-written as $B^{-1}A\mathrm{x} = \lambda\mathrm{x}$ and be solved via the myriad of numerical eigenvalue algorithms that exist. For the reader's information, the FreeFem++ computer package that uses the implicitly restarted Arnoldi method [30].

3.6.2 The FreeFem++ package and some references to the algorithms

To perform the calculation in this section, we have made extensive use of the the FreeFem++ software package [8]. FreeFem++ produces triangulations (or tetrahedralisation if working in 3-D) for domains given to it by the user in the form of parameterised curves, implements both the finite element method for solving PDEs and the implicitly restarted Arnoldi method for finding eigenvalues and eigenvectors by making use of the ARPACK subroutine [31] [30] [8].

Chapter 4

Calculating scattering matrices and their derivatives

4.1 From the Neumann to Dirichlet map to the scattering matrix and its derivatives

In this section, we will describe the method for constructing generalised eigenfunctions, then the scattering matrix and its derivatives from the Neumann to Dirichlet map. As usual, we fix a sheet of Z and define J, our indexing set, to be the $j \in \mathbb{N}$ such that $\operatorname{Im}(\sqrt{\lambda - \mu_j}) < 0$. With this in mind we can proceed as follows:

4.1.1 Calculating the S matrix

Definition 72. *Define*

$$P_J : L^2(\Gamma) \longrightarrow L^2(\Gamma),$$

to be the projection whose kernel is spanned by the Neumann eigenfunctions of Γ *associated to* μ_j *for* $j \in J$*.*

Fix a basis $\{\nu_j(y)\}$ for the space of Neumann eigenfunctions on Γ, with corresponding to Neumann eigenvalues μ_j.

From Proposition 62, we know that a solution to $(\Delta - \lambda)\varphi(\lambda, x, y) = 0$ on the cylindrical ends E of M, with homogeneous Neumann boundary conditions on the boundary will be of the form:

$$\varphi(\lambda, x, y) = \sum_{j \in J} \left(e^{-i\sqrt{\lambda - \mu_j}x} \nu_j(y) + \sum_{k \in J} S_{j,k}(\lambda) e^{i\sqrt{\lambda - \mu_k}x} \nu_k(y) \right) + \sum_{j \in \mathbb{N} \setminus J} T_j(\lambda) e^{i\sqrt{\lambda - \mu_j}x} \nu_j(y),$$

At $x = 0$, $\varphi(\lambda, x, y)$ will be of the form

$$\varphi(\lambda, 0, y) = \sum_{j \in J} \left(\nu_j(y) + \sum_{k \in J} S_{j,k}(\lambda) \nu_k(y) \right) + \sum_{j \in \mathbb{N} \setminus J} T_j(\lambda) \nu_j(y).$$

But at $x = 0$, the normal derivative will be of the form

$$\sum_{j \in J} \left(i\sqrt{\lambda - \mu_j} \nu_j(y)(\lambda) - \sum_{k \in J} i\sqrt{\lambda - \mu_k} S_{j,k}(\lambda) \nu_k(y) \right) - \sum_{j \in \mathbb{N} \setminus J} i\sqrt{\lambda - \mu_j} T_j(\lambda) \nu_j(y).$$

Definition 73. *Let*

$$D(\lambda, k) = \frac{\partial^k}{\partial \lambda^k} \begin{pmatrix} i\sqrt{\lambda-\mu_1} & 0 & \cdots & \cdots & 0 \\ 0 & i\sqrt{\lambda-\mu_2} & \cdots & \cdots & \cdots \\ \vdots & \vdots & i\sqrt{\lambda-\mu_3} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & \vdots & \vdots & 0 & i\sqrt{\lambda-\mu_{|J|}}. \end{pmatrix}$$

Here $k \in \mathbb{N}$ and $D(\lambda, k)$ is the $|J| \times |J|$ matrix acting on the set of J modes for which $\operatorname{Im}(\sqrt{\lambda-\mu_j}) < 0$. For the calculation of $S(\lambda)$, we we will only be using this in the context of $k = 0$. In this case we will denote it as $\tilde{N}(\lambda)$; it's inverse is the Neumann to Dirichlet maps on E. For the higher derivatives, we will sometimes abbreviate the notation $D(\lambda, k)$, to $D(k)$

We should note here, that this projection P_J, acting on modes $j \in \mathbb{N} \backslash J$ and sending the rest to 0, is vital to this calculation. This is because, on one hand, the Neumann to Dirichlet map on E, only acts on the boundary data of square integrable functions, thus we must project out the non square integrable modes of any such generalised eigenfunction before applying $\tilde{N}(\lambda)^{-1}$. It is only in this context that $\tilde{N}(\lambda)^{-1}$ is a Neumann to dirichlet map for $(\Delta - \lambda)$ on E.

Now observe that the internal domain X and the external domain (or ends) E share a common boundary Γ. This means that for any generalised eigenfunction (whose properties we have defined many times) defined on M, the action of the Neumann to Dirichlet map calculated on X composed with P_J and the Neumann to Dirichlet map calculated on E should coincide.

Set

$$L(\lambda) = \left(P_J N(\lambda) - \tilde{N}(\lambda)^{-1} P_J \right). \tag{4.1}$$

This matrix, will have a null-space of dimension $|J|$. Each element of $\operatorname{Ker}(L)$ can be equated with a one of the $|J|$ summands of $\varphi(\lambda, 0, y)$, as defined in equation 3.16 with in terms of some, possibly unknown orthonormal basis of Neumann eigenfunctions of Δ_Γ.

Applying a singular value decomposition algorithm, or some procedure to find the kernel of a matrix, i.e. QR, gives us $|J|$ kernel vectors.

Let us take the null-space of L

$$W = \{w^1, \cdots, w^J\}.$$

For each $\omega_j \in W$ we have a representation of a generalised eigenfunction, evaluated at 0, of the form:

$$\omega^j = \bigoplus_{j \in J} (\delta_{j,k} + S_{j,k}(\lambda)) \bigoplus_{j \in \mathbb{N} \backslash J} T_j(\lambda), \tag{4.2}$$

We could now, in theory, extract the scattering matrix from this, but before we are able to do such things, and for our scattering matrix to be of any use to us, we need to control the basis of Neumann eigenfunctions. A numerical algorithm for singular value decomposition will not necessarily give us $S(\lambda)$ in terms of the basis we want; the basis of Neumann eigenfunction of Γ that we carefully chose when we began the calculation of $N(\lambda)$ in the previous section.

Having found a basis for the null space of (4.1), we restrict our attention to the elements of these vectors that represent the J Fourier modes and discard the rest by means of application of the operator $(1-P_J)$. The image of W under both $(1-P_J)$ and $(1-P_J)N(\lambda)$ forms a basis in $\mathbb{R}^J$.

We can think of the linear map $\tau : \mathbb{R}^J \longrightarrow \mathbb{R}^J$, defined on the $(1-P_J)w^j$, by

$$(1-P_J)\omega^j \mapsto (1-P_J)N\omega^j,$$

as the identity map from the basis $\{(1-P_J)\omega^1, \cdots, (1-P_J)\omega^J\}$ of $\mathbb{R}^J$, to basis $\{(1-P_J)N\omega^1, \cdots, (1-P_J)N\omega^J\}$ of $\mathbb{R}^J$.

We must re-write τ in terms of the standard basis, whose elements represent the chosen basis of the Neumann eigenfunctions of Γ. Thus, when acting our chosen basis of $L^2(\Gamma)$, τ can be written as,

$$\tau(\lambda) = \{(1-P_J)N\omega^1, \cdots, (1-P_J)N\omega^J\}^{-1}\{(1-P_J)\omega^1, \cdots, (1-P_J)\omega^J\}.$$

Now we note that applying N to each ω_j gives:

$$N(\lambda)\omega^j = \bigoplus_{j\in J} \frac{S_{j,k}(\lambda) - \delta_{j,k}}{i\sqrt{\lambda - \mu_j}} \bigoplus_{j\in \mathbb{N}\setminus J} \frac{T_j(\lambda)}{i\sqrt{\lambda - \mu_j}}.$$

One can see now that, as the map $(1-P)N$ will take the $J \times J$ matrix form

$$\tau(\lambda) = (D(\lambda, 0) + S(\lambda).D(\lambda, 0)).(\mathrm{Id} + S(\lambda))^{-1}. \tag{4.3}$$

This means that finally

$$S(\lambda) = (\tau(\lambda) - \tilde{N}(\lambda))^{-1}(-\tilde{N}(\lambda) - \tau(\lambda)). \tag{4.4}$$

4.2 Derivatives of the S matrix

The aim of this section is to be able to calculate $S^{(n)}(\lambda) = \frac{\partial^n}{\partial \lambda^n} S(\lambda)$. This is interesting in its own right but, in our case, the motivation for calculating $S^{(n)}(\lambda)$ will be to assist us in or search for resonances later on, by use of the argument principle. We will first outline the method for obtaining the first derivative, then generalise to higher derivatives.

4.2.1 A Neumann to Dirichlet map for the system on the external domain

Let us fix a sheet of Z. We know that if $\varphi(\lambda, x, y)$ is a generalised eigenfunction, then $(\Delta - \lambda)\varphi(\lambda, x, y) = 0$. Thus, when we differentiate with respect to λ, we get:

$$\frac{\partial}{\partial \lambda}(\Delta - \lambda)\varphi(\lambda, x, y) = (\Delta - \lambda)\varphi'(\lambda, x, y) - \varphi(\lambda, x, y) = 0.$$

Indeed, inductively, for any n we get

$$\frac{\partial^n}{\partial \lambda^n}(\Delta - \lambda)\varphi(\lambda, x, y) = (\Delta - \lambda)\varphi^{(n)}(\lambda, x, y) - \varphi^{(n-1)}(\lambda, x, y) = 0,$$

where $\varphi^{(n)}(\lambda, x, y)$ denotes $\frac{\partial^n}{\partial\lambda^n}\varphi(\lambda, x, y)$ for brevity. One can simply look for a solution to the resulting system of equations in a similar manner to equation 3.22:

$$\begin{aligned}
&(\Delta-\lambda)\varphi(\lambda,x,y)=0, && \frac{\partial\varphi}{\partial n}|_{\Sigma}=0,\\
&(\Delta-\lambda)\varphi'(\lambda,x,y)-\varphi(\lambda,x,y)=0, && \frac{\partial\varphi'}{\partial n}|_{\Sigma}=0,\\
&\qquad\qquad\vdots && \quad\vdots\\
&(\Delta-\lambda)\varphi^{(n)}(\lambda,x,y)-\varphi^{(n-1)}(\lambda,x,y)=0, && \frac{\partial\varphi^{(n)}}{\partial n}|_{\Sigma}=0.
\end{aligned}\tag{4.5}$$

On the other hand, since $\varphi(\lambda, x, y)$ is known on E, we can recall equation 3.18 and see that

$$\varphi(\lambda,x,y)=\sum_{j\in J}\left(e^{-i\sqrt{\lambda-\mu_j}x}\nu_j(y)+\sum_{k\in J}S_{j,k}(\lambda)e^{i\sqrt{\lambda-\mu_k}x}\nu_k(y)\right)+\sum_{j\notin J}T_j(\lambda)e^{i\sqrt{\lambda-\mu_j}x}\nu_j(y),$$

and deduce that in E

$$\begin{aligned}
\varphi'(\lambda,x,y)=&\sum_{j\in J}\left(\frac{-ix}{2\sqrt{\lambda-\mu_j}}e^{-i\sqrt{\lambda-\mu_j}x}\nu_j(y)+\sum_{k\in J}\left[S'_{j,k}(\lambda)-\frac{xS_{j,k}(\lambda)}{2i\sqrt{\lambda-\mu_k}}\right]e^{i\sqrt{\lambda-\mu_k}x}\nu_k(y)\right)\\
&+\sum_{j\notin J}\left[T'_j(\lambda)+\frac{ixT_j(\lambda)}{2\sqrt{\lambda-\mu_j}}\right]e^{i\sqrt{\lambda-\mu_j}x}\nu_j(y),
\end{aligned}\tag{4.6}$$

$$\begin{aligned}
\frac{\partial}{\partial x}\varphi'(\lambda,x,y)=&\sum_{j\in J}\left[\frac{-i}{2\sqrt{\lambda-\mu_j}}-x\right]e^{-i\sqrt{\lambda-\mu_j}x}\nu_j(y)\\
&+\sum_{j,k\in J}\left[i\sqrt{\lambda-\mu_k}S'_{j,k}(\lambda)+\frac{iS_{j,k}(\lambda)}{2\sqrt{\lambda-\mu_k}}-xS_{j,k}(\lambda)\right]e^{i\sqrt{\lambda-\mu_k}x}\nu_k(y)\\
&+\sum_{j\notin J}\left[i\sqrt{\lambda-\mu_j}T'_j(\lambda)+\frac{iT_j(\lambda)}{2\sqrt{\lambda-\mu_j}}-xT_j(\lambda)\right]e^{i\sqrt{\lambda-\mu_j}x}\nu_j(y),
\end{aligned}\tag{4.7}$$

At $x = 0$ these two generalised functions become:

$$\varphi'(\lambda,0,y)=\sum_{j\in J}\left(\sum_{k\in J}S'_{j,k}(\lambda)\nu_k(y)\right)+\sum_{j\in\mathbb{N}\setminus J}T'_j(\lambda)\nu_j(y),\tag{4.8}$$

$$\begin{aligned}
\frac{\partial}{\partial x}\varphi'(\lambda,0,y)=&\sum_{j\in J}\left(\frac{1}{2i\sqrt{\lambda-\mu_j}}\nu_j(y)+\sum_{k\in J}\left[i\sqrt{\lambda-\mu_k}S'_{j,k}(\lambda)-\frac{S_{j,k}(\lambda)}{2i\sqrt{\lambda-\mu_k}}\right]\nu_k(y)\right)\\
&+\sum_{j\in\mathbb{N}\setminus J}\left(i\sqrt{\lambda-\mu_j}T'_j(\lambda)-\frac{T_j(\lambda)}{2i\sqrt{\lambda-\mu_j}}\right)\nu_j(y).
\end{aligned}\tag{4.9}$$

We can now see that the Neumann to Dirichlet map for the system of equations given by (4.8) and (4.8) will be of the form

$$\tilde{N}_1(\lambda): l^2\oplus l^2\to l^2\oplus l^2.$$

So by the fact that

$$\frac{\partial}{\partial x}\varphi'(\lambda,0,y)=D(0)\varphi'(\lambda,0,y)+D(1)\varphi(\lambda,0,y)$$

in addition to the fact that

$$\frac{\partial}{\partial x}\varphi(\lambda,0,y)=D(0)\varphi(\lambda,0,y),$$

We can see that

$$\tilde{N}_{(1)}(\lambda)=\begin{pmatrix}\tilde{N}(\lambda)P_J & 0\\ \frac{\partial}{\partial\lambda}\tilde{N}(\lambda) & \tilde{N}(\lambda)\end{pmatrix}^{-1}.$$

There is no reason for us to limit ourselves to first derivatives. We should go further now and do the same thing for $\varphi^{(n)}(\lambda,x,y)$. It is at this point the we take note of the fact that each successive differentiation of $e^{\pm i\sqrt{\lambda-\mu}x}$ respect to λ, produces a factor of of x. Since we will be focussing on φ and $\frac{\partial}{\partial x}\varphi$ at the boundary, it is unnecessary to differentiate $e^{\pm i\sqrt{\lambda-\mu}x}$ more than once, and all terms that result from such actions, terms in these summands with a factor of x^2 will simply be denoted them as h.o.t..

We will now introduce some new notation: $\tilde{N}_{(n)}(\lambda)$ and $N_{(n)}(\lambda)$ to be the Neumann to Dirichlet maps for $\varphi^{(n)}(\lambda,x,y)$ on the external and internal domains respectively. Now let us begin:

$$\begin{aligned}\varphi^{(n)}(\lambda,x,y)&=\sum_{j\in J}\left[-xD_j(n)+\text{h.o.t}\right]e^{-i\sqrt{\lambda-\mu_j x}}\nu_j(y) &(4.10)\\ &+\sum_{j,k\in J}\left[x\sum_{q=1}^{n}\binom{n}{q}D_k(q).S_{j,k}^{(n-q)}(\lambda)+S_{j,k}^{(n)}(\lambda)+\text{h.o.t}\right]e^{i\sqrt{\lambda-\mu_k x}}\nu_k(y)\\ &+\sum_{j\notin J}\left[x\sum_{q=1}^{n}\binom{n}{q}D_j(q).T_j^{(n-q)}(\lambda)+T_j^{(n)}(\lambda)+\text{h.o.t}\right]e^{i\sqrt{\lambda-\mu_j}x}\nu_j(y).\end{aligned}$$

So we can see that

$$\varphi^{(n)}(\lambda,0,y)=\sum_{j,k\in J}S_{j,k}^{(n)}(\lambda)\nu_k(y)+\sum_{j\notin J}T_j^{(n)}(\lambda)\nu_j(y), \qquad (4.11)$$

$$\begin{aligned}\frac{\partial}{\partial x}\varphi^{(n)}(\lambda,0,y)&=\sum_{j\in J}\left[-D_j(n)\right]\nu_j(y)+\sum_{j,k\in J}\left[\sum_{q=0}^{n}\binom{n}{q}D_k(q).S_{j,k}^{(n-q)}(\lambda)\right]\nu_k(y) &(4.12)\\ &+\sum_{j\notin J}\left[\sum_{q=0}^{n}\binom{n}{q}D_j(q).T_j^{(n-q)}(\lambda)\right]\nu_j(y).\end{aligned}$$

We can now see that $\tilde{N}_{(n)}(\lambda)$ will be a block-upper triangular matrix,

$$\tilde{N}_{(n)}(\lambda):\bigoplus_{\{0,\ldots,n\}}l^2\longrightarrow\bigoplus_{\{0,\ldots,n\}}l^2,$$

given by

$$\tilde{N}_{(n)}(\lambda)=\begin{pmatrix}\tilde{N}(\lambda)P_J & 0 & \cdots & \cdots & \cdots & 0\\ \frac{\partial}{\partial\lambda}\tilde{N}(\lambda) & \tilde{N}(\lambda) & 0 & \cdots & \cdots & 0\\ \binom{2}{0}\frac{\partial^2}{\partial\lambda^2}\tilde{N}(\lambda) & \binom{2}{1}\frac{\partial}{\partial\lambda}\tilde{N}(\lambda) & \binom{2}{2}\tilde{N}(\lambda) & 0 & \cdots & 0\\ \cdots & \cdots & \cdots & \cdots & \cdots & 0\\ \cdots & \cdots & \cdots & \cdots & \cdots & 0\\ \binom{n}{0}\frac{\partial^n}{\partial\lambda^n}\tilde{N}(\lambda) & \binom{n}{1}\frac{\partial^{n-1}}{\partial\lambda^{n-1}}\tilde{N}(\lambda) & \cdots & \cdots & \cdots & \binom{n}{n}\tilde{N}(\lambda)\end{pmatrix}^{-1}$$

4.2.2 A Neumann to Dirichlet map for the system on the internal domain

Now we should turn our attention towards the internal domain X. We can compute $N_{(n)}(\lambda)$ with an adaptation of Levitin-Marletta's method. Equation 3.29 describes the method when used to compute $N(\lambda)$. The subsequent trick to increase the rate of convergence is unaffected by our extension.

In the same context as equation 3.29, we will use $\{U_m\}$ to denote the orthonormal Neumann eigenfunctions, μ_m their corresponding eigenvalues. $\{\nu_j\}$ will denote orthonormal basis of $L^2(\Gamma)$ and finally, we will use $\{\Phi_k\}$ and $\{\Phi_k^{(n)}\}$, all elements $L^2(X)$, to denote solutions to the following system:

$$\begin{aligned}
&(\Delta-\lambda)\Phi_k(\lambda,x,y)=0, && \frac{\partial\Phi_k}{\partial n}|_\Sigma=0,\ \frac{\partial\Phi_k}{\partial n}|_\Gamma=\nu_{k_0} \qquad (4.13)\\
&(\Delta-\lambda)\Phi'_k(\lambda,x,y)-\Phi_k(\lambda,x,y)=0, && \frac{\partial\Phi'_k}{\partial n}|_\Sigma=0,\ \frac{\partial\Phi'_k}{\partial n}|_\Gamma=\nu_{k_1}\\
&\quad\vdots && \quad\vdots\\
&(\Delta-\lambda)\Phi_k^{(n)}(\lambda,x,y)-\Phi_k^{(n-1)}(\lambda,x,y)=0, && \frac{\partial\Phi_k^{(n)}}{\partial n}|_\Sigma=0,\ \frac{\partial\Phi_k^{(n)}}{\partial n}|_\Gamma=\nu_{k_n},
\end{aligned}$$

where k is the n-tuple $\{k_0,k_1,\dots k_n\}$.

We begin, by fixing λ and will now calculate elements of $N_{(n)}(\lambda)$, which map to $\Phi^{(n)}$, where $k_n\in\mathbb{N}$ and $l\in\mathbb{N}^n$:

$$\begin{aligned}
N_{(n),k_n,l}&=\langle N\nu_{k_n},\nu_l\rangle=\langle\Phi_k^{(n)}|_\Gamma,\nu_l\rangle=\langle\Phi_k^{(n)}|_\Gamma,\frac{\partial\Phi_l^{(n)}}{\partial_n}_\Gamma\rangle\\
&=\langle\nabla\Phi_k^{(n)},\nabla\Phi_l^{(n)}\rangle+\langle\Delta\Phi_k^{(n)},\Phi_l^{(n)}\rangle=\langle\nabla\Phi_k^{(n)},\nabla\Phi_l^{(n)}\rangle+\lambda\langle\Phi_k^{(n)},\Phi_l^{(n)}\rangle+\langle\Phi_k^{(n-1)},\Phi_l^{(n)}\rangle.
\end{aligned}$$

Since each Φ_k can be written as $\sum_m\Phi_k\langle\Phi_k,U_m\rangle$ (the same is true for $\Phi_k^{(n)}$), and by definition, $\Phi_k=(\Delta-\lambda)^{-j}\Phi^{(n-j)}$ we get that.

$$\begin{aligned}
N_{(n),k_n,l}&=\sum_m(\langle\nabla U_m,\nabla U_m\rangle+\lambda)\langle\Phi_k^{(n)},U_m\rangle\langle U_m,\Phi_l^{(n)}\rangle+\langle\Phi_l^{(n-1)},\Phi_l^{(n)}\rangle\\
&=\sum_m(\langle\nabla U_m,\nabla U_m\rangle+\lambda)\langle\Phi_k^{(n)},U_m\rangle\langle U_m,\Phi_l^{(n)}\rangle+\sum_m\langle\Phi_k^{(n-1)},U_m\rangle\langle U_m,\Phi_l^{(n)}\rangle\\
&=\sum_m(\lambda-\mu_m)\langle\Phi_k^{(n)},U_m\rangle\langle U_m,\Phi_l^{(n)}\rangle+\sum_m\langle\Phi_k^{(n-1)},U_m\rangle\langle U_m,\Phi_l^{(n)}\rangle. \qquad (4.14)
\end{aligned}$$

Now, using Green's second identity, for any j;

$$\begin{aligned}
\langle\Phi_j,U_m\rangle\ &=\sum_m\frac{1}{\lambda-\mu_m}\left(\langle\Delta\Phi_j,U_m\rangle-\langle\Phi_j,\Delta U_m\rangle\right)=\sum_m\frac{1}{\lambda-\mu_m}\langle\nu_j,U_m|_\Gamma\rangle \qquad (4.15)\\
\langle\Phi_j^{(n)},U_m\rangle&=\sum_m\frac{1}{\lambda-\mu_m}\left(\langle\Delta\Phi_j^{(n)},U_m\rangle-\langle\Phi_j^{(n)},\Delta U_m\rangle-\langle\Phi_j^{(n-1)},U_m\rangle\right).\\
&=\sum_m\frac{1}{\lambda-\mu_m}\left(\langle\nu_{j_n},U_m|_\Gamma\rangle-\langle\Phi_j^{(n-1)},U_m\rangle\right).
\end{aligned}$$

So then $N_{(n),k,l}$ becomes

$$\sum_m\left(\langle\nu_k,U_m|_\Gamma\rangle-\langle\Phi_j^{(n-1)},U_m\rangle\right)\langle U_m,\Phi_l^{(n)}\rangle,$$

and inductively, we see that

$$N_{(n),k_n,l} = \sum_m \langle \nu_{k_n}, U_m|_\Gamma \rangle \langle U_m, \Phi_l^{(n)} \rangle = \sum_m \sum_{p=1}^{n} \frac{(-1)^{p-1}}{(\lambda - \mu_m)^p} \langle \nu_{k_n}, U_m|_\Gamma \rangle \langle U_m|_\Gamma, \nu_{l_p}, \rangle.$$

Thus the n^{th} block-row for $N_{(n)}(\lambda)$ is made up of the direct sum of maps defined componentwise by

$$\eta_{(p)}(\lambda) = \sum_m \frac{(-1)^{p-1}}{(\lambda - \mu_m)^p} \langle \nu_k, U_m|_\Gamma \rangle \langle U_m|_\Gamma, \nu_{l,} \rangle,$$

where p runs through $1, \ldots n$.

We can now finally see that for the system, $N_n(\lambda)$ is a block upper-triangular matrix, acting on $\Phi \oplus \cdots \oplus \Phi^{(n)}$, of the same form as $\tilde{N}_{(n)}(\lambda)$, given by

$$N_{(n)}(\lambda) = \begin{pmatrix} \eta_{(1)}(\lambda) & 0 & \cdots & \cdots & \cdots & 0 \\ \eta_{(2)}(\lambda) & \eta_{(1)}(\lambda) & 0 & \cdots & \cdots & 0 \\ \eta_{(3)}(\lambda) & \eta_{(2)}(\lambda) & \eta_{(1)}(\lambda) & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \eta_{(n)}(\lambda) & \eta_{(n-1)}(\lambda) & \cdots & \cdots & \cdots & \eta_{(1)}(\lambda) \end{pmatrix}$$

4.2.3 Extracting $S^{(n)}(\lambda)$

With this, we can now compute the coefficients of $\varphi_j^{(n)}(\lambda, 0, y)$ by finding a basis for the null space of

$$L_n = \left(P_J N_{(n)} - \tilde{N}_{(n)} \right).$$

We will denote such a basis as $\omega_1^{(n)}, \ldots \omega_J^{(n)}$. Now, finally, using the same argument made when calculating $S(\lambda)$

$$\tau_n(\lambda) = \{(1 - P_J)\omega_1^{(n)}, \cdots, (1 - P_J)\omega_J^{(n)}\}\{(1 - P_J)N\omega_1^{(n)}, \cdots, (1 - P_J)N\omega_J^{(n)}\}^{-1},$$

thus

$$\begin{aligned} \tau_n(\lambda) &= \left(D(n) - \sum_{q=0}^{n} \binom{n}{q} D(q).S^{(n-q)}(\lambda) \right) S^{(n)}(\lambda)^{-1} \\ &= \left(D(n) - \sum_{q=1}^{n} \binom{n}{q} D(q).S^{(n-q)}(\lambda) - D(0)S^{(n)}(\lambda) \right) S^{(n)}(\lambda)^{-1} \end{aligned}$$

Where all of the $S^{(n-q)}(\lambda)$ are known, having been previously calculated. Finally we can say that

$$S^{(n)}(\lambda) = \left(\mathrm{Id} - D(0)\tau_n(\lambda) \right)^{-1} . \left(D(n).\tau_n(\lambda) - \sum_{q=1}^{n} \binom{n}{q} D(q).\tau_n(\lambda).S^{(n-q)}(\lambda) \right).$$

4.2.4 A few words about computational cost

Before moving on, we ought to give at least a cursory mention to the issue of computational complexity and cost. The main strength of this procedure is that the very costly finite element method only needs to be performed once for each domain. Once this has been performed, the computation of the scattering matrix and its derivatives is then a matter of data processing and linear algebra. We will nevertheless make some comments on the computational complexity of these procedures. The formula for computing $N(\lambda)$ uses equation 3.30. When computing $N(\lambda)$ in practice, we must truncate these sums and only use a finite number of modes (ν_k) and eigenfunctions (U_m). Sticking with the same notation as in example 67, we will have $K \times M$ modes in total (there are K ends) and E eigenfunctions.

We can see that to compute $N(\lambda)$, we must perform a matrix multiplication between two rectangular matrices of dimensions $K.M \times E$ and $E \times K.M$. This has computational complexity of $\mathcal{O}(K^2M^2E)$.

Once we have obtained the $N(\lambda)$ or $N_{(n)}(\lambda)$ we must perform singular value decomposition and several matrix inversions on square matrices of dimensions $K.M$ or $K.M.n^2$ in the case where we wish to calculate derivatives. The computational complexity of such operations is $\mathcal{O}(3K^3M^3.n^6)$ and approximately $\mathcal{O}((K.M.n^2)^{2.4})$ respectively [32].

As well as the cost of these matrix computations, there is also the unknown, but highly non-trivial cost of parsing and processing the data we have in order to produce the matrices we perform these operations on. This becomes more costly with more modes, a higher rank scattering matrix (larger $|J|$), more derivatives and more ends.

4.3 Embedded eigenvalues and resonances

4.3.1 Embedded eigenvalues

Our operator; the self adjoint extension of $(\Delta - \lambda)$, on M, with Neumann boundary conditions, has spectrum $\mathbb{R}_+$. We know that the spectrum of an operator is made up of a disjoint union of its pure point spectrum; eigenvalues, and the residual or continuous spectrum (See definition 40). The question now arises as to when does the operator have an eigenvalue embedded in its continuous spectrum; an **embedded eigenvalue**.

The paper by Evans, Levitin and Vassiliev [33] proved the existence of embedded eigenvalues for the Neumann Laplacian on two dimensional waveguides with an obstacle, and/or deformation of the waveguide so long as the domain has cross-sectional symmetry. The argument involved looking for antisymmetric solutions φ to $(\Delta - \lambda)\varphi = 0$ with all of the usual boundary conditions. Solutions to such a problem are also solutions to the original problem without the condition of antisymmetry. The continuous spectrum of such an operator (antisymmetric) starts at the first transverse Neumann eigenvalue $(\frac{\pi^2}{w^2})$, thus to prove existence of an embedded eigenvalue, it is sufficient to prove existence of an eigenvalue below the continuous spectrum for the antisymmetric problem. This was achieved using cutoff functions and variational methods. The problem is greatly simplified due to the symmetry, as when we are considering anti-symmetric solutions on such a domain, it can be reduced to the top or bottom cross section, with Dirichlet boundary conditions on the midpoint. This was further generalised to waveguides with cylindrical ends by Davies and Parnovski [34]. Parnovski and Levitin have, amongst others, produced two other papers on this topic [35] [36].

Embedded eigenvalues can be calculated numerically by looking for zero eigenvalues of the sub-matrix of L in equation 4.1, obtained by omitting the rows and columns representing non square integrable modes. As resonances, which are neither Neumann eigenvalues nor embedded eigenvalues, inhabit the complex plane, this is not an adequate method to locate them. Fortunately, we can make use of the very useful properties of the scattering matrix and its derivatives to achieve this.

4.3.2 Resonances and their calculation

We define a **resonance** to be a pole of the scattering matrix. The relationship between the resolvent, the Neumann to Dirichlet map and the scattering matrix means that poles of the resolvent coincide with zeros of the determinant of the the inverse of the scattering matrix, and their multiplicities will be the same. Previous work on this topic has focused on on either providing asymptotic bounds for the number of resonances([2] [37] [38] [19] [39]), or their calculation ([7] [40] [9] [41] [42] [43] [44]). We will of course focus on the latter.

Theorem 65 is now extremely useful to us because it tells us that every pole of $S(\lambda)$ on a non-physical sheet of Z coincides with a zero of $S(\lambda^*)$ in the physical sheet and vice versa. Remember, we have used λ^* to denote the canonical projection of λ to the physical sheet if there is ever any ambiguity about which is which. When both λ and λ^* are identified with a subset of the complex plane, they will be in the same location. Since the resolvent and scattering matrix are holomorphic in the physical sheet, it can't have poles there, and we will have no zeros in a non-physical sheet of Z. This means we can now make use of the argument principle to locate the resonances, and locating resonances in a non-physical sheet of Z has been reduced to locating zeros in the physical sheet.

When we apply the argument principle to the determinant of $S(\lambda)$ and, using the Jacobi formula, integrate $\mathrm{Tr}(S(\lambda)^{-1}.S'(\lambda))$ over a contour in the physical sheet of Z; there will be no poles to cancel out the effect of any zeros there may be within that contour. Thus we can detect and even count the number resonances enclosed by a contour. This approach was featured in the paper by Davies and Aslanyan, but not applied to the scattering matrix [40]. To summarise:

Proposition 74. *Let C be a contour in a non-physical sheet of Z with winding number one then defining the counting function $\#$, $C \mapsto \mathbb{N}$ which counts the number of poles enclosed by C, we get that.*

$$\#(C) = \frac{1}{2\pi i} \oint_C \mathrm{Tr}(S(\lambda)^{-1}.S'(\lambda)).$$

If a contour can be found that contains one or more zeros, we can subdivide then integrate over the subdivisions and repeat the process until a small enough contour has been found, containing a single resonance. Then we can use Newton's method to obtain its exact location, or in practical terms, to a desired accuracy. We can then multiply the scattering matrix $S(\lambda)$ by $(\lambda_0 - \lambda)^{-1}$, where λ_0 is the location of the zero, then apply Newton's method again, repeating if necessary to find the order of the zero.

Parnovski and Vassiliev [9] were the first to explicitly compute embedded eigenvalues and resonances for waveguides with cylindrical ends in two dimensions. Levitin and Marletta replicated their results with their simple algorithm for calculating Neumann to Dirichlet maps [7], which has been the inspiration for this work. In the next section, we have used this method to replicate these results again and to obtain data for a larger class of objects.

Chapter 5

Numerical computations on various domains using the scattering matrix

As we now have a fast and efficient algorithm for the computation of the scattering matrix, we can now put it to work. This section will describe the outcome of some of our our numerical experiments. We will first apply it to replicate the results of Levitin and Marletta and Aslanyan, Parnovski and Vassiliev.

5.1 Cylinders with a Circular Obstacle

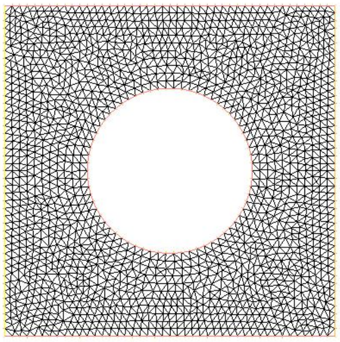


Figure 5.1: One of our triangulated interior domains. The waveguide is composed of this interior domain, with the two ends the same width as the interior domain joined on the left and right sides.

Two papers provided the inspiration for this. The first by Aslanyan, Parnovski, etc. etc. and the second by Michael Levitin and Marco Marletta. Amongst other things, Levitin and Marletta looked at a cylinder (continuous rectangle) of width 2, with a single circular obstruction. The radius of this obstruction is varied, along with the position of its centre (vertical displacement), relative to the centre line of the cylinder. It was found that when the vertical displacement was 0, there exist embedded eigenvalues, as there ought to be! When this displacement becomes non-zero, the embedded eigenvalue decays to a resonance [7]. With the parameterisation of $\lambda \mapsto \lambda^2$, they presented a number of values for these resonances. We will perform our calculation to the highest accuracy we are able to and compare our results to theirs. We largely agree with the calculation of Levitin and Marletta and believe that our calculations offer an improvement on the number of decimal places.

	Our calculation	Levitin-Marletta	Aslanyan et al
$R = 0.3$			
$\delta = 0$	1.50497	1.50486	1.5048
$\delta = 0.1$	$1.50783 + 0.0001205i$	$1.5078 + 10^{-4}i$	$1.5102 + \times 10^{-4}i$
$\delta = 0.2$	$1.51651 + 0.0004740i$	$1.5165 + 5 \times 10^{-4}i$	$1.5188 = 5 \times 10^{-4}i$
$R = 0.5$			
$\delta = 0$	1.39138	1.39134	1.3913
$\delta = 0.1$	$1.39785 + 0.0009255i$	$1.3979 + 9 \times 10^{-4}i$	$1.3998 + 9 \times 10^{-4}i$
$\delta = 0.2$	$1.41779 + 0.0039101i$	$1.4178 + 3.90 \times 10^{-3}i$	$1.4196 + 3.93 \times 10^{-3}i$

5.1.1 Some notes on mesh refinement

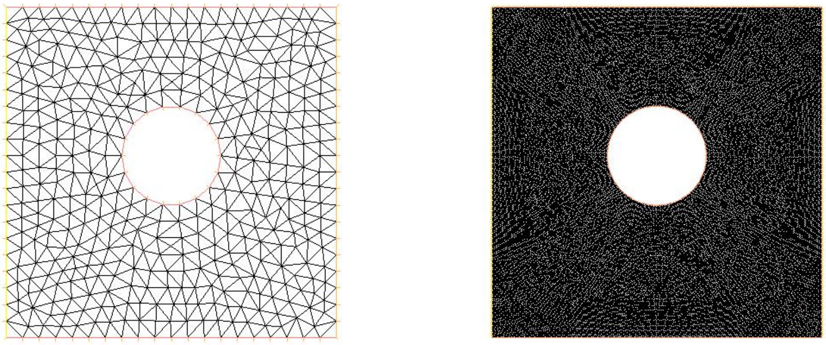


Figure 5.2: Some internal domains for $R = 0.3$, $\delta = 0.1$ with mesh refinement of 10 and 80 respectively.

With these results as a starting point, we were able to look at the effects of mesh refinements, on the accuracy of the results we obtained. Whilst it is obvious that the more refined the mesh becomes, the more accurate the result, we have tabulated the results of some experiments to demonstrate just to what extent. The standard method for producing a triangulated domain in FreeFem is to draw the outline as a union of parameterised curves, then use the programs own triangulation algorithm after specifying the number of points on each such curve. Our scale of mesh refinement will be the number of such points per unit length on the boundary. We have tabulated some results for the calculation of these resonances, using 2000 eigenvalues and 20 modes for each connected component of Γ (see Figure 5.2) The reader can see how as the mesh refinement increases, the result stabilises.

Mesh refinement	$R = 0.3$ $\delta = 0.1$	$R = 0.3$ $\delta = 0.2$
10	$1.50943 + 0.0001157i$	$1.51791 + 0.0004530i$
15	$1.50847 + 0.0001185i$	$1.51708 + 0.0004657i$
20	$1.50821 + 0.0001193i$	$1.51684 + 0.0004691i$
25	$1.50805 + 0.0001198i$	$1.51670 + 0.0004712i$
30	$1.50797 + 0.0001200i$	$1.51663 + 0.0004722i$
35	$1.50793 + 0.0001202i$	$1.51660 + 0.0004727i$
40	$1.50790 + 0.0001203i$	$1.51657 + 0.0004731i$
45	$1.50788 + 0.0001203i$	$1.51655 + 0.0004734i$
50	$1.50786 + 0.0001204i$	$1.51654 + 0.0004736i$
55	$1.50785 + 0.0001204i$	$1.51653 + 0.0004737i$
60	$1.50785 + 0.0001204i$	$1.51652 + 0.0004738i$
65	$1.50784 + 0.0001204i$	$1.51651 + 0.0004739i$
70	$1.50783 + 0.0001205i$	$1.51651 + 0.0004740i$
75	$1.50783 + 0.0001205i$	$1.51651 + 0.0004740i$
80	$1.50783 + 0.0001205i$	$1.51651 + 0.0004740i$
N	$R = 0.5$ $\delta = 0.1$	$R = 0.5$ $\delta = 0.2$
10	$1.39874 + 0.0009122i$	$1.41857 + .00385170i$
15	$1.39822 + 0.0009199i$	$1.41811 + 0.0038854i$
20	$1.39805 + 0.0009225i$	$1.41796 + 0.0038971i$
25	$1.39797 + 0.0009237i$	$1.41789 + 0.0039021i$
30	$1.39793 + 0.0009243i$	$1.41786 + 0.0039049i$
35	$1.39791 + 0.0009247i$	$1.41784 + 0.0039063i$
40	$1.39789 + 0.0009249i$	$1.41782 + 0.0039075i$
45	$1.39788 + 0.0009251i$	$1.41781 + 0.0039083i$
50	$1.39787 + 0.0009252i$	$1.41781 + 0.0039089i$
55	$1.39786 + 0.0009253i$	$1.41780 + 0.0039092i$
60	$1.39786 + 0.0009254i$	$1.41780 + 0.0039095i$
65	$1.39786 + 0.0009254i$	$1.41779 + 0.0039097i$
70	$1.39785 + 0.0009255i$	$1.41779 + 0.0039098i$
75	$1.39785 + 0.0009255i$	$1.41779 + 0.0039101i$
80	$1.39785 + 0.0009255i$	$1.41779 + 0.0039101i$

5.1.2 Some notes on the number of eigenvalues

In the previous subsection, we mentioned that we have used 2000 eigenvalues but gave no justification for this. Here we intend to present the reader with some graphs and charts to demonstrate convincingly, if not necessarily rigorously, why this was a good choice. On the same theme, we ought to observe what happens when we increase the number of eigenvalues. We will display some convergence graphs that track the value of the leading coefficient of the scattering matrix as the number of eigenvalues increases, for a selection of the domains above at a random point.

Number of eigenvalues	$R = 0.3$ $\delta = 0.1$	$R = 0.3$ $\delta = 0.2$
200	$1.50783 + 0.000120482i$	$1.51651 + 0.000474043i$
400	$1.50783 + 0.000120482i$	$1.51651 + 0.000474045i$
600	$1.50783 + 0.000120482i$	$1.51651 + 0.000474045i$
800	$1.50783 + 0.000120483i$	$1.51651 + 0.000474045i$
1000	$1.50783 + 0.000120483i$	$1.51651 + 0.000474046i$
1200	$1.50783 + 0.000120483i$	$1.51651 + 0.000474046i$
1400	$1.50783 + 0.000120483i$	$1.51651 + 0.000474046i$
1600	$1.50783 + 0.000120483i$	$1.51651 + 0.000474046i$
1800	$1.50783 + 0.000120483i$	$1.51651 + 0.000474046i$
2000	$1.50783 + 0.000120483i$	$1.51651 + 0.000474046i$
N	$R = 0.5$ $\delta = 0.1$	$R = 0.5$ $\delta = 0.2$
200	$1.39785 + 0.000925529i$	$1.41779 + 0.00391010i$
400	$1.39785 + 0.000925536i$	$1.41779 + 0.00391013i$
600	$1.39785 + 0.000925536i$	$1.41779 + 0.00391014i$
800	$1.39785 + 0.000925537i$	$1.41779 + 0.00391014i$
1000	$1.39785 + 0.000925537i$	$1.41779 + 0.00391014i$
1200	$1.39785 + 0.000925537i$	$1.41779 + 0.00391014i$
1400	$1.39785 + 0.000925537i$	$1.41779 + 0.00391014i$
1600	$1.39785 + 0.000925537i$	$1.41779 + 0.00391014i$
1800	$1.39785 + 0.000925537i$	$1.41779 + 0.00391014i$
2000	$1.39785 + 0.000925537i$	$1.41779 + 0.00391014i$

From this it might seem like it is unnecessary to use many eigenvalues, however the number if eigenvalues does have a significant impact on the coefficients of the scattering matrix as we shall demonstrate with some graphs. We have picked, as an example, the domain where $R = 0.3$ and $\delta = 0.1$ with a mesh refinement of 80. We have plotted the real and imaginary components of the leading coefficient of the scattering matrix at the value $1 + 0.1i$. This is typical behaviour for any arbitrarily chosen point.

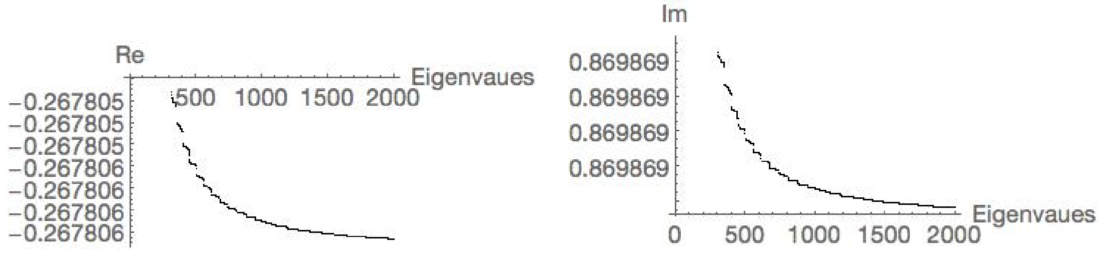


Figure 5.3: The real and imaginary components of the first coefficient of the scattering matrix plotted against the number of eigenvalues and eigenvectors used to compute it.

It should be noted that the number of modes and the auxiliary point chosen has an undetectable effect on the calculations as long as the choice is "sensible". It should also be noted from our discussions on computational cost that the increasing the number of modes is the most computationally costly action we can take and that they should be minimised. For the rest of the results, we have used 20 modes for each end, 1000 eigenvalues and a mesh refinement of 30.

5.1.3 Observing a resonance transition from one sheet to another

We have calculated resonances and embedded eigenvalues over a wider variety of radii and displacements, and we have also looked beyond the first non-physical sheet. In the tables below and throughout, we will be using the parameterisation $\lambda \mapsto \lambda$ instead of $\lambda \mapsto \lambda^2$.

	$R = 0.3$	$R = 0.4$	$R = 0.5$	$R = 0.6$
0	2.26495	2.09281	1.93595	1.81802
0.1	$2.27386 + 0.00030i$	$2.10666 + 0.00111i$	$1.95414 + 0.00258i$	$1.84091 + 0.0051i$
0.2	$2.30007 + 0.00143i$	$2.14855 + 0.00460i$	$2.01025 + 0.01108i$	$1.91246 + 0.024i$
0.3	$2.34144 + 0.00309i$	$2.21863 + 0.01073i$	$2.10809 + 0.02749i$	$2.04002 + 0.0631i$
0.4	$2.39242 + 0.00481i$	$2.31323 + 0.01870i$	$2.24875 + 0.05208i$	-
0.5	$2.4409 + 0.00510i$	$2.41444 + 0.02222i$	-	-
0.6	$2.67240 + 0.00134i$	-	-	-

Of particular interest here is the case where $R = 0.2$. As δ increases from 0.6 to 0.7 the resonance moves from the sheet $J = \{1\}$ to $J = \{2\}$ as can be seen in the table below.

$R = 0.2$	$J = \{1\}$	$J = \{2\}$
$\delta = 0$	2.4036	-
$\delta = 0.1$	$2.40712 + 0.00006i$	-
$\delta = 0.2$	$2.41709 + 0.00021i$	-
$\delta = 0.3$	$2.43170 + 0.00040i$	-
$\delta = 0.4$	$2.44777 + 0.00059i$	-
$\delta = 0.5$	$2.46101 + 0.00053i$	-
$\delta = 0.6$	$2.46725 + 0.00013i$	-
$\delta = 0.7$	-	$2.46475 + 0.00063i$

Figure 5.4: Resonances for the domain $R = 0.2$ showing the resonance moving to a different sheet of Z.

We have plotted the absolute value of the determinant of the scattering matrix for this occurrence. We can observe that the "tail" of the resonance is visible on both sheets prior to the resonance moving sheets.

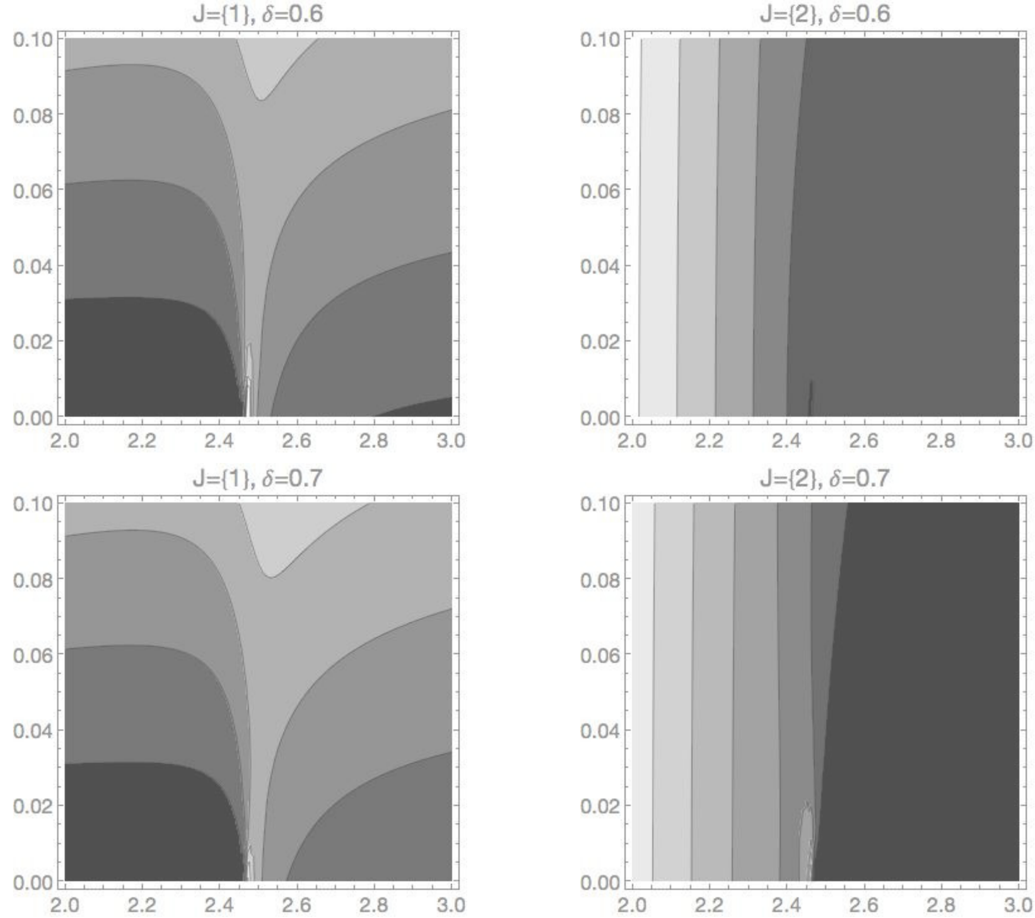


Figure 5.5: Contour plots showing the resonance crossing from one sheet of Z to another. Here $R = 0.2$ throughout.

5.2 Some more numerical results

The main bulk of our numerical experiments concerns circular domains, with one or more ends attached, and with the possibility of one circular "obstacle" inside. There are a number of variables we can alter. We can adjust the the radius and position of the obstacle, the width of the ends, the number of ends, and in the case of multiple ends, the angles between them. For each such instance, we have searched for resonances, on sheets $J = \{1\}$, $J = \{2\}$ and $J = \{3\}$, within in the search area given by

$$\{\lambda : 0 \leq \mathrm{Re}(\lambda) \leq 15, -3 \leq \mathrm{Im}(\lambda) \leq 3\}, \tag{5.1}$$

and tabulated the results found. We can be confident that the accuracy of the resonances calculated is at least three decimal places, though we have included the fourth place in a lighter shade for the reader's information. In addition to this, we have plotted them on graphs, with colour coded markers indicating the respective sheet of Z they reside on; black for $J = \{1\}$, red for $J = \{2\}$ and green for $J = \{3\}$. In the case of varying widths, the "paths" taken by the resonances as the widths vary continuously are clearly visible in table form, and we have included them there too.

5.3 Varying the widths of ends

We use w to denote the width of the end(s). Beforehand, we will show first 9 non-zero Neumann eigenvalues, of the circle of radius 2, which can be compared to the resonances, especially when w is small.

0.8476	2.3323	3.6709	4.4130	7.0698	7.1068	10.2911	11.2442	12.3059

Figure 5.6: Neumann eigenvalues for a circle of radius 2

5.3.1 The singe ended case

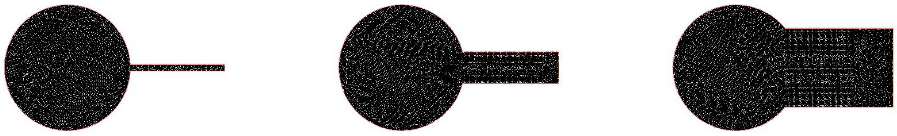
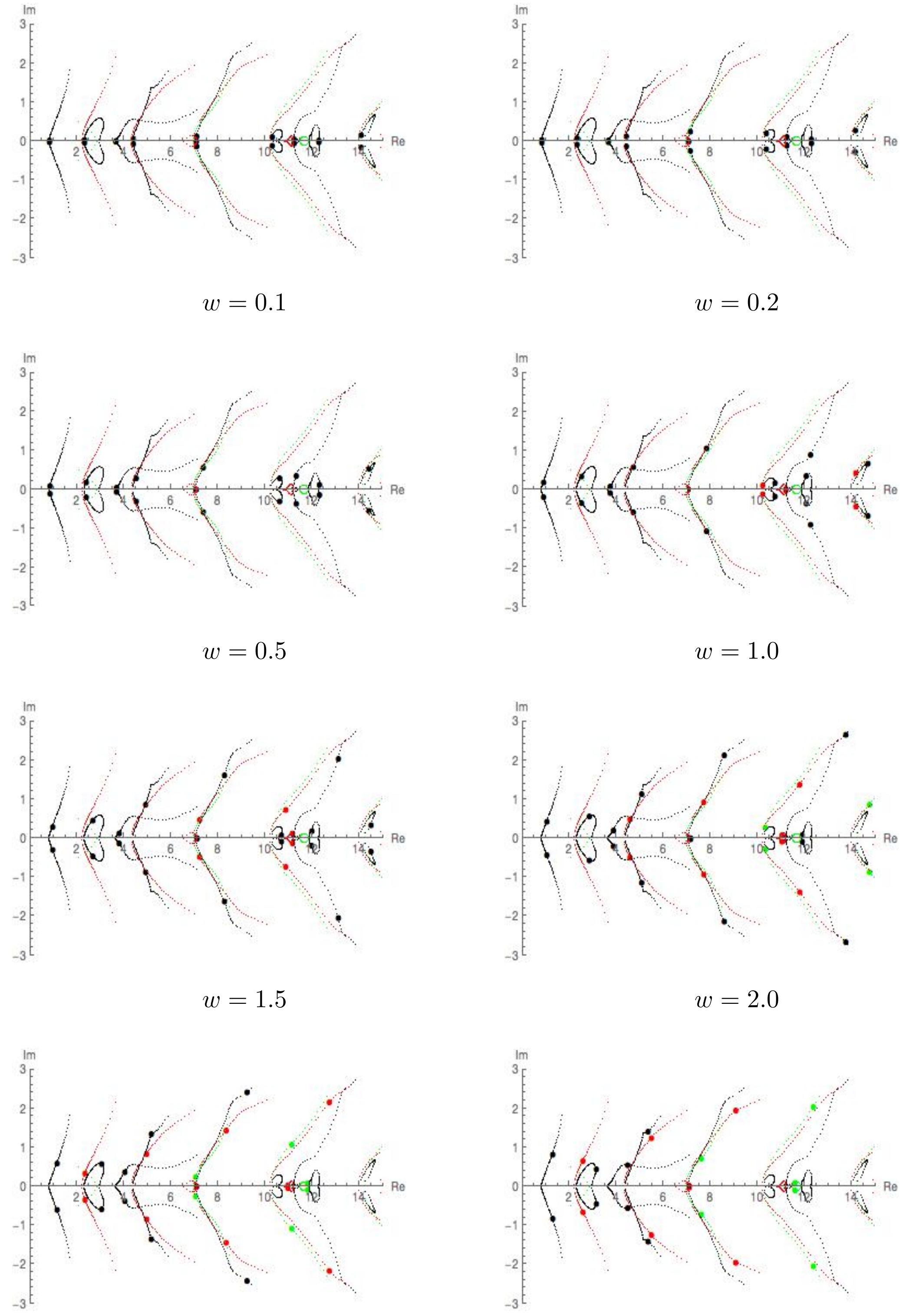


Figure 5.7: An example of some internal domains from waveguides with a single end: varying widths.

$J = \{1\}$			
$w = 0.1$	$w = 0.2$	$w = 0.5$	$w = 1$
$0.8496 + 0.0206i$ $2.3388 + 0.0416i$ $3.6742 + 0.0145i$ $4.4246 + 0.0664i$ $7.0947 + 0.0022i$ $7.1124 + 0.1328i$ $10.3293 + 0.1186i$ $11.2547 + 0.0572i$ $12.3122 + 0.0263i$ $14.1139 + 0.1527i$	$0.8541 + 0.0408i$ $2.3525 + 0.0807i$ $3.6814 + 0.0265i$ $4.4480 + 0.1285i$ $7.0946 + 0.0011i$ $7.1721 + 0.2541i$ $10.4037 + 0.2071i$ $11.2733 + 0.1151i$ $12.3237 + 0.0502i$ $14.1949 + 0.2766i$	$0.8753 + 0.0992i$ $2.413 + 0.1848i$ $3.7091 + 0.0512I$ $4.5452 + 0.3009I$ $7.0947 + 0.0004i$ $7.4065 + 0.5657i$ $10.6664 + 0.2940i$ $11.368 + 0.35467i$ $12.3567 + 0.1348i$ $14.4649 + 0.5345i$	$0.9255 + 0.1957i$ $2.5431 + 0.3334i$ $3.7577 + 0.0802i$ $4.7425 + 0.5764i$ $7.8573 + 1.0573i$ $10.7726 + 0.1699i$ $12.2884 + 0.8945i$ $12.0991 + 0.3521i$ $14.7205 + 0.6654i$
$w = 1.5$	$w = 2$	$w = 2.5$	$w = 3$
$0.9900 + 0.3010i$ $2.6965 + 0.4627i$ $3.8154 + 0.1224i$ $4.9403 + 0.8597i$ $7.1276 + 0.0014i$ $8.2933 + 1.6165i$ $10.7234 + 0.0883i$ $12.0129 + 0.1897i$ $13.1459 + 2.0493i$ $14.5252 + 0.3368i$	$1.0710 + 0.4264i$ $2.8746 + 0.5614i$ $3.9000 + 0.2054i$ $5.0970 + 1.1417i$ $7.1757 + 0.0151i$ $8.6102 + 2.1380i$ $11.9290 + 0.0758i$ $13.780 + 2.6635i$	$1.1752 + 0.5898i$ $3.0696 + 0.5767i$ $4.0569 + 0.3750i$ $5.1843 + 1.3491i$ $9.2528 + 2.4257i$	$1.3176 + 0.8290i$ $3.1713 + 0.4476i$ $4.5055 + 0.5486i$ $5.3676 + 1.4029i$

$J = \{2\}$			
$w = 0.1$	$w = 0.2$	$w = 0.5$	$w = 1$
			$10.2536 + 0.11562i$ $11.2435 + 0.01777i$ $14.2167 + 0.42736i$
$w = 1.5$	$w = 2$	$w = 2.5$	$w = 3$
$7.2439 + 0.4698i$ $7.7433 + 0.9299i$ $10.9022 + 0.7275i$ $11.1970 + 0.1239i$	$4.6065 + 0.4829i$ $11.0769 + 0.08842i$ $11.8264 + 1.38616i$	$2.3695 + 0.3389i$ $4.9869 + 0.8324i$ $7.1156 + 0.0011i$ $8.3670 + 1.44198i$ $10.9968 + 0.0355i$ $12.7631 + 2.1634i$	$2.6042 + 0.6511i$ $5.5217 + 1.2393i$ $7.1356 + 0.0192i$ $9.1096 + 1.9492i$ $14.1616 + 2.7545i$
$J = \{3\}$			
$w = 0.1$	$w = 0.2$	$w = 0.5$	$w = 1$
$w = 1.5$	$w = 2$	$w = 2.5$	$w = 3$
	$10.3673 + 0.2721i$ $14.7867 + 0.8682i$	$7.0639 + 0.2546i$ $11.831 + 0.0873i$ $11.161 + 1.0833i$	$7.6306 + 0.7147i$ $11.6268 + 0.0988i$ $12.4059 + 2.0430i$



$w = 0.1$

$w = 0.2$

$w = 0.5$

$w = 1.0$

$w = 1.5$

$w = 2.0$

$w = 2.5$

$w = 3.0$

Figure 5.8: Some colour coded plots of the location of resonances. In this instance the width of a single end is changed. The larger markers are the resonances for the domain indicated, the smaller markers are the entire family of domains with varying widths. This allows the reader to visualise the paths the resonances take as the width of the end increases.

5.3.2 The two ended case

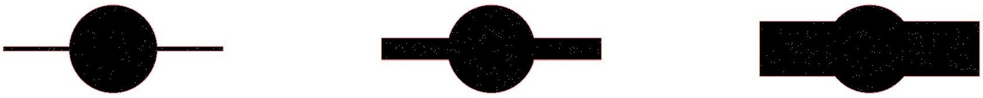
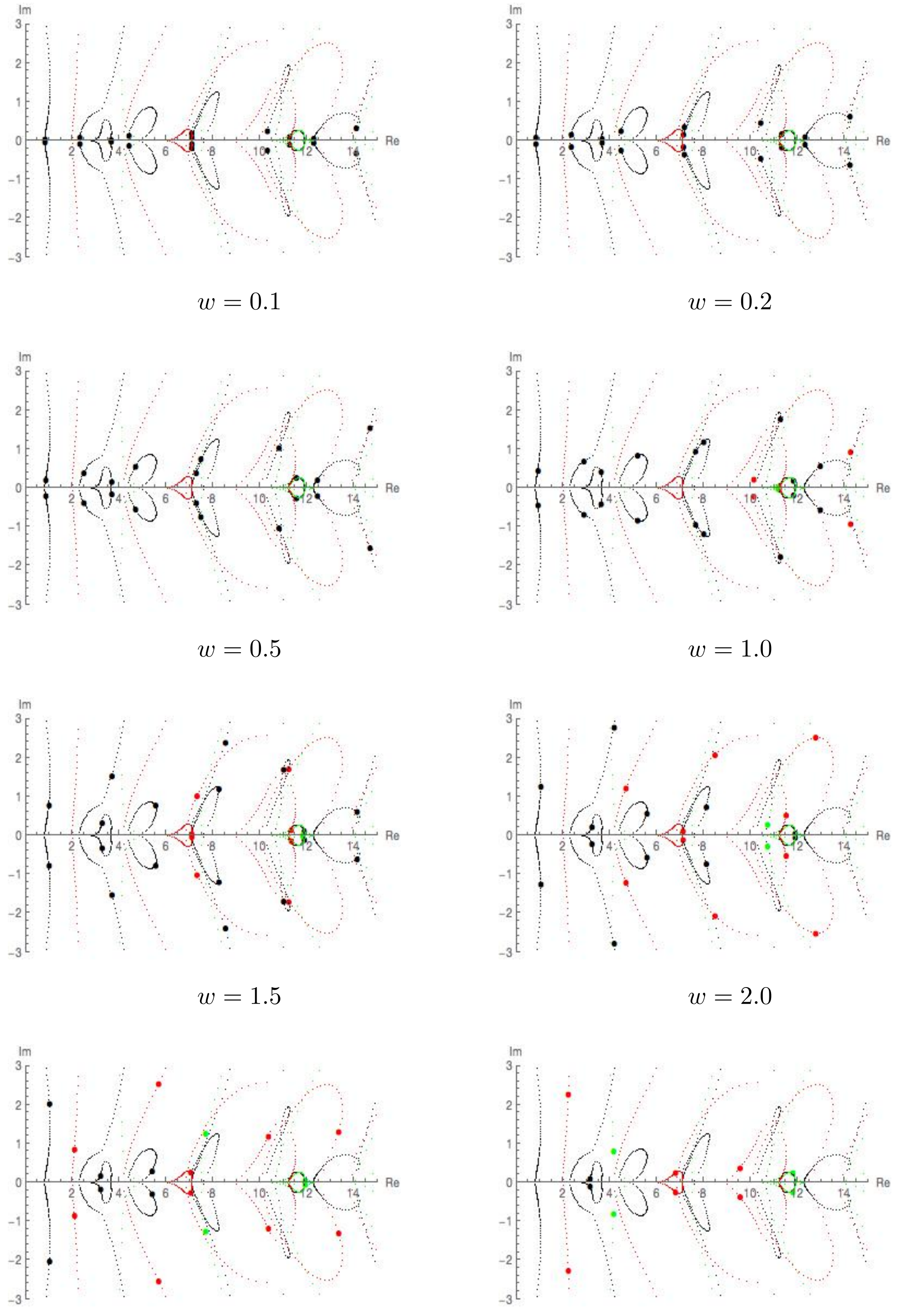


Figure 5.9: An example of some internal domains from waveguides with two ends; varying widths.

$J = \{1\}$			
$w = 0.1$	$w = 0.2$	$w = 0.5$	$w = 1$
$0.8509 + 0.0414i$ $2.3452 + 0.0835i$ $3.6747 + 0.0301i$ $4.4392 + 0.1318i$ $7.1259 + 0.0840i$ $7.1121 + 0.1860i$ $10.3503 + 0.2468i$ $11.2828 + 0.1040i$ $12.3216 + 0.0514i$ $14.1394 + 0.3189i$	$0.8580 + 0.0826i$ $2.3731 + 0.1625i$ $3.6818 + 0.0596i$ $4.4930 + 0.2511i$ $7.1630 + 0.1596i$ $7.1941 + 0.3502i$ $10.4585 + 0.4660i$ $11.3545 + 0.1746i$ $12.3520 + 0.0935i$ $14.2642 + 0.6227i$	$0.8892 + 0.2092i$ $2.5064 + 0.3788i$ $3.7019 + 0.1561i$ $4.7222 + 0.5472i$ $7.3102 + 0.3771i$ $7.5071 + 0.7514i$ $10.8385 + 1.0359i$ $11.5877 + 0.2568i$ $12.4763 + 0.2109i$ $14.7212 + 1.5512i$	$0.9556 + 0.4519i$ $2.9102 + 0.6890i$ $3.6465 + 0.4109i$ $5.2043 + 0.8424i$ $7.6756 + 0.9436i$ $8.0220 + 1.1812i$ $11.8564 + 0.1919i$ $11.2959 + 1.7761i$ $13.0017 + 0.5615i$
$w = 1.5$	$w = 2$	$w = 2.5$	$w = 3$
$1.0257 + 0.7763i$ $3.2914 + 0.3258i$ $3.7096 + 1.5368i$ $5.5731 + 0.7769i$ $8.2802 + 1.2045i$ $8.5558 + 2.3949i$ $11.9213 + 0.1188i$ $11.0326 + 1.7017i$ $14.1736 + 0.6063i$	$1.0765 + 1.2550i$ $3.2596 + 0.2248i$ $4.2086 + 2.7882i$ $5.6029 + 0.5707i$ $8.1398 + 0.7284i$ $11.9305 + 0.0507i$	$1.0520 + 2.0286i$ $3.2260 + 0.1682i$ $5.4271 + 0.3014i$	$3.1694 + 0.1047i$
$w = 1.50$	$w = 2$	$w = 2.5$	$w = 3$

$J = \{2\}$			
$w = 0.1$	$w = 0.2$	$w = 0.5$	$w = 1$
			$10.1725 + 0.2234i$ $11.2516 + 0.0346i$ $14.2966 + 0.9315i$
$w = 1.5$	$w = 2$	$w = 2.5$	$w - 3$
$7.1219 + 0.0348i$ $7.3486 + 1.0152i$ $11.3740 + 0.1479i$ $11.2578 + 1.7061i$	$4.7058 + 1.2089i$ $7.1508 + 0.1085i$ $8.5068 + 2.0717i$ $11.5491 + 0.5205i$ $12.8056 + 2.5197i$	$2.1175 + 0.8556i$ $5.7087 + 2.5471i$ $7.0720 + 0.2686i$ $10.3901 + 1.1825i$ $13.3884 + 1.2995i$	$2.2528 + 2.2713i$ $6.8281 + 0.2552i$ $9.5801 + 0.3726i$

$J = \{3\}$			
$w = 0.1$	$w = 0.2$	$w = 0.5$	$w = 1$
			$11.1845 + 0.0014i$
$w = 1.5$	$w = 2$	$w = 2.5$	$w = 3$
$11.8592 + 0.0015i$	$10.7461 + 0.2789i$	$7.7114 + 1.2593i$ $11.9540 + 0.0523i$	$4.1757 + 0.7998i$ $11.8240 + 0.2510i$



$w = 0.1$ $w = 0.2$

$w = 0.5$ $w = 1.0$

$w = 1.5$ $w = 2.0$

$w = 2.5$ $w = 3.0$

Figure 5.10: Some colour coded plots of the location of resonances. This is similar to the previous set of plots, but the domains now have two ends.

5.3.3 The three ended case

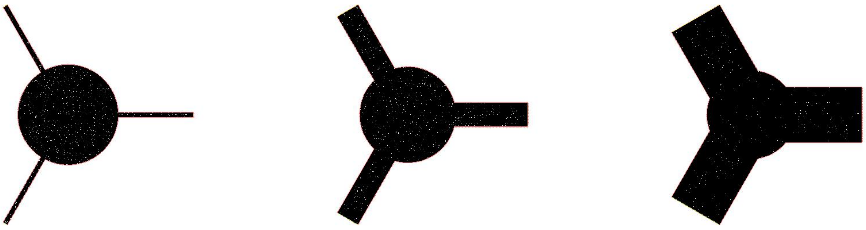
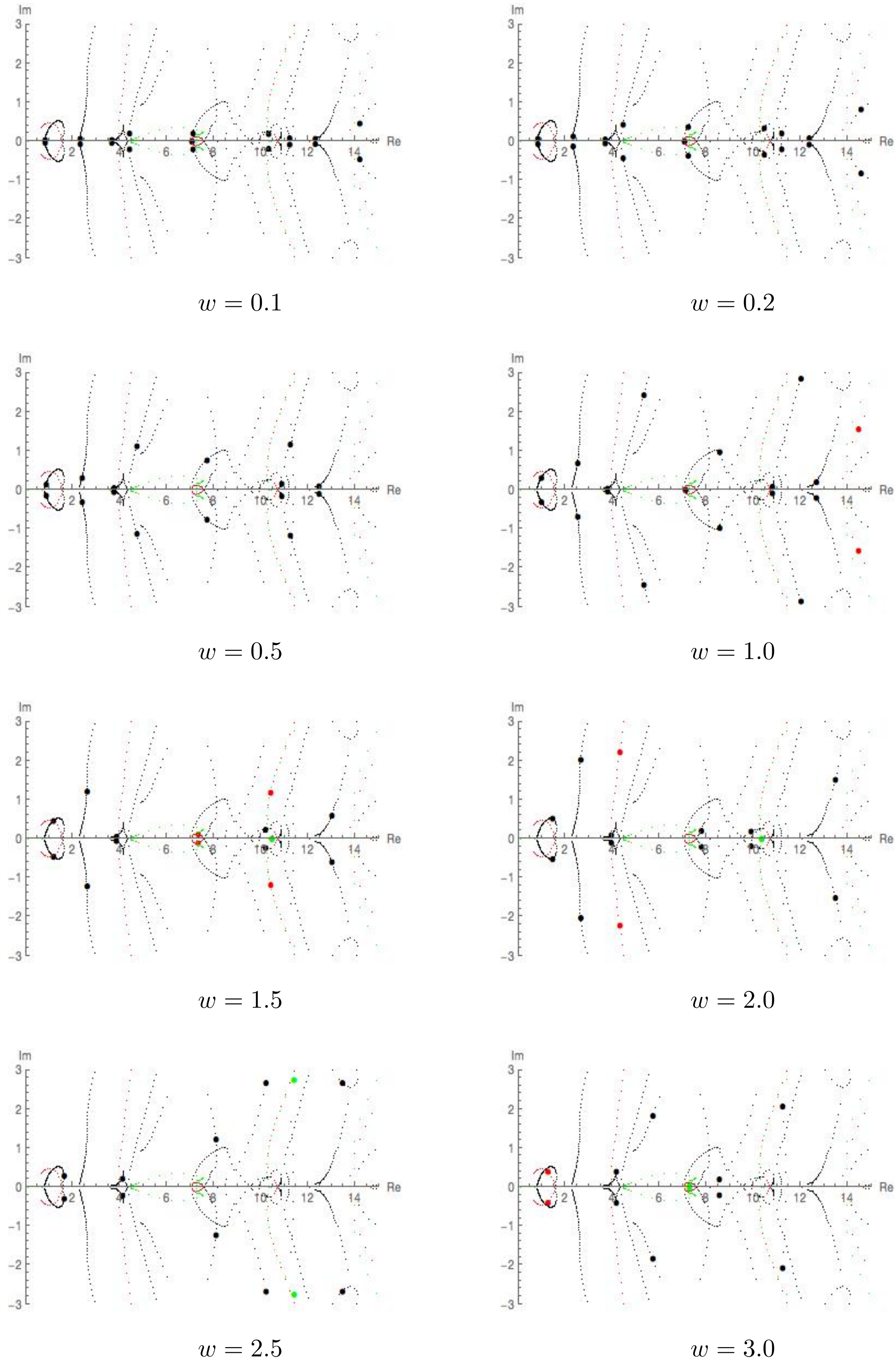


Figure 5.11: An example of some internal domains for waveguides with three ends; varying widths.

$J = \{1\}$			
$w = 0.1$	$w = 0.2$	$w = 0.5$	$w = 1$
$0.8509 + 0.0310i$ $2.3406 + 0.0631i$ $3.6876 + 0.0392i$ $4.4364 + 0.2060i$ $7.0954 + 0.0012i$ $7.1367 + 0.1995i$ $10.3521 + 0.1822i$ $11.2476 + 0.0902i$ $12.3419 + 0.0642i$ $14.2288 + 0.4620i$	$0.8585 + 0.0613i$ $2.3575 + 0.1249i$ $3.7181 + 0.0572i$ $4.4858 + 0.4231i$ $7.0973 + 0.0003i$ $7.2502 + 0.3719i$ $10.4892 + 0.3319i$ $11.2313 + 0.1970i$ $12.3922 + 0.0862i$ $14.6062 + 0.8252i$	$0.8977 + 0.1504i$ $2.4287 + 0.3136i$ $3.7754 + 0.0522i$ $4.7570 + 1.1240i$ $7.7405 + 0.7594i$ $10.9137 + 0.1569i$ $11.2761 + 1.1723i$ $12.4942 + 0.0991i$	$1.0110 + 0.3023i$ $2.5605 + 0.6908i$ $3.8210 + 0.0438i$ $5.3757 + 2.4361i$ $7.1394 + 0.0028i$ $8.5970 + 0.9766i$ $10.8319 + 0.0763i$ $12.6999 + 0.2015i$ $12.0468 + 2.8639i$
$w = 1.5$	$w = 2$	$w = 2.5$	$w = 3$
$1.2045 + 0.4550i$ $2.6372 + 1.2188i$ $3.8809 + 0.0547i$ $10.2237 + 0.2288i$ $13.0471 + 0.6002i$	$1.4852 + 0.5157i$ $2.6861 + 2.0245i$ $3.9864 + 0.0980i$ $7.8132 + 0.2051i$ $9.9248 + 0.1923i$ $13.5139 + 1.5089i$	$1.6623 + 0.2910i$ $4.1456 + 0.2225i$ $8.1191 + 1.2241i$ $10.2449 + 2.6780i$ $13.4925 + 2.6813i$	$4.1989 + 0.4051i$ $5.7571 + 1.8297i$ $8.5785 + 0.2062i$ $11.2726 + 2.0682i$
$J = \{2\}$			
$w = 0.1$	$w = 0.2$	$w = 0.5$	$w = 1$
			$14.4973 + 1.5573i$
$w = 1.5$	$w = 2$	$w = 2.5$	$w = 3$
$7.3603 + 0.1101i$ $10.4455 + 1.1805i$	$4.3538 + 2.2170i$		$1.2935 + 0.4018i$
$J = \{3\}$			
$w = 0.1$	$w = 0.2$	$w = 0.5$	$w = 1$
$w = 1.50$	$w = 2$	$w = 2.5$	$w = 3$
$10.4964 + 0.0004i$	$10.3713 + 0.0036i$	$11.4364 + 2.7495i$	$7.3004 + 0.0429i$



$w = 0.1$

$w = 0.2$

$w = 0.5$

$w = 1.0$

$w = 1.5$

$w = 2.0$

$w = 2.5$

$w = 3.0$

Figure 5.12: We now repeat this for three ended domains and a similar pattern emerges. We can see that with all three diagrams, the resonances start off very close to the eigenvalues of the object without ends (Figure 5.6) and then migrate outwards as the widths of the ends are increased. Some resonances curl around and cross sheets before doing this though.

5.4 Varying the size of an obstacle of fixed position

With the width of the end(s) fixed at one, we introduce an obstacle in the centre and vary its width, which we shall denote by R.

5.4.1 The single ended case

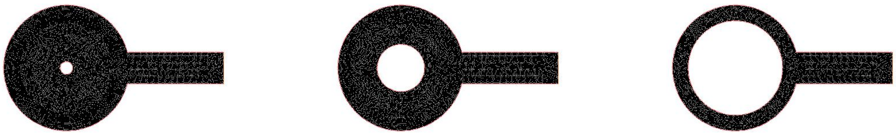
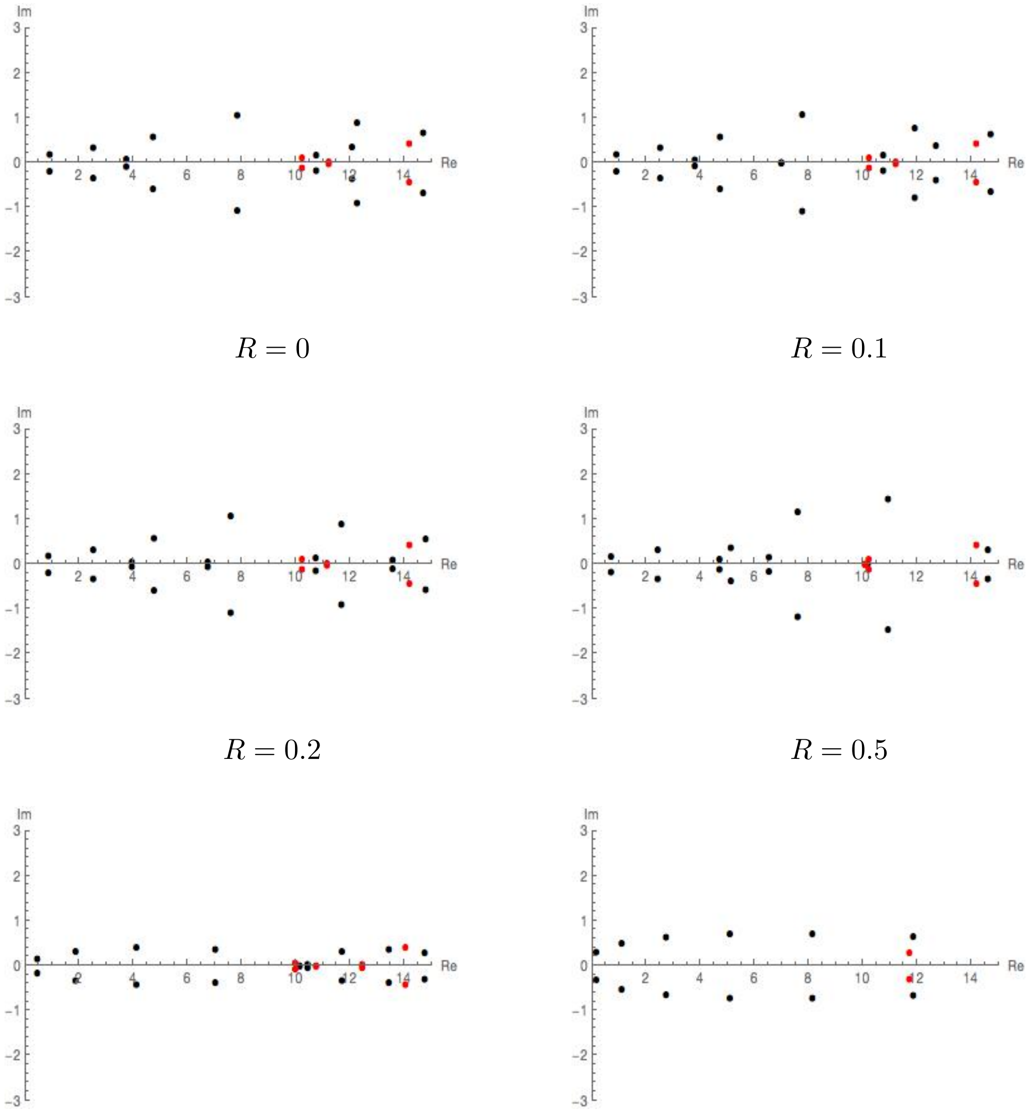


Figure 5.13: An example of some internal domains from waveguides with a single end; varying the obstacle radius.

$J = \{1\}$			
$R = 0$	$R = 0.1$	$R = 0.2$	$R = 0.5$
$0.9255 + 0.1957i$	$0.9151 + 0.1942i$	$0.8843 + 0.1896i$	$0.72284 + 0.1683i$
$2.5431 + 0.3334i$	$2.5420 + 0.3328i$	$2.5374 + 0.3309i$	$2.44862 + 0.3196i$
$3.7577 + 0.0802i$	$3.8122 + 0.0730i$	$3.9667 + 0.0509i$	$4.73258 + 0.1127i$
$4.7425 + 0.5764i$	$4.7513 + 0.5775i$	$4.7840 + 0.5778i$	$5.14458 + 0.3634i$
$7.8573 + 1.0573i$	$7.7798 + 1.0731i$	$6.7662 + 0.0535i$	$6.55368 + 0.1566i$
$10.7726 + 0.1699i$	$10.7738 + 0.1702i$	$7.6139 + 1.0741i$	$7.61349 + 1.1698i$
$12.0991 + 0.3521i$	$11.9394 + 0.7791i$	$10.7582 + 0.1504i$	$10.1935 + 0.0126i$
$12.2884 + 0.8944i$	$12.7264 + 0.3854i$	$11.7100 + 0.9037i$	$10.9546 + 1.4582i$
$14.7205 + 0.6654i$	$14.7399 + 0.6424i$	$13.6001 + 0.1020i$	$14.6424 + 0.3234i$
		$14.8140 + 0.5715i$	
1	1.5		
$0.4708 + 0.1652i$	$0.1683 + 0.3035i$		
$1.8868 + 0.3168i$	$1.1088 + 0.5125i$		
$4.1287 + 0.4188i$	$2.7548 + 0.6438i$		
$7.0520 + 0.3768i$	$5.1091 + 0.7114i$		
$10.1588 + 0.0025i$	$8.1600 + 0.7175i$		
$10.4560 + 0.0420i$	$11.8852 + 0.6586i$		
$11.7208 + 0.3288i$			
$13.4591 + 0.3708i$			
$14.7848 + 0.2970i$			

$J = \{2\}$			
$R = 0$	$R = 0.1$	$R = 0.2$	$R = 0.5$
$10.2536 + 0.1156i$	$10.2536 + 0.1156i$	$10.2537 + 0.1157i$	$10.1075 + 0.0077i$
$11.2435 + 0.0178i$	$11.2395 + 0.0177i$	$11.1788 + 0.0171i$	$10.2452 + 0.1138i$
$14.2167 + 0.4274i$	$14.2169 + 0.4273i$	$14.2174 + 0.427i$	$14.2245 + 0.4250i$
1	1.5		
$9.9999 + 0.0662i$	$11.7471 + 0.2963i$		
$10.7729 + 0.0065i$			
$12.4841 + 0.0345i$			
$14.0740 + 0.4083i$			



$R = 0$

$R = 0.1$

$R = 0.2$

$R = 0.5$

$R = 1.0$

$R = 1.5$

Figure 5.14: Some plots of the location of resonances with centrally located obstacles of different radii in the single ended case.

5.4.2 The two ended case

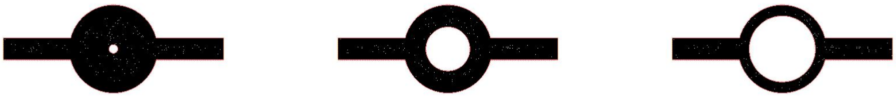
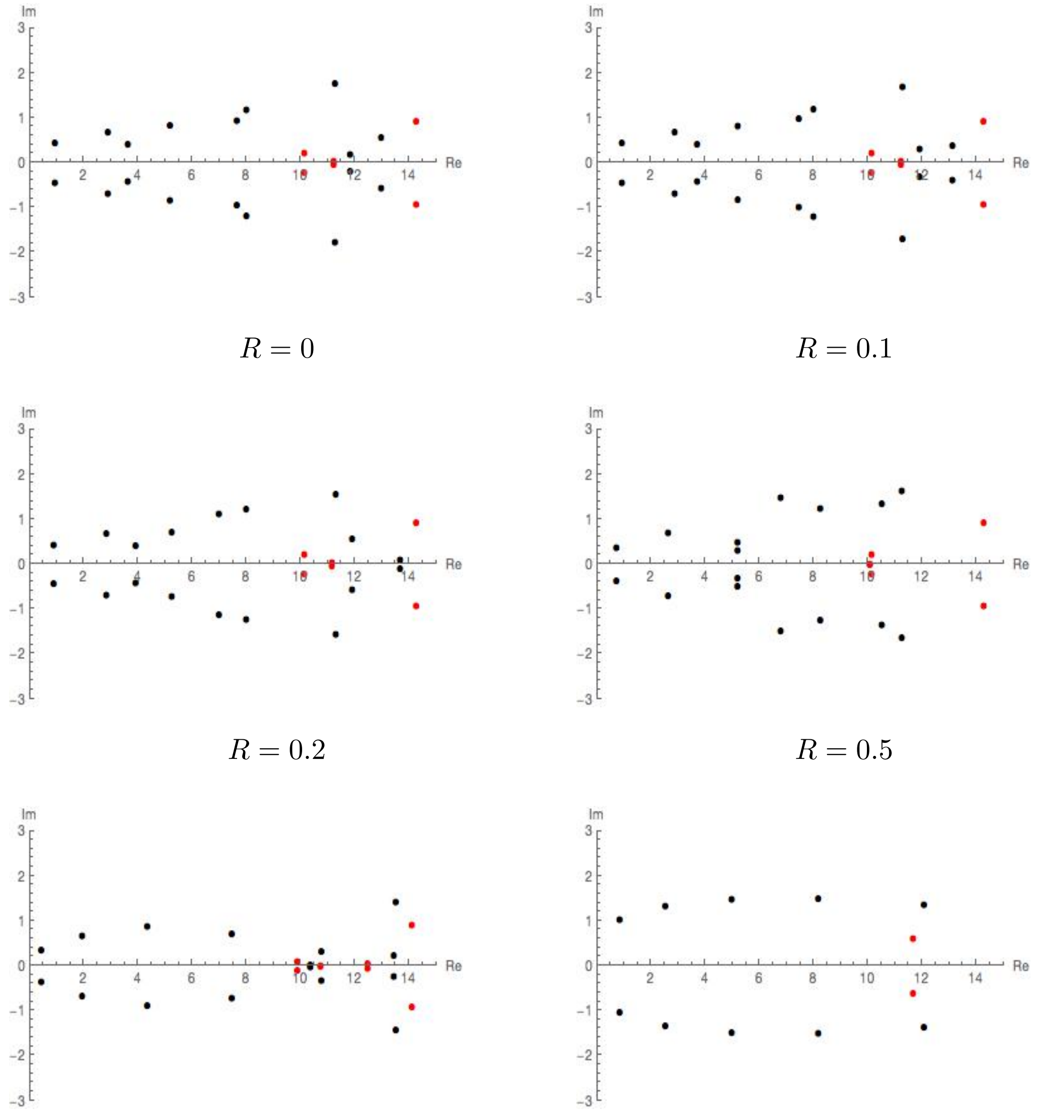


Figure 5.15: An example of some internal domains from waveguides with two ends; varying the obstacle radius.

$J = \{1\}$			
$R = 0$	$R = 0.1$	$R = 0.2$	$R = 0.5$
$0.9556 + 0.4519i$	$0.9448 + 0.4470i$	$0.9127 + 0.4326i$	$0.7412 + 0.3701i$
$2.9102 + 0.6890i$	$2.8937 + 0.6887i$	$2.8535 + 0.6919i$	$2.6540 + 0.7014i$
$3.6465 + 0.4109i$	$3.7229 + 0.4135i$	$3.9369 + 0.4135i$	$5.2090 + 0.3038i$
$5.2043 + 0.8424i$	$5.2215 + 0.8149i$	$5.2678 + 0.7164i$	$5.2118 + 0.4894i$
$7.6756 + 0.9436i$	$7.4808 + 0.9888i$	$7.0069 + 1.1196i$	$6.8060 + 1.4799i$
$8.0220 + 1.1812i$	$8.0192 + 1.1962i$	$8.0186 + 1.2313i$	$8.2708 + 1.2394i$
$11.8564 + 0.1919i$	$11.9416 + 0.3141i$	$11.9336 + 0.5642i$	$10.5429 + 1.3549i$
$11.2958 + 1.7761i$	$11.3061 + 1.7026i$	$11.3279 + 1.5569i$	$11.2757 + 1.6419i$
$13.0017 + 0.5615i$	$13.1531 + 0.3910i$	$13.6983 + 0.0910i$	
$R = 1$	$R = 1.5$		
$0.4537 + 0.3499i$	$0.8662 + 1.0339i$		
$1.9609 + 0.6718i$	$2.5511 + 1.3346i$		
$4.3654 + 0.8781i$	$4.9976 + 1.4890i$		
$7.4886 + 0.7113i$	$8.1929 + 1.4988i$		
$10.3866 + 0.0248i$	$12.0949 + 1.3574i$		
$10.7912 + 0.3194i$			
$13.4641 + 0.2284i$			
$13.5376 + 1.4311i$			

$J = \{2\}$			
$R = 0$	$R = 0.1$	$R = 0.2$	$R = 0.5$
$10.1725 + 0.2234i$	$10.1725 + 0.2234i$	$10.1726 + 0.2235i$	$10.1016 + 0.0143i$
$11.2516 + 0.0346i$	$11.2474 + 0.0345i$	$11.1865 + 0.0337i$	$10.1721 + 0.2234i$
$14.2966 + 0.9315i$	$14.2966 + 0.9315i$	$14.2973 + 0.9309i$	$14.3038 + 0.9249i$
$R = 1$	$R = 1.5$		
$9.9120 + 0.1040i$	$11.7001 + 0.6162i$		
$10.7696 + 0.0115i$			
$12.5095 + 0.0554i$			
$14.1336 + 0.9146i$			



$R = 0$

$R = 0.1$

$R = 0.2$

$R = 0.5$

$R = 1.0$

$R = 1.5$

Figure 5.16: Some colour coded plots of the location of resonances with centrally located obstacles of different radii in the two ended case.

5.4.3 Three ended case

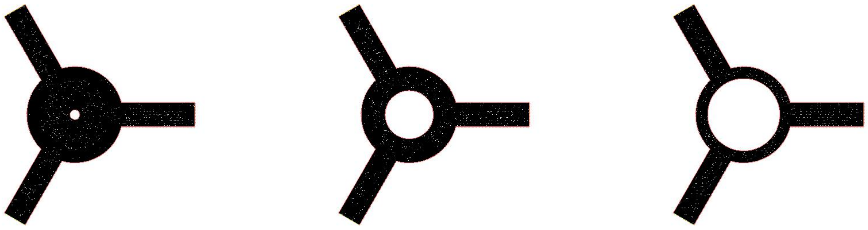
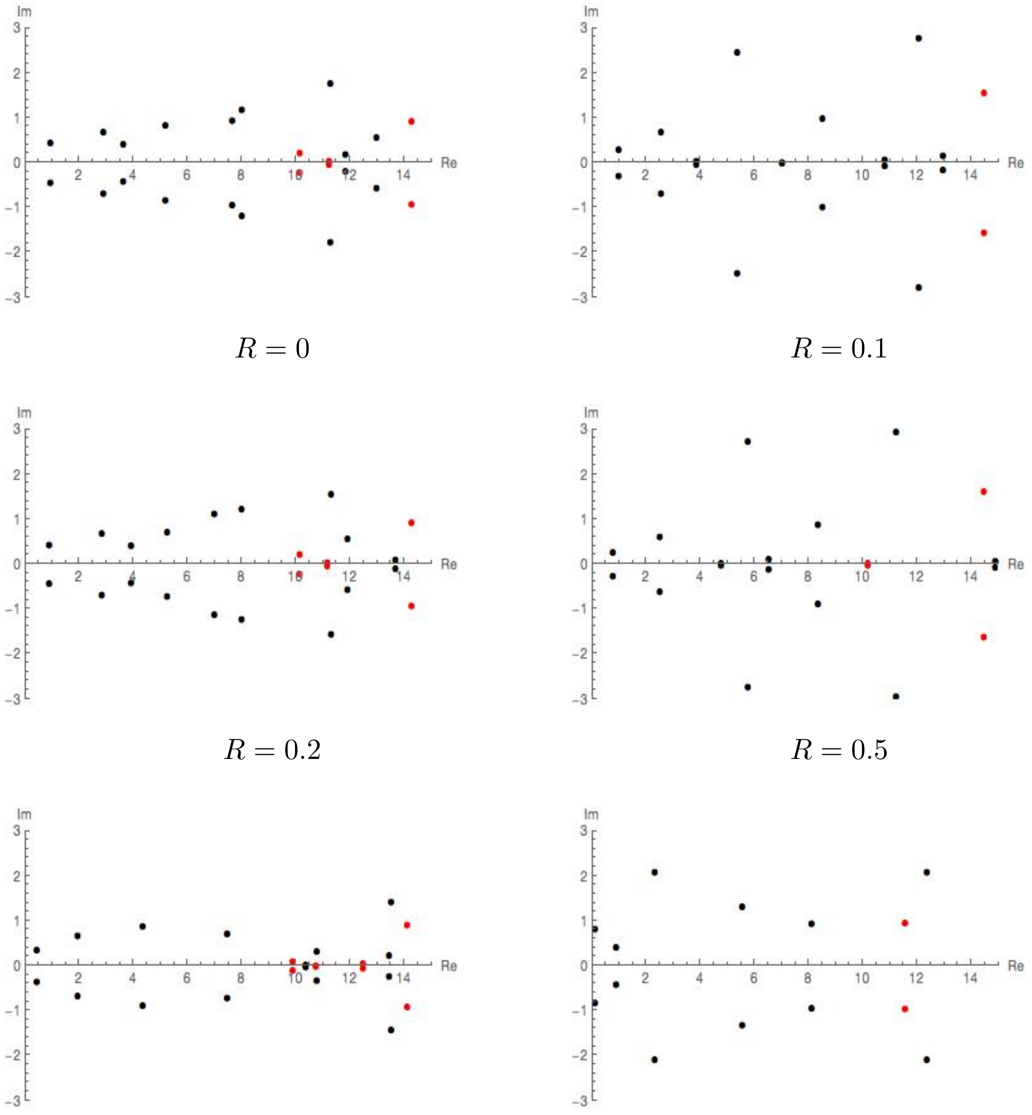


Figure 5.17: An example of some internal domains from waveguides with three ends; varying the obstacle radius.

$J = \{1\}$			
$R = 0$	$R = 0.1$	$R = 0.2$	$R = 0.5$
$1.0110 + 0.3023i$ $2.5604 + 0.6909i$ $3.8210 + 0.0438i$ $5.3757 + 2.4361i$ $8.5969 + 0.9766i$ $10.8319 + 0.0763i$ $12.6999 + 0.2015i$ $12.0468 + 2.8639i$	$0.9992 + 0.2998i$ $2.5641 + 0.6877i$ $3.8680 + 0.0385i$ $5.3817 + 2.4582i$ $7.0429 + 0.0110i$ $8.5259 + 0.9947i$ $10.8268 + 0.0743i$ $12.9799 + 0.1596i$ $12.0919 + 2.7766i$	$0.9642 + 0.2921i$ $2.5727 + 0.6779i$ $4.0022 + 0.0245i$ $5.4087 + 2.5183i$ $6.7829 + 0.0511i$ $8.3739 + 1.0045i$ $10.7913 + 0.0578i$ $12.1541 + 2.6009i$ $13.6786 + 0.0603i$	$0.7857 + 0.2582i$ $2.5182 + 0.6155i$ $4.7887 + 0.0252i$ $5.7730 + 2.7362i$ $6.5441 + 0.1110i$ $8.3653 + 0.8888i$ $11.2519 + 2.9488i$ $14.9104 + 0.0708i$
$R = 1$	$R = 1.5$		
$0.5276 + 0.2620i$ $1.8946 + 0.5344i$ $4.6503 + 1.3915i$ $7.4274 + 0.4451i$ $11.1490 + 0.6697i$ $11.7862 + 0.2202i$ $14.6938 + 0.0636i$	$0.1266 + 0.8249i$ $0.9068 + 0.4113i$ $2.3353 + 2.0831i$ $5.5668 + 1.3166i$ $8.1279 + 0.9452i$ $12.3883 + 2.0901i$		

$J = \{2\}$			
$R = 0$	$R = 0.1$	$R = 0.2$	$R = 0.5$
$14.4973 + 1.5573i$	$14.4972 + 1.5574i$	$14.4972 + 1.5577i$	$10.2020 + 0.0202i$ $14.4950 + 1.6162i$
$R = 1$	$R = 1.5$		
$10.3039 + 0.0913i$ $14.4266 + 1.6114i$	$11.5871 + 0.9552i$		



$R = 0$

$R = 0.1$

$R = 0.2$

$R = 0.5$

$R = 1.0$

$R = 1.5$

Figure 5.18: Some colour coded plots of the location of resonances with centrally located obstacles of different radii in the three ended case. In all three cases, the resonances move more erratically as the radius of the obstacle increases with particularly vigorous motion at certain times. For this reason, the small markers are unhelpful here. This has been animated and clip are available online (https://figshare.com/s/5c73783781b80265879a). A point of interest is that once the obstacle reaches a certain size, all resonances begin to move in unison towards certain discrete points on the real line.

5.5 Varying the position of an obstacle of radius 0.5

With the ends of fixed width and angles (where they exist equal), we will fix the radius of the obstacle at 0.5, translate it, by a vector δ; first in the y direction, then the x direction.

5.5.1 The single ended case

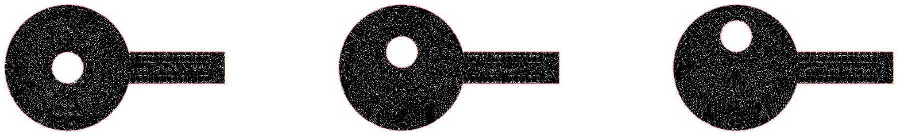
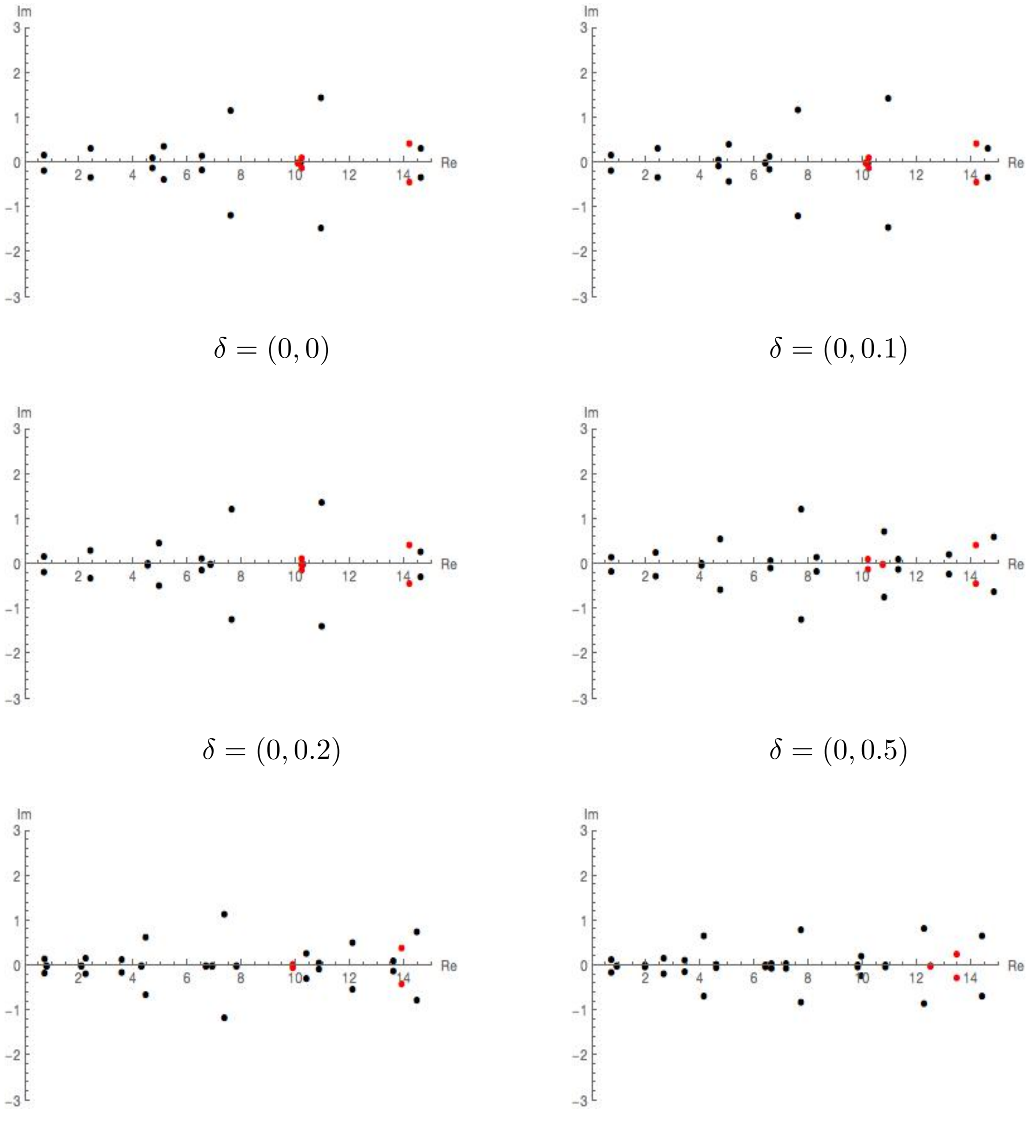


Figure 5.19: An example of some internal domains from waveguides with a single end; varying the vertical position of the obstacle.

$J = \{1\}$			
$\delta = (0,0)$	$\delta = (0,0.1)$	$\delta = (0,0.2)$	$\delta = (0,0.5)$
$0.7228 + 0.1683i$	$0.7230 + 0.1682i$	$0.7235 + 0.1678i$	$0.7269 + 0.1655i$
$2.4486 + 0.3196i$	$2.4450 + 0.3173i$	$2.4344 + 0.3106i$	$2.3669 + 0.2674i$
$4.7326 + 0.1126i$	$4.6878 + 0.0702i$	$4.5518 + 0.0159i$	$4.0631 + 0.0285i$
$5.1446 + 0.3634i$	$5.0712 + 0.4095i$	$4.9670 + 0.4813i$	$4.7605 + 0.5692i$
$6.5537 + 0.1566i$	$6.4279 + 0.0048i$	$6.5436 + 0.1213i$	$6.6118 + 0.0833i$
$7.6135 + 1.1698i$	$6.5722 + 0.1449i$	$6.8712 + 0.0075i$	$7.7488 + 1.2256i$
$10.1935 + 0.0126i$	$7.6224 + 1.1871i$	$7.6556 + 1.2323i$	$8.3154 + 0.1535i$
$10.9546 + 1.4589i$	$10.2133 + 0.0093i$	$10.2793 + 0.0020i$	$10.8161 + 0.7311i$
$14.6424 + 0.3232i$	$10.9615 + 1.4410i$	$10.9825 + 1.3823i$	$11.3380 + 0.1179i$
	$14.6403 + 0.3168i$	$14.6292 + 0.2848i$	$13.2054 + 0.2212i$
			$14.8662 + 0.6091i$
$\delta = (0,1)$	$\delta = (0,1.4)$		
$0.8157 + 0.0003i$	$0.7338 + 0.1499i$		
$0.7369 + 0.1591i$	$0.9400 + 0.0030i$		
$2.1027 + 0.0006i$	$1.9774 + 0.0170i$		
$2.2612 + 0.1758i$	$2.6705 + 0.1760i$		
$3.5906 + 0.1372i$	$3.4467 + 0.1314i$		
$4.3189 + 0.0009i$	$4.1512 + 0.6733i$		
$4.4729 + 0.6474i$	$4.6044 + 0.0334i$		
$6.7028 + 0.0023i$	$6.4204 + 0.0244i$		
$6.9348 + 0.0038i$	$6.6509 + 0.0521i$		
$7.8345 + 0.0062i$	$7.1855 + 0.0496i$		
$7.3849 + 1.1557i$	$7.7420 + 0.8023i$		
$10.4159 + 0.2818i$	$9.8379 + 0.0272i$		
$10.8786 + 0.0644i$	$9.9630 + 0.2118i$		
$12.1136 + 0.5138i$	$10.8567 + 0.0273i$		
$13.6234 + 0.1082i$	$12.2886 + 0.8418i$		
$14.4912 + 0.7606i$	$14.4295 + 0.6705i$		
$J = \{2\}$			
$\delta = (0,0)$	$\delta = (0,0.1)$	$\delta = (0,0.2)$	$\delta = (0,0.5)$
$10.1075 + 0.0077i$	$10.1422 + 0.0072i$	$10.2402 + 0.0014i$	$10.7611 + 0.0123i$
$10.2452 + 0.1138i$	$10.2435 + 0.1148i$	$10.2417 + 0.1216i$	$10.2196 + 0.1112i$
$14.2245 + 0.4250i$	$14.2246 + 0.4251i$	$14.2246 + 0.4258i$	$14.1989 + 0.4334i$
$\delta = (0,1)$	$\delta = (0,1.4)$		
$9.9183 + 0.0432i$	$12.5301 + 0.0065i$		
$13.9308 + 0.3962i$	$13.4939 + 0.2635i$		



$\delta = (0, 0)$

$\delta = (0, 0.1)$

$\delta = (0, 0.2)$

$\delta = (0, 0.5)$

$\delta = (0, 1)$

$\delta = (0, 1.4)$

Figure 5.20: Plots showing the location of the resonances for the three sheets as the obstacle is moved in the y direction.

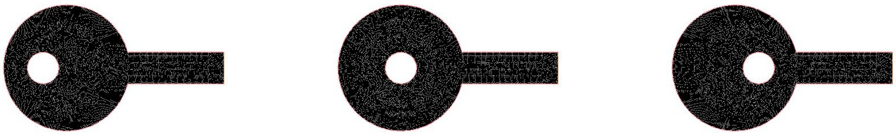
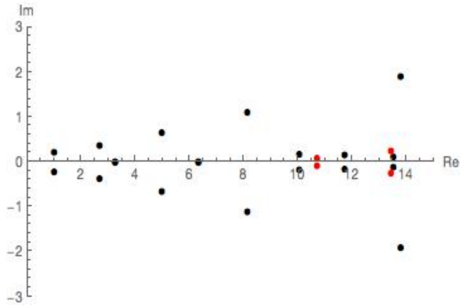
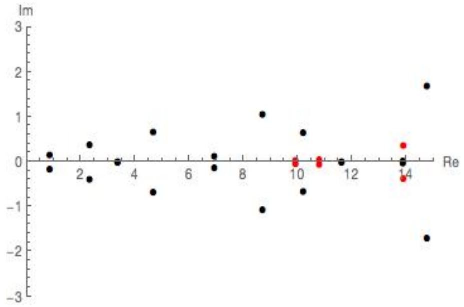


Figure 5.21: An example of some internal domains from waveguides with a single end; varying the horizontal position of the obstacle.

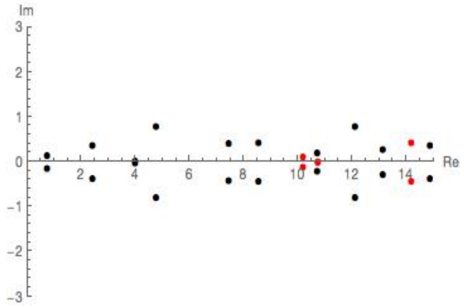
$J = \{1\}$			
-1.4	-1	-0.5	-0.2
$1.0222 + 0.2165i$	$0.8728 + 0.1512i$	$0.7549 + 0.1377i$	$0.7270 + 0.1514i$
$2.6956 + 0.3753i$	$2.3424 + 0.3880i$	$2.4379 + 0.3748i$	$2.4626 + 0.3346i$
$3.2732 + 0.0118i$	$3.3779 + 0.0011i$	$4.0149 + 0.0166i$	$4.5173 + 0.0140i$
$4.9982 + 0.6553i$	$4.6885 + 0.6671i$	$4.7881 + 0.7985i$	$5.1374 + 0.8093i$
$6.3463 + 0.0012i$	$6.9434 + 0.1284i$	$7.4654 + 0.4123i$	$6.8889 + 0.0100i$
$8.1574 + 1.1108i$	$8.7223 + 1.0651i$	$8.5700 + 0.4235i$	$7.5298 + 0.9330i$
$10.0743 + 0.1693i$	$10.2260 + 0.6504i$	$10.7400 + 0.2029i$	$10.2827 + 0.0021i$
$11.7517 + 0.1644i$	$11.6403 + 0.0073i$	$12.1332 + 0.7969i$	$11.0462 + 1.1096i$
$13.5506 + 0.1174i$	$13.9068 + 0.0185i$	$13.1575 + 0.2820i$	$14.4983 + 0.1285i$
$13.8170 + 1.9108i$	$14.7920 + 1.6958i$	$14.9036 + 0.3622i$	
-0.1	0	0.1	0.2
$0.7235 + 0.1590i$	$0.7228 + 0.1683i$	$0.7250 + 0.1794i$	$0.73019 + 0.1925i$
$2.4583 + 0.3253i$	$2.4486 + 0.3196i$	$2.4339 + 0.3179i$	$2.41459 + 0.3212i$
$4.6444 + 0.0562i$	$4.7326 + 0.1126i$	$4.7822 + 0.0709i$	$4.60956 + 0.0050i$
$5.2553 + 0.6274i$	$5.1446 + 0.3634i$	$4.8467 + 0.3207i$	$4.72489 + 0.3767i$
$6.5746 + 0.0883i$	$6.5537 + 0.1566i$	$6.7403 + 0.0874i$	$6.97491 + 0.0108i$
$7.5108 + 1.0358i$	$7.6135 + 1.1698i$	$7.8554 + 1.2193i$	$8.10826 + 1.0813i$
$10.2097 + 0.0093i$	$10.1935 + 0.0126i$	$10.2231 + 0.0091i$	$10.2860 + 0.0020i$
$10.9721 + 1.2496i$	$10.9546 + 1.4589i$	$11.0539 + 1.7736i$	$11.4426 + 2.1556i$
$14.5846 + 0.2358i$	$14.6424 + 0.3232i$	$14.6990 + 0.3802i$	$14.7446 + 0.3539i$
0.5	1	1.4	
$0.7674 + 0.2468i$	$1.0102 + 0.3480i$	$1.1140 + 0.1237i$	
$2.3354 + 0.3731i$	$2.3919 + 1.0966i$	$2.9288 + 0.1006i$	
$3.9822 + 0.0065i$	$3.3878 + 0.0409i$	$3.5587 + 0.1320i$	
$4.5723 + 0.4418i$	$4.1455 + 1.0430i$	$5.5494 + 0.2310i$	
$7.4270 + 0.3564i$	$6.6443 + 0.9726i$	$6.8410 + 0.0962i$	
$8.4904 + 0.2103i$	$7.8119 + 0.0374i$	$8.7682 + 0.3389i$	
$10.5693 + 0.0919i$	$10.0088 + 0.9315i$	$11.3862 + 0.0109i$	
$12.7211 + 0.4540i$	$11.6363 + 0.0002i$	$12.6191 + 0.4425i$	
$13.1348 + 0.3870i$	$13.3636 + 0.0486i$		
$14.5575 + 0.2097i$	$14.0624 + 0.7323i$		
$J = \{2\}$			
-1.4	-1	-0.5	-0.2
$10.7373 + 0.0755i$	$9.9428 + 0.0433i$	$10.2165 + 0.1125i$	$10.2414 + 0.0021i$
$13.4628 + 0.2454i$	$10.8172 + 0.0504i$	$10.7621 + 0.0083i$	$10.2413 + 0.1191i$
	$13.9285 + 0.3677i$	$14.2091 + 0.4366i$	$14.2279 + 0.4247i$
-0.1	0	0.1	0.2
$10.1418 + 0.0065i$	$10.1075 + 0.0077i$	$10.1429 + 0.0085i$	$10.2394 + 0.0011i$
$10.2445 + 0.1145i$	$10.2452 + 0.1138i$	$10.2420 + 0.1148i$	$10.2410 + 0.1256i$
$14.2259 + 0.4244i$	$14.2245 + 0.4250i$	$14.2244 + 0.4251i$	$14.2257 + 0.4233i$
0.5	1	1.4	
$10.2245 + 0.1030i$	$9.9238 + 0.0498i$	$10.7435 + 0.0041i$	
$10.7572 + 0.0282i$	$10.8234 + 0.0013i$	$13.5869 + 1.2249i$	
$14.2180 + 0.3811i$	$13.9119 + 0.4090i$		



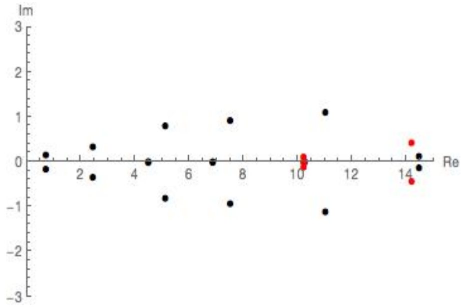
$\delta = (-1.4, 0)$



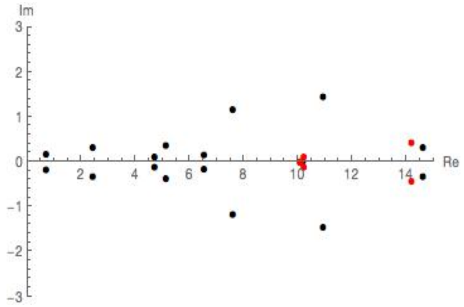
$\delta = (-1, 0)$



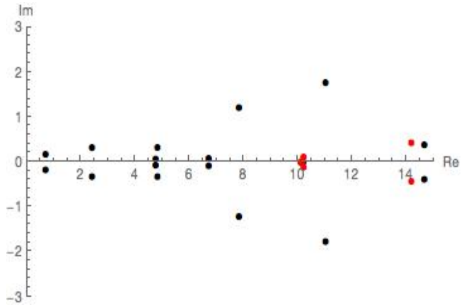
$\delta = (-0.5, 0)$



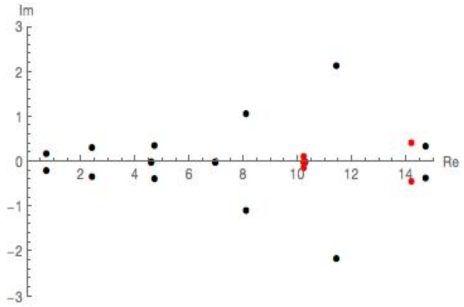
$\delta = (-0.2, 0)$



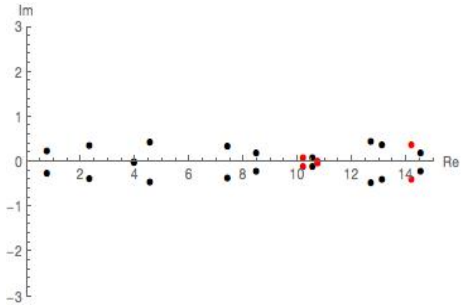
$\delta = (-0.1, 0)$



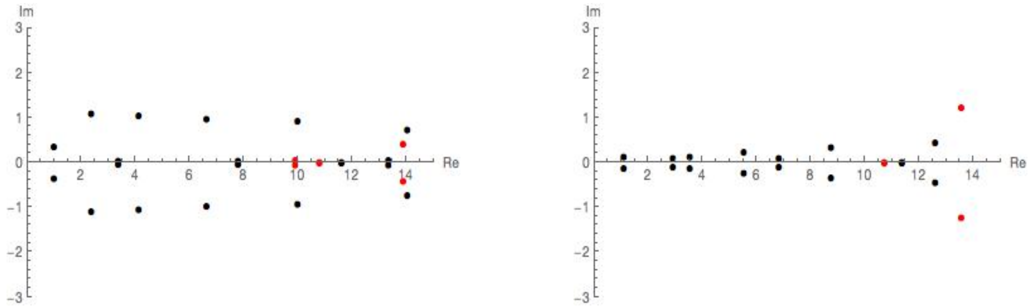
$\delta = (0, 0)$



$\delta = (0.1, 0)$



$\delta = (0.5, 0)$



$\delta = (1, 0)$

$\delta = (1.4, 0)$

Figure 5.22: Plots showing the location of the resonances for the three sheets as the obstacle is moved in the x direction.

5.5.2 The two ended case

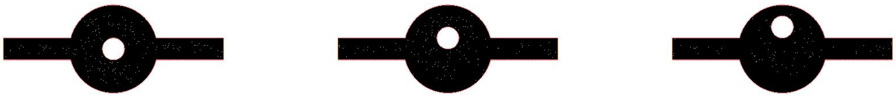
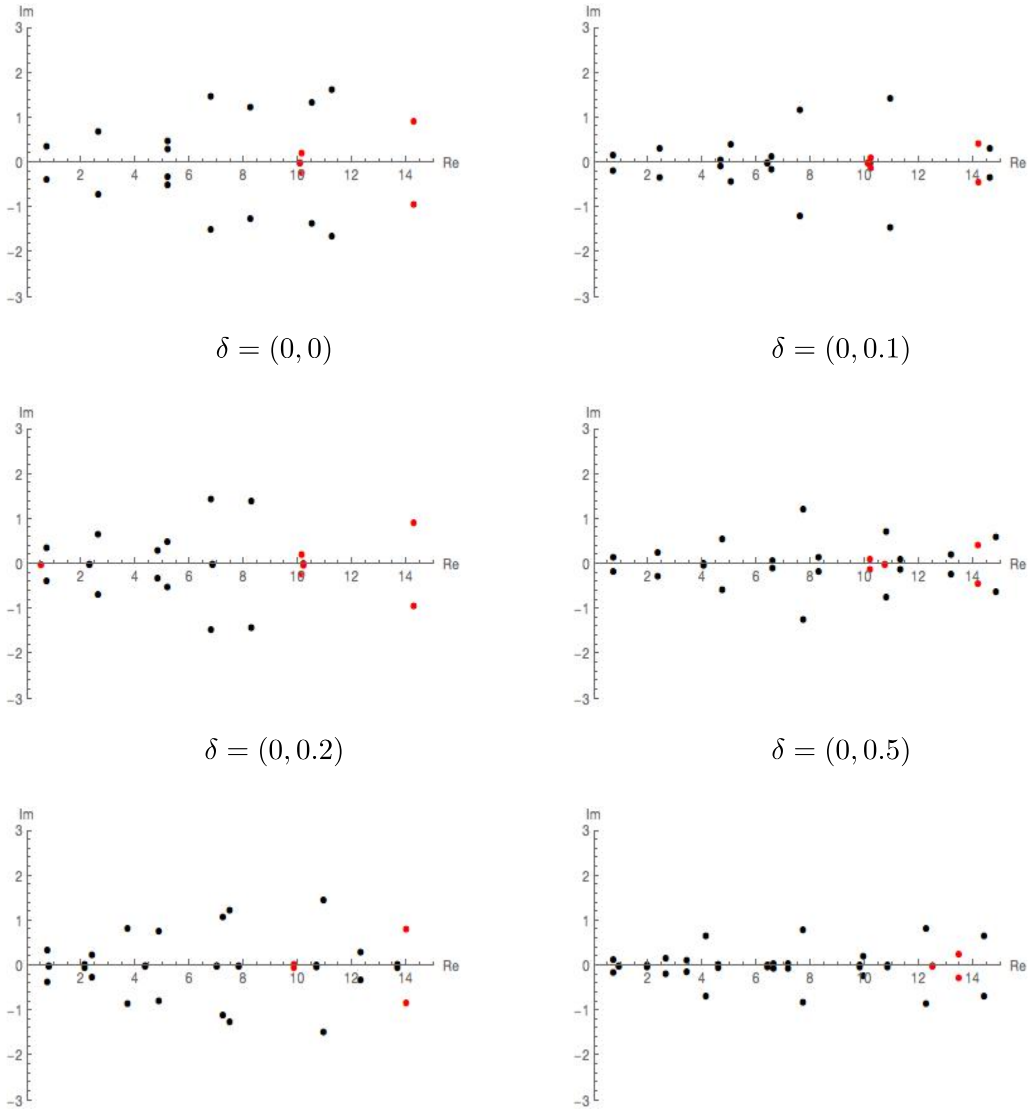


Figure 5.23: An example of some internal domains from waveguides with two ends; varying the vertical position of the obstacle.

$J = \{1\}$			
$\delta = (0, 0)$	$\delta = (0, 0.1)$	$\delta = (0, 0.2)$	$\delta = (0, 0.5)$
$0.7412 + 0.3701i$ $2.6540 + 0.7014i$ $5.2090 + 0.3038i$ $5.2118 + 0.4894i$ $6.8060 + 1.4799i$ $8.2708 + 1.2394i$	$0.7415 + 0.3699i$ $2.3321 + 0.0004i$ $2.6517 + 0.6943i$ $5.0847 + 0.2988i$ $5.2108 + 0.4947i$ $6.4576 + 0.0075i$ $6.8089 + 1.4735i$ $8.2799 + 1.2812i$	$0.7424 + 0.3691i$ $2.3243 + 0.0016i$ $2.6446 + 0.6732i$ $4.8398 + 0.3140i$ $5.2075 + 0.5104i$ $6.8753 + 0.0032i$ $6.8179 + 1.4545i$ $8.3053 + 1.4024i$	$0.7487 + 0.3639i$ $2.2747 + 0.0097i$ $2.5884 + 0.5326i$ $4.1928 + 0.4792i$ $4.4816 + 0.0001i$ $5.1636 + 0.6086i$ $6.8976 + 1.3343i$ $8.1725 + 0.1664i$ $11.4557 + 0.1122i$ $13.3686 + 0.1799i$
$\delta = (0, 1)$	$\delta = (0, 1.4)$		
$0.8243 + 0.0005i$ $0.7669 + 0.3492i$ $2.1449 + 0.0355i$ $2.4196 + 0.2518i$ $3.7307 + 0.8344i$ $4.3826 + 0.0016i$ $4.8888 + 0.7709i$ $7.0332 + 0.0052i$ $7.8400 + 0.0102i$ $7.2547 + 1.0979i$ $7.5023 + 1.2409i$ $10.7093 + 0.0183i$ $10.9728 + 1.4674i$ $12.3397 + 0.3141i$ $13.7010 + 0.0439i$	$0.7716 + 0.3274i$ $0.9529 + 0.0044i$ $2.0128 + 0.0711i$ $2.8646 + 0.1576i$ $3.5064 + 1.0050i$ $4.5129 + 0.7434i$ $4.6833 + 0.0488i$ $6.5208 + 0.1096i$ $6.8180 + 0.0371i$ $7.6191 + 0.9962i$ $7.8962 + 0.7465i$ $10.3594 + 0.2451i$ $12.7574 + 0.5214i$		
$J = \{2\}$			
$\delta = (0, 0)$	$\delta = (0, 0.1)$	$\delta = (0, 0.2)$	$\delta = (0, 0.5)$
$10.1016 + 0.0142i$ $10.1721 + 0.2234i$ $14.3038 + 0.9249i$	$10.1353 + 0.0151i$ $10.1714 + 0.2234i$ $14.3041 + 0.9252i$	$10.2340 + 0.0171i$ $10.1691 + 0.2231i$ $14.3044 + 0.9266i$	$10.1356 + 0.2078i$ $10.7664 + 0.0267i$ $14.2807 + 0.9385i$
$\delta = (0, 1)$	$\delta = (0, 1.4)$		
$9.8802 + 0.0315i$ $14.0230 + 0.8172i$	$9.9015 + 0.0009i$ $12.5366 + 0.0112i$ $13.7606 + 0.4991i$		



$\delta = (0, 0)$ $\delta = (0, 0.1)$

$\delta = (0, 0.2)$ $\delta = (0, 0.5)$

$\delta = (0, 1)$ $\delta = (0, 1.4)$

Figure 5.24: Some plots showing the location of the resonances as the obstacle is moved in the y direction in the two ended case. We can see that, unlike in the singe ended case, moving the obstacle destroys the symmetry of the domain and causes embedded eigenvalues to decay to resonances.

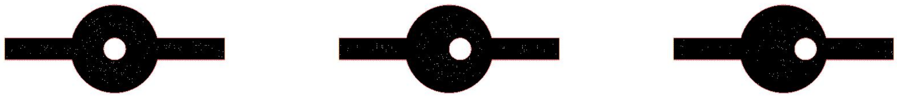
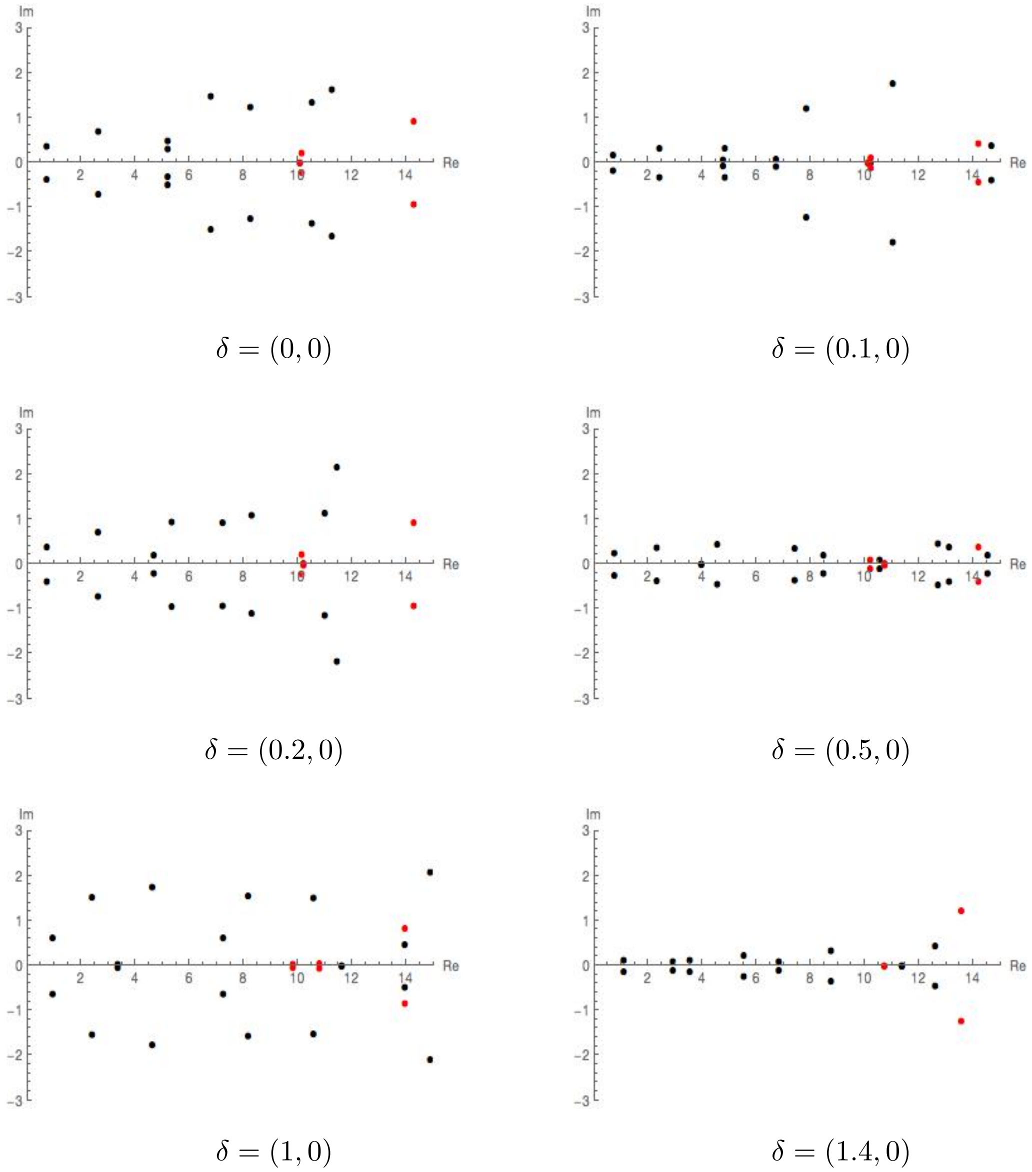


Figure 5.25: An example of some internal domains from waveguides with a two ends; varying the horizontal position of the obstacle.

$J = \{1\}$			
$\delta = (0, 0)$	$\delta = (0.1, 0)$	$\delta = (0.2, 0)$	$\delta = (0.5, 0)$
$0.7412 + 0.3701i$ $2.6540 + 0.7014i$ $5.2090 + 0.3038i$ $5.2118 + 0.4894i$ $6.8060 + 1.4799i$ $8.2708 + 1.2394i$ $10.5429 + 1.3549i$ $11.2757 + 1.6419i$	$0.7423 + 0.3722i$ $2.6513 + 0.7064i$ $4.9678 + 0.2570i$ $5.3879 + 0.6558i$ $6.8994 + 1.2908i$ $8.2906 + 1.2048i$ $10.7324 + 1.3443i$ $11.2279 + 1.7507i$	$0.7458 + 0.3786i$ $2.6430 + 0.7213i$ $4.7031 + 0.1972i$ $5.3651 + 0.9373i$ $7.2405 + 0.9213i$ $8.3159 + 1.0858i$ $11.0150 + 1.1390i$ $11.4670 + 2.1666i$	$0.7716 + 0.4262i$ $2.5748 + 0.8261i$ $4.0017 + 0.1106i$ $5.0600 + 1.1587i$ $7.9022 + 0.9331i$ $8.7312 + 0.3562i$ $11.4731 + 1.0368i$ $13.3491 + 0.4910i$ $13.0244 + 0.7479i$
$\delta = (1, 0)$	$\delta = (1.4, 0)$		
$0.9619 + 0.6279i$ $2.4196 + 1.5252i$ $3.3663 + 0.0313i$ $4.6408 + 1.7521i$ $7.2658 + 0.6318i$ $8.1850 + 1.5538i$ $10.6018 + 1.5176i$ $11.6388 + 0.0077i$ $13.9743 + 0.4695i$ $14.9101 + 2.0925i$	$1.2088 + 0.4344i$ $3.1540 + 0.1222i$ $3.6774 + 0.7649i$ $5.9924 + 0.3516i$ $7.1447 + 1.0036i$ $9.3574 + 0.9252i$ $11.6548 + 0.2055i$ $12.5919 + 0.3646i$ $13.8100 + 2.0111i$		

$J = \{2\}$			
$\delta = (0, 0)$	$\delta = (0.1, 0)$	$\delta = (0.2, 0)$	$\delta = (0.5, 0)$
$10.1016 + 0.0142i$ $10.1721 + 0.2234i$ $14.3038 + 0.9249i$	$10.1352 + 0.0157i$ $10.1716 + 0.2232i$ $14.3034 + 0.9249i$	$10.2330 + 0.0195i$ $10.1704 + 0.2226i$ $14.3021 + 0.9246i$	$10.1424 + 0.2054i$ $10.7656 + 0.0385i$ $14.2818 + 0.9025i$
$\delta = (1, 0)$	$\delta = (1.4, 0)$		
$9.8450 + 0.0336i$ $10.8212 + 0.0507i$ $13.9816 + 0.8366i$	$10.7451 + 0.0794i$ $13.5035 + 1.5953i$		



$\delta = (0, 0)$

$\delta = (0.1, 0)$

$\delta = (0.2, 0)$

$\delta = (0.5, 0)$

$\delta = (1, 0)$

$\delta = (1.4, 0)$

Figure 5.26: Some plots showing the location of the resonances as the obstacle is moved in the x direction in the two ended case. In this case, moving the obstacle in the x direction doesn't destroy symmetry to the same extent as in the y direction, and we don't immediately see new resonances formed from decaying embedded eigenvalues.

5.5.3 The three ended case

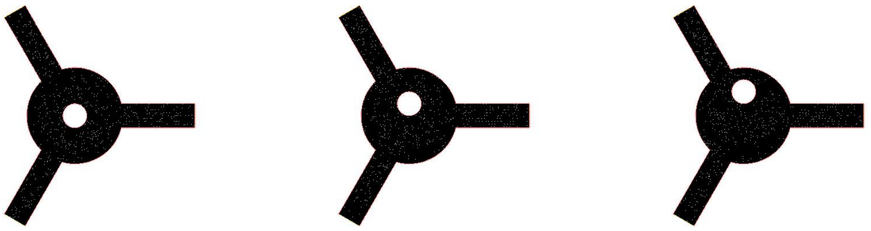
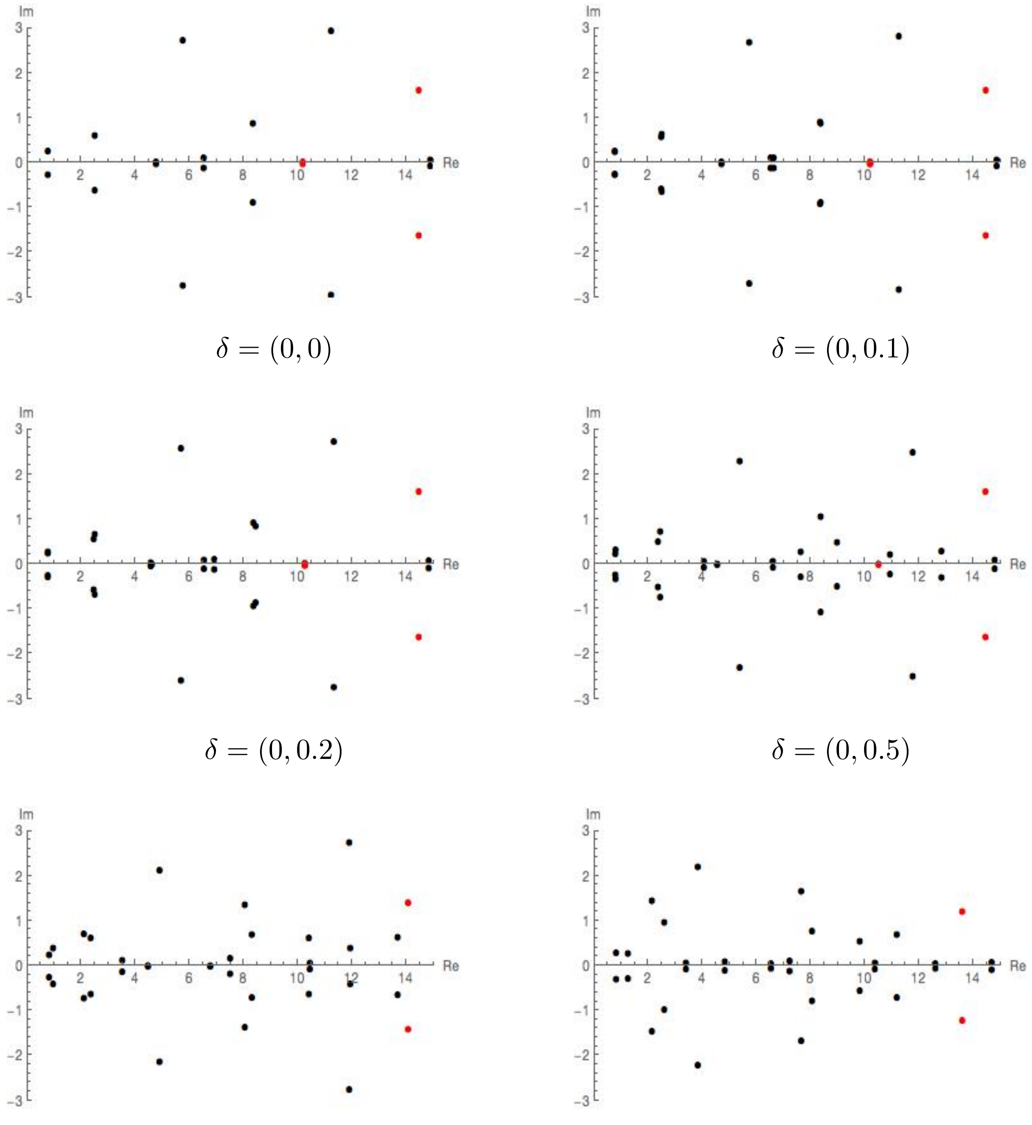


Figure 5.27: An example of some internal domains from waveguides with three ends; varying the vertical position of the obstacle.

$J = \{1\}$			
$\delta = (0,0)$	$\delta = (0,0.1)$	$\delta = (0,0.2)$	$\delta = (0,0.5)$
$0.7857 + 0.2582i$ $2.5182 + 0.6155i$ $4.7887 + 0.0252i$ $5.7730 + 2.7362i$ $6.5441 + 0.1110i$ $8.3653 + 0.8888i$ $11.2518 + 2.9486i$ $14.9104 + 0.0708i$	$0.7870 + 0.2498i$ $0.7863 + 0.2678i$ $2.5042 + 0.5866i$ $2.5240 + 0.6449i$ $4.7296 + 0.0284i$ $5.7577 + 2.6940i$ $6.5472 + 0.1091i$ $6.6563 + 0.1138i$ $8.3686 + 0.9081i$ $8.3892 + 0.8755i$ $11.2818 + 2.8222i$ $14.8967 + 0.0717i$	$0.7899 + 0.2428i$ $0.7889 + 0.2788i$ $2.4827 + 0.5598i$ $2.5209 + 0.6736i$ $4.5855 + 0.0357i$ $4.6226 + 0.0008i$ $5.7035 + 2.5887i$ $6.5583 + 0.1042i$ $6.9370 + 0.1190i$ $8.3842 + 0.9327i$ $8.4629 + 0.8586i$ $11.3515 + 2.7356i$ $14.8513 + 0.0846i$	$0.8063 + 0.2315i$ $0.8135 + 0.3188i$ $2.3847 + 0.5047i$ $2.4607 + 0.7370i$ $4.0750 + 0.0662i$ $4.5678 + 0.0028i$ $5.3970 + 2.2979i$ $6.6300 + 0.0742i$ $7.6579 + 0.2786i$ $8.9914 + 0.4912i$ $8.3916 + 1.0652i$ $10.9543 + 0.2226i$ $11.7925 + 2.4953i$ $12.8555 + 0.2927i$ $14.8184 + 0.0992i$
$\delta = (0,1)$	$\delta = (0,1.4)$		
$0.8363 + 0.2549i$ $0.9847 + 0.4056i$ $2.1161 + 0.7152i$ $2.3674 + 0.6254i$ $3.5335 + 0.1248i$ $4.4867 + 0.0104i$ $4.9140 + 2.1375i$ $6.7836 + 0.0136i$ $7.5278 + 0.1717i$ $8.3286 + 0.6988i$ $8.0627 + 1.3637i$ $10.4712 + 0.0738i$ $10.4274 + 0.6301i$ $11.9549 + 0.4040i$ $11.9206 + 2.7510i$ $13.7051 + 0.6438i$	$0.8298 + 0.2946i$ $1.2676 + 0.2862i$ $2.6131 + 0.9794i$ $2.1611 + 1.4600i$ $3.4163 + 0.0740i$ $3.8523 + 2.2149i$ $4.8536 + 0.0954i$ $6.5543 + 0.0550i$ $7.2450 + 0.1168i$ $7.6758 + 1.6590i$ $8.0726 + 0.7751i$ $9.8347 + 0.5492i$ $10.3957 + 0.0704i$ $11.1995 + 0.7058i$ $12.6264 + 0.0533i$ $14.7061 + 0.0763i$		
$J = \{2\}$			
$\delta = (0,0)$	$\delta = (0,0.1)$	$\delta = (0,0.2)$	$\delta = (0,0.5)$
$10.2020 + 0.0202i$ $14.4950 + 1.6162i$	$10.2192 + 0.0201i$ $10.2250 + 0.0249i$ $14.4940 + 1.6189i$ $3.0647 + 0.0001i$	$3.0580 + 0.0005i$ $10.2787 + 0.0246i$ $10.2877 + 0.0296i$ $14.4906 + 1.6260i$	$10.5338 + 0.0132i$ $14.4798 + 1.6150i$
$\delta = (0,1)$	$\delta = (0,1.4)$		
$14.0978 + 1.4082i$	$13.6302 + 1.2088i$		



$\delta = (0, 0)$

$\delta = (0, 0.1)$

$\delta = (0, 0.2)$

$\delta = (0, 0.5)$

$\delta = (0, 1)$

$\delta = (0, 1.4)$

Figure 5.28: Some plots showing the location of the resonances as the obstacle is moved in the y direction in the three ended case. Here we are able to observe the bifurcation of some resonances the obstacle is moved from the centre.

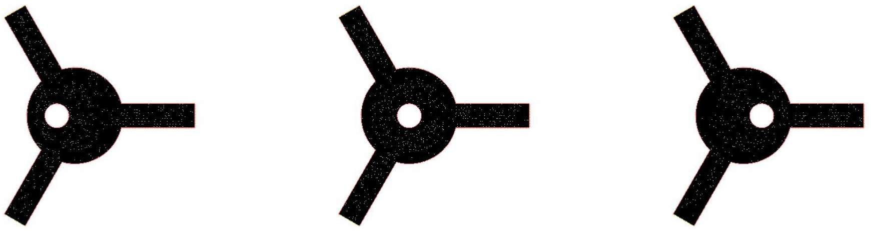
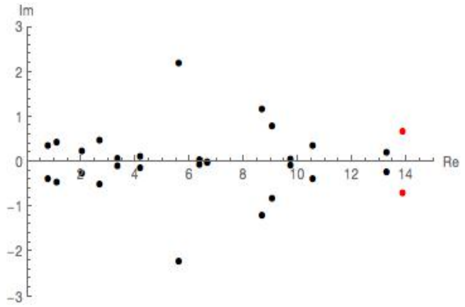


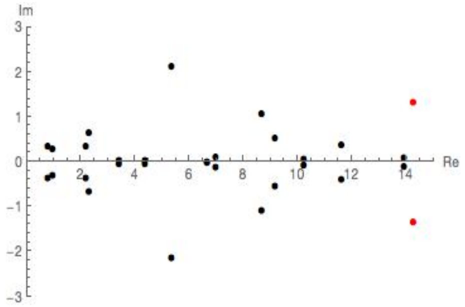
Figure 5.29: An example of some internal domains from waveguides with three ends; varying the horizontal position of the obstacle.

$J=\{1\}$			
$\delta=(-1.4,0)$	$\delta=(-1,0)$	$\delta=(-0.5,0)$	$\delta=(-0.2,0)$
$0.7873+0.3652i$	$0.7962+0.3487i$	$0.8289+0.2378i$	$0.7930+0.2442i$
$1.1153+0.4457i$	$0.9697+0.2885i$	$0.7879+0.3066i$	$0.7856+0.2770i$
$2.0490+0.2501i$	$2.2066+0.3583i$	$2.3924+0.4731i$	$2.4841+0.5562i$
$2.6925+0.4968i$	$2.3148+0.6485i$	$2.4480+0.7393i$	$2.5189+0.6757i$
$3.3620+0.0785i$	$3.4217+0.0431i$	$4.0120+0.0136i$	$4.5741+0.0278i$
$4.1929+2.2096i$	$5.3573+2.1341i$	$5.5097+2.2797i$	$5.7234+2.5863i$
$6.3912+0.0461i$	$6.6786+0.0039i$	$6.6261+0.0678i$	$6.5718+0.1047i$
$6.6761+0.0142i$	$6.9882+0.1134i$	$7.5044+0.2146i$	$6.9325+0.1252i$
$8.6966+1.1772i$	$8.6909+1.0842i$	$8.5055+0.9751i$	$8.3954+0.8893i$
$9.0696+0.8080i$	$9.1826+0.5290i$	$9.7626+0.4315i$	$8.5124+0.9378i$
$9.7527+0.0743i$	$10.2396+0.0607i$	$11.4442+0.3104i$	$14.8397+0.0964i$
$10.5725+0.3687i$	$11.6292+0.3872i$	$13.0882+0.1067i$	
$13.2967+0.2212i$	$13.9393+0.1028i$		
$\delta=(-0.1,0)$	$\delta=(0,0)$	$\delta=(0.1,0)$	$\delta=(0.2,0)$
$0.7878+0.2502i$	$0.7857+0.2582i$	$0.7863+0.2494i$	$0.7873+0.2411i$
$0.7855+0.2674i$	$2.5182+0.6155i$	$0.7870+0.2683i$	$0.7916+0.2808i$
$2.5046+0.5858i$	$4.7887+0.0252i$	$2.5038+0.5875i$	$2.4811+0.5634i$
$2.5235+0.6456i$	$5.7730+2.7362i$	$2.5245+0.6443i$	$2.5232+0.6713i$
$4.6317+0.0001i$	$6.5441+0.1110i$	$4.7316+0.0297i$	$4.5981+0.0447i$
$4.7276+0.0270i$	$8.3653+0.8888i$	$5.7546+2.6940i$	$4.6220+0.0003i$
$5.7608+2.6939i$	$11.2518+2.9486i$	$6.5397+0.1083i$	$5.6841+2.5904i$
$6.5556+0.1099i$	$14.9104+0.0708i$	$6.6618+0.1132i$	$6.5440+0.1026i$
$6.6500+0.1145i$		$8.3938+0.8672i$	$6.9417+0.1132i$
$8.3760+0.8808i$		$8.3578+0.9117i$	$8.3447+0.9422i$
$8.3881+0.9076i$		$14.8974+0.0706i$	$8.4478+0.8156i$
$14.8959+0.0729i$			$14.8607+0.0710i$
$\delta=(0.5,0)$	$\delta=(1,0)$	$\delta=(1.4,0)$	
$0.7930+0.2199i$	$0.8136+0.1988i$	$0.8459+0.1987i$	
$0.8302+0.3365i$	$1.1194+0.4407i$	$1.1897+0.1824i$	
$2.3719+0.5387i$	$2.3421+0.7619i$	$2.2623+0.7388i$	
$2.4791+0.7338i$	$2.1522+1.1817i$	$3.3234+0.7410i$	
$4.1512+0.1274i$	$3.6979+0.2156i$	$3.6111+0.1130i$	
$4.5589+0.0020i$	$4.3883+0.0128i$	$4.2666+0.0284i$	
$5.2895+2.2934i$	$4.4419+2.3607i$	$6.8529+0.2168i$	
$6.6167+0.0750i$	$6.8669+0.0191i$	$6.9969+0.0005i$	
$7.8457+0.4308i$	$7.8427+0.1651i$	$6.3864+1.9588i$	
$8.5287+0.2866i$	$7.8404+0.7543i$	$7.8151+0.6937i$	
$8.2125+0.9667i$	$7.8729+1.6446i$	$8.9574+0.3020i$	
$10.5532+0.0046i$	$10.0236+1.1017i$	$10.6537+0.0187i$	
$10.7212+0.0814i$	$10.6436+0.0096i$	$11.5515+0.0824i$	
$11.9479+2.5133i$	$12.0365+0.5682i$	$12.5734+0.1726i$	
$12.5405+0.3457i$	$13.3287+0.0781i$	$14.2606+1.1892i$	
$14.7397+0.0950i$			

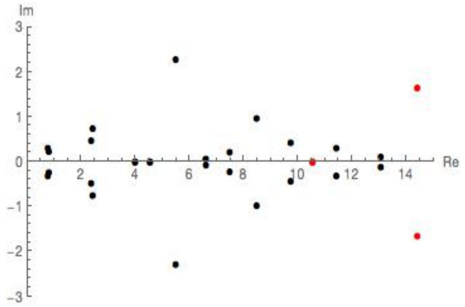
$J = \{2\}$			
$\delta = (-1.4, 0)$	$\delta = (-1, 0)$	$\delta = (-0.5, 0)$	$\delta = (-0.2, 0)$
$13.8989 + 0.6895i$	$14.2838 + 1.3295i$	$10.5659 + 0.0098i$ $14.4389 + 1.6448i$	$10.2868 + 0.0250i$ $10.2834 + 0.0320i$ $14.4878 + 1.6279i$
$\delta = (-0.1, 0)$	$\delta = (0, 0)$	$\delta = (0.1, 0)$	$\delta = (0.2, 0)$
$10.2204 + 0.0203i$ $10.2243 + 0.0251i$ $14.4936 + 1.6192i$	$10.2020 + 0.0202i$ $14.495 + 1.6162i$	$10.2181 + 0.0199i$ $10.2258 + 0.0247i$ $14.4943 + 1.6187i$	$10.2714 + 0.0222i$ $10.2902 + 0.0295i$ $14.4932 + 1.6244i$
$\delta = (0.5, 0)$	$\delta = (1, 0)$	$\delta = (1.4, 0)$	
$14.5090 + 1.6076i$	$14.0822 + 1.4900i$	$12.9339 + 2.3059i$	



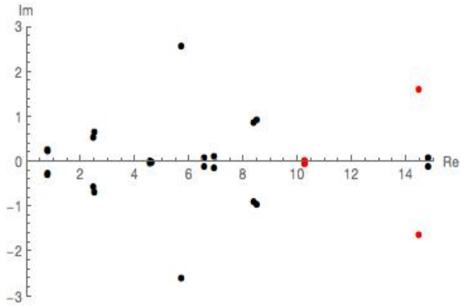
$\delta = (-1.4, 0)$



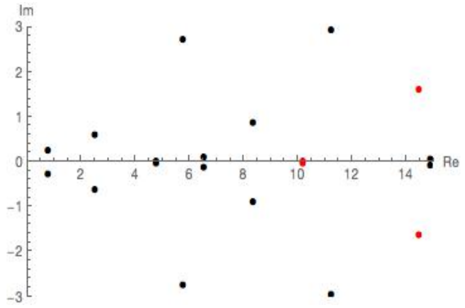
$\delta = (-1, 0)$



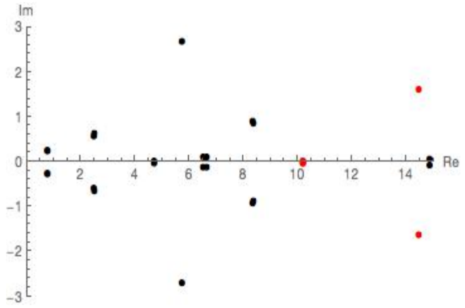
$\delta = (-0.5, 0)$



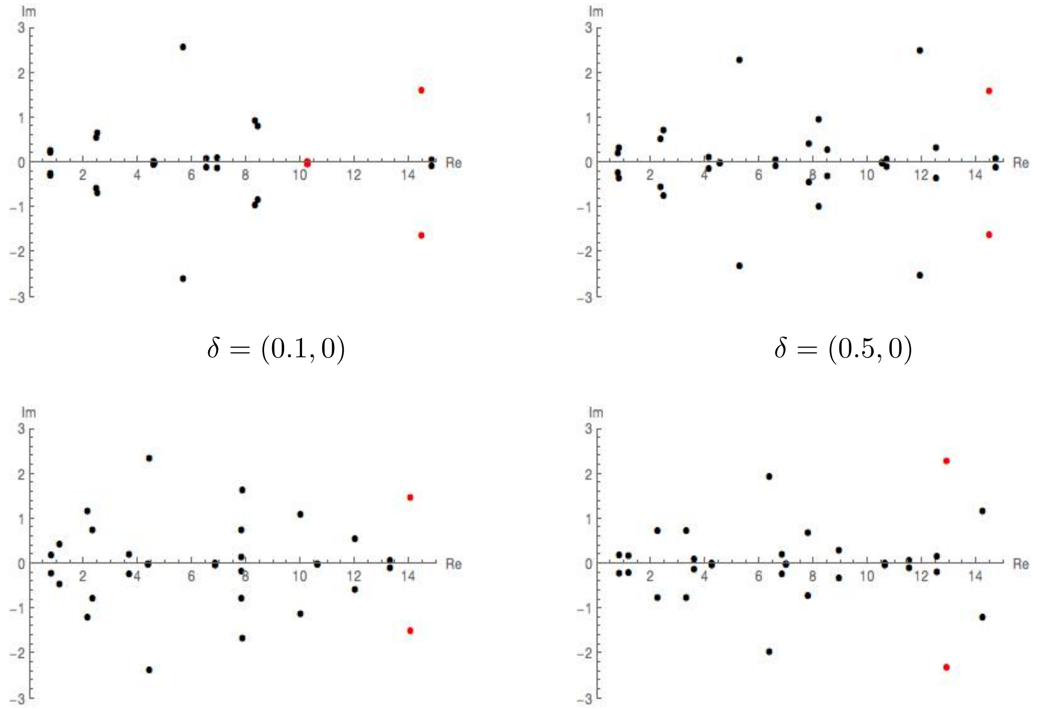
$\delta = (-0.2, 0)$



$\delta = (-0.1, 0)$



$\delta = (0, 0)$



$\delta = (0.1, 0)$ $\delta = (0.5, 0)$

$\delta = (1, 0)$ $\delta = (1.4, 0)$

Figure 5.30: Some plots showing the location of the resonances as the obstacle is moved now in the x direction in the three ended case. We see a similar phenomena; some resonances bifurcate once the obstacle ceases to be located in the centre.

5.6 Time delay and scattering length

As well as the calculation of resonances, there are other useful things we can obtain from the scattering matrix and its derivative; among these are the time delay and scattering length. The concept is outlined by Reed-Simon [1]. Müller and Strohmaier have also covered it in their paper [10], where they give results that relate the time delay to the geometry of the internal domain. In this section, we will apply this to a selection of our domains. We should note here that the λ in this context will be a real number less than μ_1 representing the energy of the system and not an element of the Z that is has been up until now. The notion of time delay hails from dynamic scattering theory, which we do not intend to cover in this thesis. The Appendix of Müller and Strohmaier's paper provides a good overview of the time delay in this setting [4]. For our purposes, we will take the (non standard) definition to be:

Definition 75. ***Time delay***

$$T(\lambda) = -2\sqrt{\lambda}S^{-1}(\lambda).S'(\lambda)$$

*when $\lambda = 0$, we define this to be the **scattering length**.*

Wigner and Eisenbud were the first to present the time delay in this manner for potential scattering and $T(\lambda)$ is often called the Eisenbud-Wigner time delay operator [45][46]. Müller and Strohmaier have, amongst other things, proved this formula for the case of manifolds with cylindrical ends and, in the case of a single end

$$T(0) = 2\frac{\mathrm{Vol}(X)}{\mathrm{Vol}(\Gamma)}.$$

We will pick a selection of our single ended domains from above and plot their time delay as λ approaches 0.

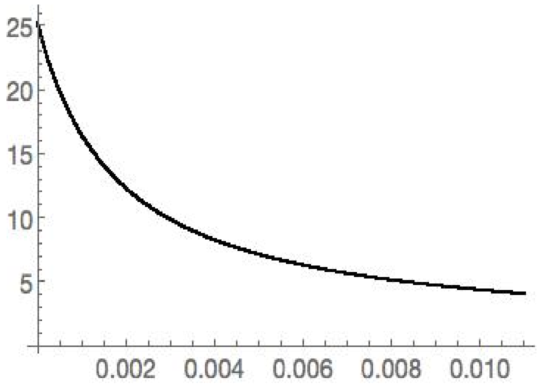


Figure 5.31: Circle of radius 2, with end width 1.5. $T(0)$ should theoretically be 25.1327.

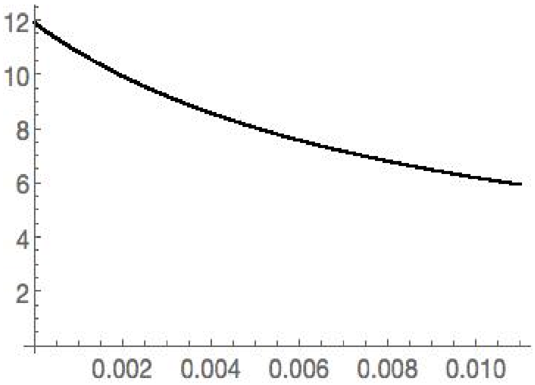


Figure 5.32: Circle of radius 2 with end width 1, obstacle radius 0.5. $T(0)$ should theoretically be 11.781.

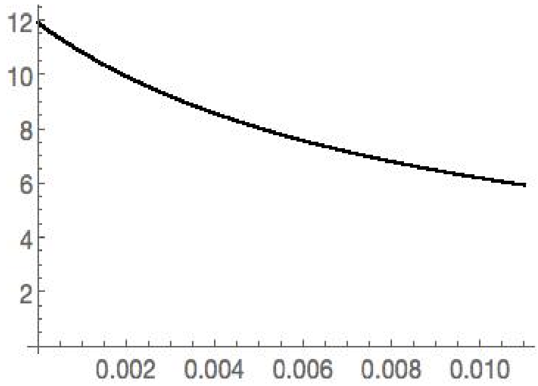


Figure 5.33: Circle of radius 2 with end width 1, obstacle radius 0.5 and translated by 0.5 in the y axis. $T(0)$ should theoretically be 11.781.

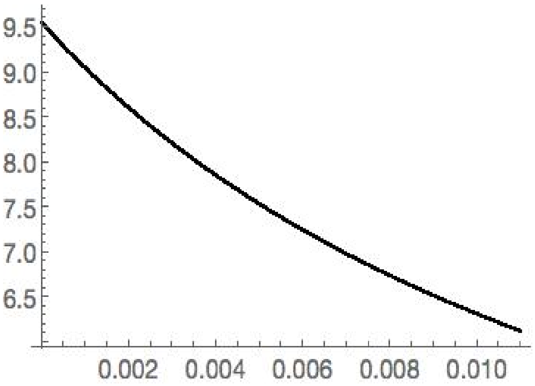


Figure 5.34: Circle of radius 2 with end width 1, obstacle radius 0.5. $T(0)$ should theoretically be 9.42478

Chapter 6

Documentation of code

To produce such results, we have used a collection of FreeFem++ [8], Mathematica and shell scripts, together with a coherent system of organising the data they produced, which taken together make up a program. This section will explain all aspects of this, in order to maximise the utility of this code for anyone who wishes to make use of it and/or add to it. We should state that, largely due to time constraints, we have only considered the Neumann waveguides, but this can be very easily adapted for Dirichlet waveguides. This section contains a detained description of the algorithm, its use and implementation. If the reader simply wishes to see an overview of the program, we have included a diagram/flowchart at the end of this section.

6.1 Documentation Freefem++ of code

6.1.1 File structure

As we are dealing with potentially hundreds of domains, we need some for of coherent system to organise them. We have put FreeFem scripts themselves in the same directory of FreeFem. Output data from our programs is stored in various nested subdirectories from where each individual domain has its own further, unique subdirectory containing all the data associated with that domain. The name of each such directory will contain data for that domain and, in addition to a possibly brief brief text description, the directory name will be made up of several numbers, separated by underscores, that uniquely determine the domain it represents. This is best shown by some examples:

```
/freefem/circleinrectangle/8_2_2_0.5_0/
/freefem/circle1ends/8_2_3_0_0_0/
/freefem/circle1ends/8_2_3_0.3_0_0/
/freefem/circle2ends/8_2_3_0_0_0_0_0/
/freefem/circle2ends/8_2_3_0_0_0.3_0_0/
/freefem/circle3ends/8_2_3_0_0_0_0_0/
/freefem/circle3ends/8_2_3_0_0_0_0.3_0_0/
```

The precise meaning of these numbers will be contained in the documentation of the piece of code that generated the directory. They exist to provide each domain with its own directory and provide information to a human reader; no other part of the program reads them.

6.1.2 Generating domains

In FreeFem each domain is constructed from piecewise curves. To create a domain from scratch is a time consuming, laborious process. To partially alleviate this, we have used a series of "templates": FreeFem scripts that create a particular type of domain. Changing a small parameter within the code of the template alters an aspect of the domain produced, the width of the ends for instance, without having to redraw the domain again and again from scratch. These templates create mesh files and boundary data and create the directory they are are saved in. The eigenvalue generating and direct Neumann to Dirichlet map generating scripts will read these files and store their output in the same directory. In addition to the mesh file being saved, we also save a JPEG file of a plot of the domain (some of these are displayed in the previous section) and a text file containing information about the boundary Γ.

For example, if we were to take the first example of

```
/freefem/circleinrectangle/8_2_2_0.5_0/
```

The script will create/has created the subdirectory

```
/8_2_2_0.5_0/
```

automatically but when the template was first used, the directory

```
/circleinrectangle/
```

needed to be both created manually and the permissions set to make it writable. For the use of existing templates, this is not an issue, but if the reader wishes to create his own, or modify existing templates, it is necessary to perform this step.

In the example we have used in this section, the script entitled `makecircleinrectangle` will create a rectangle with 8 mesh refinements, of length 2, width 2, and a circle of radius 0.5 cut out from the centre of the rectangle, with Γ consisting of the left and right sides of the rectangle. The name of these domain directories will vary depending on the type of domain and, again, the name of the directory will not be read by any further scripts, except in order to open the directory. The variables that alter the domain produced will be clearly documented with comments inside the code of each template. To create a very large number of domains, the entire code of a template can be wrapped in a loop, with the looping variable used to change the parameters.

Once the mesh generating template script has been run, the mesh containing directory will contain three files:

```
bdata.txt
mesh.msh
meshplot.jpg
```

The `mesh.msh` file is FreeFem's own file format and can be imported and exported. The text file contains information about the boundaries and the number of ends. Whilst this information is contained in the `.msh` file, extracting it is very labour intensive. The contents of the `bdata.txt` file for our example is:

```
2,2,0,2,2,0,0,0,2,2
```

The first number is always the number of ends. The subsequent numbers are the x and then the y coordinates of the first start point, the x and y coordinates of the first end point, the x and the y coordinates of the second start point, the x and the y values of the second end point. If there were more ends then this would continue in a similar manner. The final two values are the lengths of the two boundaries. Domains/mesh in FreeFem are made from the triangulations of boundaries, which are themselves made up piecewise of parameterised curves and lines. Each curve/line will be given a numerical label. Those making up $\partial X \setminus \Gamma$ will have a label of 0, those making up a connected component of Γ will be given a number which **must** correspond to its position in the `bdata.txt` file. For example The line from $(2, 0)$ to $(2, 2)$ will have the label 1.

6.1.3 Generating a directly computed Neumann to Dirichlet map

This script is run in tandem with the `readerautoevs` script, which we shall document in the next subsection. Its purpose is to directly compute a Neumann to Dirichlet map of specified spectral parameter and specified dimensions, so that we can make use of the technique from equation (3.30). To run this script we must have a `mesh.msh` file, a `bdata.txt` file and a directory for them as described above. This script needs three parameters in its standard input;

- the path to a directory containing only one `mesh.msh` file and only one `bdata.txt` file corresponding to the mesh.
- A value of λ. Throughout this work we have used the value 0.4, but there is no particular reason for this, it can be any value that is not a Neumann eigenvalue.
- The number of modes to compute $N(\lambda)$ up to: If you have a domain with k ends, the dimension of the resulting matrix will be k times this number.

Note 76. *When running the "*`readerautoevs` *script, which we shall go over in the next subsection, the number of modes given to it, for the same domain,* ***must be exactly the same as this*** *otherwise the Mathematica code that reads the output from both of these scripts will fail!*

The command to activate this script for our sample domain is shown below.

```
cd "<Path to FreeFem>/freefem" ; /usr/local/bin/FreeFem++
    "<Path to FreeFem>/freefem/pivotndm"
    <Path to FreeFem>/freefem/circleinrectangle/8_2_2_0.5_0/ 0.4 20
```

This script will create two text files and store them in the directory that was given to it as a parameter and which also contains the `mesh.msh` and `bdata.txt` files. These are as follows:

```
ppmcalcsize.txt
ppmndfcs.txt
```

The `ppmcalcsize.txt` file for this domain and parameters entered above will be:

```
20
0.4
2
```

This should be fairly self-explanatory; number of eigenvalues of each connected component of Γ, the spectral parameter for which we have directly compute a Neumann to Dirichlet map for, the number of ends of our domain (the latter is duplicated as it also appears in the `bdata.txt` file, but makes life easier later on, when constructing a Matrix from this in mathematica to have all of this information in a single file).

The `ppmndfcs.txt` file contains all of the entries that will make up our directly computer matrix. The entries are computed exactly as described in Definition 66, for our working example, there will be 1600 entries in this file; $(20 \times 2) \times (20 \times 2)$, where each 40 entries is a column of the Neumann to Dirichlet map.

6.1.4 Generating eigendata

- the path to a directory containing only one `mesh.msh` file and only one `bdata.txt` file corresponding to the the mesh
- A value of λ. Throughout this work we have used the value 0.4, but there is no particular reason for this, it can be any value that is not a Neumann eigenvalue.
- the basis elements of each connected component of Γ to compute $N(\lambda)$ up to. i.e. if you have a domain with 2 ends, the dimension of the resulting matrix will be 2 times this number.

```
cd "<Path to FreeFem>/freefem" ; /usr/local/bin/FreeFem++
    "<Path to FreeFem>/freefem/readerautoevs"
    <Path to FreeFem>/freefem/circleinrectangle/8_2_2_0.5_0/ 200 20
```

The command above would call the script `readerautoevs` which would compute eigendata for X for 200 eigenvalues and 20. This will eventually be used to construct equation 3.29 with the U_m truncated in this case at 200 values and the ϕ_j running from 1 to $20 \times 2 = 4$ values. We should repeat: the latter parameter **must** be the same as in the previous script otherwise the code will fail to work.

This script will export the output of its computations to three text files, stored in the same directory as the mesh.msh et al files. These are:

```
meval.txt
mevec.txt
mnums.txt
```

The meval.txt file stores the Neumann eigenvalues of the internal domain X; 200 for our working example. The mevec.txt file stores the values of

$$\langle \phi_k, U_m|_\Gamma \rangle_{L^2(\Gamma)} \quad \text{–see equations 3.29 and 3.30}$$

6.1.5 A shell script to automate the production of, Neumann to Dirichlet maps and eigendata

There have been times when we have wanted to apply the above two scripts to large number of domains, indeed most of the time! We have used a shell script to automate this process somewhat. This should be fairly self-explanatory:

```
#!/bin/bash

for entry in  "<Path to FreeFem>/freefem/circle3ends/"/*
do
if [ -e "$entry/meval.txt" ]
then
echo "${entry:28} Already has eigenvalues and eigenvectors"
else
cd "<Path to FreeFem>/freefem";
     /usr/local/bin/FreeFem++ "<Path to FreeFem>/freefem/readerautoevs"
     ${entry:28} 2000 20
fi
if [ -e "$entry/ppmndfcs.txt" ]
```

```
then
echo "${entry:28} Already has a directly computed nd map"
else
cd "<Path to FreeFem>/freefem";
    /usr/local/bin/FreeFem++ "<Path to FreeFem>/freefem/pivotndm"
    ${entry:28} 0.4 20
fi
done
```

Obviously the first line can be changed depending on the location of the mesh data to be processed and the numbers passed into the standard input for the `readerautoevs` and `pivotndm` can be changed to taste.

6.2 Documentation Mathematica code

In the previous section we have described the system we have created to organise our data and the FreeFem scripts used to generate it, we will now move on to describe the mathematica code we have used to process it, with the aim of calculating resonances.

6.2.1 Reading the eigendata

As was stated before, each domain is contained in its own directory, whose unique name describes it in terms of parameters (width of ends, mesh refinements etc). The parent directory of these is a subdirectory of the ”freefem” directory that describes a class of objects, in this case a circle with three ends, all generated by a single template script.

```
Domains =
  FileNames[ {"<Path to freefem>/freefem/gaunt/*",
    "<Path to freefem>/freefem/circle3ends/*"}];

PopupMenu[Dynamic[jj], Domains]
Dynamic[jj]
Dynamic[ImageTake[Import[jj, "meshplot.jpg"], {400, 800}]]
Print[Dynamic[
  First[Import[StringJoin[jj, "/mnums.txt"], "Table"][[
    2]]]], " Eigenvalues, ",
 Dynamic[First[
   Import[StringJoin[jj, "/mnums.txt"], "Table"][[3]]]], " Modes"]
```

This piece of code reads all data containing subdirectories of parent directories given in the form of a list at the beginning and generates a popup menu of all the available domains, dynamically displays a plot of their mesh the number of eigenvalues and modes available.

```
{CDIM,tempLAMBDA,ENDSc}= Flatten[Import[StringJoin[jj,
                   "/ppmcalcsize.txt"],"Table"]];

tempndm = Transpose[Partition[ToExpression[First[Import[StringJoin[jj,
                   "/ppmndfcs.txt"], "Table"]]], ENDSc*CDIM]];

H=Flatten[ToExpression[Import[StringJoin[jj, "/mnums.txt"], "Table"]]];
```

```
{leneval,lenfcs} = {H[[2]],H[[3]]};
Widths = Table[First[H[[i]]], {i, 4, ENDSc + 3}];
Clear[H];

NEVEC  = Partition[ToExpression[Flatten[Import[StringJoin[jj, "/mevec.txt"],
             "Table"]]],lenfcs*ENDSc];

EVAL =ToExpression[Flatten[Import[StringJoin[jj, "/meval.txt"], "Table"]]];
```

Here, the reader will be able to see that `tempndm` is a directly computed Neumann to Dirichlet map, `EVAL` are the Neumann eigenvalues for the internal domain and `NEVEC` are the values for $\langle \phi_k, U_m|_\Gamma \rangle_{L^2(\Gamma)}$, which was mentioned in the previous section and its role in the indirectly computed Neumann to Dirichlet map is explained in equations 3.29 and 3.30.

6.2.2 Calculating scattering matrices and their derivatives

Given that a domain has been selected, the `tempndm`, `EVAL` and `EVEC` tables obtained from it's data directory, this next, piece of code is what produces the scattering matrices and their derivatives.

Here we can see that `l` is the spectral parameter, `e` is the number of eigenvalues we wish to use to calculate the scattering matrix and derivatives, `m` the number of modes, `der` the number of derivatives to calculate up to and `mo` is used to describe the set J.

`mo` will be how we pass the set J to our program and will take the form of $\#ENDS$ nested tables, each table will decide whether each $\sqrt{\lambda - \mu_j}$ for each end is the positive or negative branch of the square root; if negative then j appears in the table.

Note 77. *We should caution the reader here that this input has the potential to accept ambiguous information and it is the responsibility of the user to avoid this. If a transverse Neumann Eigenvalue has multiplicity greater than 1, then the "mo" should take this into account. For example, a three ended domain, where all ends have equal width then*

$$\texttt{mo=\{\{1,2\},\{1,2\},\{2\}\}} \tag{6.1}$$

is wrong. It is telling the program that $\sqrt{\lambda - \mu_1}$ will be on the negative branch when it appears as a transverse eigenvalue of the first and second ends, but not the third.

Obviously if a domain's widths are all the same, all tables in "mo" must be the same. Examples of values for "mo" for three ended domains whose ends have equal width are:

```
{{1},{1},{1}}
{{2},{2},{2}}
{{1,2},{1,2},{1,2}}.
```

As we can see the first entry corresponds to $J = \{1\}$, the second to $J = \{2\}$ and the third to $J = \{1, 2\}$.

```
TSM[l_, e_, m_, der_, mo_] :=
 Block[{picker =
    Flatten[Table[j + lenfcs (i - 1), {i, 1, ENDSc}, {j, 1, m}], 1],
   tempndm = tempndm[[picker, picker]],
   NEVEC = Drop[NEVEC, {e + 1, leneval}, None],
   EVAL = EVAL[[1 ;; e]],
   NDMAP,
   rEXNDM,
   rDaut,
   reEXT,
   rxEXT,
   rDNATNDMAP,
   rDXXK,
   rDiagdif,
   U, V, W,
   TA, OP, dim},


  $HistoryLength = 1;


  NEVEC = NEVEC[[All, picker]];

  (*Indirectly computed Neumann to Dirichlet map with convergence
   acceleration "trick"*)

  NDMAP = tempndm -
    ConjugateTranspose[
      NEVEC].DiagonalMatrix[(tempLAMBDA- l)/(l*tempLAMBDA - l*EVAL -
         tempLAMBDA*EVAL + EVAL^2)].NEVEC;

  (*Neumann to Dirichlet map for the system up to the n-th derivative*)

  rDNATNDMAP[n_] :=
   Which[n < 0, Return[ConstantArray[0, {m*ENDSc, m*ENDSc}]], n > 0,
    Return[-ConjugateTranspose[
        NEVEC].DiagonalMatrix[(-1)^n/(l - EVAL)^(n + 1)].NEVEC], True,
     Return[NDMAP]];

  (*Computes square root of x minus the j-
  the Neumann eigenvalue on the w-th end*)

  rEXNDM[x_, j_, w_, trig_] :=
   If[Im[l] < 0, -I*Sqrt[x - (j*Pi/Widths[[w]])^2],
    I*Sqrt[x - (j*Pi/Widths[[w]])^2]];

  (*this is the Neumann to dirichlet map on the external domain
differentiated n times*)

  rDaut[n_] :=
   If[n >= 0,
    DiagonalMatrix[
     Flatten[Table[
       D[rEXNDM[x, jj, ii, 1], {x, n}] /. {x -> l}, {ii, 1,
        ENDSc}, {jj, 0, m - 1}]]],
    ConstantArray[0, {m*ENDSc, m*ENDSc}]];


  (*Puts all of the der*der block-
  matrices together to construct the external Neumann to Dirichlet
map for the system*)

  reEXT := Inverse[
    Join[ Flatten[
      Table[Transpose[
        Join[Flatten[
          Table[Transpose[rDaut[i - j]*Binomial[i - 1, j - 1]],
          {j, 1,der}], 1]]] , {i, 1, der}], 1]]];

  (*Puts all of the der*der block-
  matrices together to construct the internal Neumann to Dirichlet
map for the system*)

  rxEXT :=
   Join[ Flatten[
     Table[Transpose[
       Join[Flatten[Table[Transpose[rDNATNDMAP[i - j]], {j, 1, der}],
         1]]] , {i, 1, der}], 1]];

  (*combines the internal and external Neumann to Dirichlet maps*)

  rDXXK = rxEXT - reEXT;

  (*this applies the appropriate projection to the combined Neumann
to Dirichlet maps*)

  rDXXK[[Flatten[
     Table[endz*m + mo[[endz + 1]], {endz, 0, ENDSc - 1}]], All]] =
   0;

  (*this is multiplied to the previously calculated derivatives of
scattering matrices to help compute the derivative*)

  rDiagdif[n_] :=
   DiagonalMatrix[
    Flatten[Table[
      D[rEXNDM[x, jj, ii, 1], {x, n}] /. {x -> 1,
        jj -> (mo[[ii]] - 1)}, {ii, 1, ENDSc}]]];


  {U, V, W} = SingularValueDecomposition[rDXXK, Tolerance -> 0];
  dim = Length[Flatten[mo]];


  TA = Table[
    Transpose[
      Partition[
       Flatten[Table[
         W[[All, j]][[m*ENDSc*(cdrv - 1) + i*m + mo[[i + 1]]]], {j,
          m*ENDSc*der - dim + 1, m*ENDSc*der}, {i, 0, ENDSc - 1}]],
       dim]].Inverse[
      Transpose[
       Partition[
        Flatten[Table[(rxEXT.W[[All, j]])[[
           m*ENDSc*(cdrv - 1) + i*m + mo[[i + 1]]]], {j,
           m*ENDSc*der - dim + 1, m*ENDSc*der}, {i, 0, ENDSc - 1}]],
        dim]]], {cdrv, 1, der}];

  OP = {-Inverse[(TA[[1]] - rDiagdif[0])].(rDiagdif[0] + TA[[1]])};
  Do[OP =
    Join[OP, {Inverse[(TA[[cdrv]] - rDiagdif[0])].(Sum[
          Binomial[cdrv - 1, kk]*rDiagdif[kk].OP[[cdrv - kk]], {kk, 1,
            cdrv - 1}] - rDiagdif[(cdrv - 1)])}], {cdrv, 2, der}
   ];
  OP]
```

Exactly how scattering matrices and their derivatives can be/are computed was discussed in detail in section 4.1; the code featured above is simply an implementation of that, so it is unnecessary to go into further detail about it here. The output of this function is a table of matrices; each of appropriate rank. The first entry, the scattering matrix, the second, its derivative, the third its second derivative etc. etc.

6.2.3 Calculating resonances

We will now give on overview of some functions which, together, allow us to calculate resonances.

```
FUNC[a_,modecontrol_] :=
  With[{temp = TSM[a, 200,10, 2,modecontrol}]},
   Tr[(Inverse[temp[[1]]]).temp[[2]]]];
```

`FUNC` Applies the Jacobi formula to obtain the derivative of the determinant of the scattering matrix then divides by the determinant. This will be our integrand (See Proposition 74).

```
SMSONATOR[a_, b_, density_] :=  Block[{simpnum =
Max[IntegerPart[20*Abs[b - a]*density], 20], h},
   h = (b - a)/simpnum;
   $HistoryLength = 1;
   (h/3)*Sum[
     FUNC[a + (2 in - 2) h] + 4*FUNC[a + (2 in - 1) h] +
      FUNC[a + (2 in) h], {in, 1, (simpnum/2)}]
   ];
```

`SMSONATOR` performs a numerical integration from `a` to `b` using the Composite Simpson's rule with the density parameter multiplying 20 times the distance from a to b giving the number of subintervals. If this number is insufficiently large, then twenty subintervals will be used.

```
Clear[fastgrid]
fastgrid[rmin_, rmax_, rni_, immin_, immax_, ini_, density_] :=
  Block[{rh = (rmax - rmin)/rni,
    ih = (immax - immin)/ini,
    horizints,
    vertints,
    values},
   $HistoryLength = 1;
   $Messages = {};
   values =
    Table[rmin + immin I + rr*rh + ii*ih I, {ii, 0, ini}, {rr, 0,
      rni}];
   horizints =
    Table[SMSONATOR[values[[cc, rr]], values[[cc, rr + 1]],
      density], {cc, 1, ini + 1}, {rr, 1, rni}];
   vertints =
    Table[SMSONATOR[values[[cc, rr]], values[[cc + 1, rr]],
      density], {cc, 1, ini}, {rr, 1, rni + 1}];
   Flatten[
    Table[{-vertints[[ii, jj]] + horizints[[ii, jj]] +
        vertints[[ii, jj + 1]] -
        horizints[[ii + 1, jj]], {Re[values[[ii, jj]]],
        Re[values[[ii + 1, jj + 1]]], 2, Im[values[[ii, jj]]],
        Im[values[[ii + 1, jj + 1]]], 2, density}}, {jj, 1, rni}, {ii,
      1, ini}], 1]];
Share[fastgrid]
```

`fastgrid` performs numerical integration over several rectangular contours using Simpson's rule. The values `rmin` and `rmax` are the minimum and maximum real values of the search area, similarly `immin` and `immax` are the minimum and maximum values of the imaginary component of the total area. `rni` and `ini` are the number of intervals the real and imaginary intervals should be divided into, meaning that a total of `rni`×`ini` contour integrations will be performed. The program will output as many tables of the form:

$$\texttt{\{<value of the integration>,\{ rmin,rmax,2,immin,immax,2,density\}\}} \quad (6.2)$$

nested in an array themselves, where the `rmin`, `rmax`, `immin`, `immax` here are the values for that particular sub-contour, not the ones inputted. The `density` value here will be the same as inputted. Each such table is designed so that it can be fed back into the `fastgrid` module, without modification, if desired, where it will be further subdivided and contour integrations performed as before. It is intended that this code be applied iteratively to various selected elements of its output to give an array of exponentially shrinking contours. The user can define a search area that is not necessarily square on the first iteration and use the `rni` and `ini` parameters to ensure that the sub-contours will be, after which, if an element of the output is fed back into the program, these values will both be 2.

Since performing numerical integration over the derivative of the determinant of the scattering matrix is computationally costly, this code re-cycles the integrals of intervals, where two sub-contours share a line segment, providing greater efficiency. A further improvement on this is the following:

```
fastergrid[rmin_, rmax_, rni_, immin_, immax_, ini_, density_,
  iterations_, vintsl_ , vintsr_, hintst_, hintsb_] :=
 Module[{rh, imh, values, horizints, vertints, rsublen, imsublen},
  $HistoryLength = 1;
  Clear[vertints, horizints];
  rh = If[NumberQ[hintst] == True,
    N[(rmax - rmin)/(rni*2^(iterations - 1))],
    N[(rmax - rmin)/Length[hintst]]];
  imh = If[NumberQ[vintsl] == True,
    N[(immax - immin)/(ini*2^(iterations - 1))],
    N[(immax - immin)/Length[vintsl]]];
  values =
   Table[rsp + isp I, {isp, immin, immax, imh}, {rsp, rmin, rmax, rh}];
  rsublen = (Length[values[[1, All]]] - 1)/rni;
  imsublen = (Length[values[[All, 1]]] - 1)/ini;



  horizints =
   If[NumberQ[hintst] == True,
    Transpose[
     Table[Table[
       SMSONATOR[values[[1 + cc*imsublen, rsi + rsublen*ri]],
        values[[1 + cc*imsublen, rsi + rsublen*ri + 1]],
        density], {rsi, 1, rsublen}], {ri, 0, rni - 1}, {cc, 0, ini}]],
    Join[{Partition[hintst, rsublen]}, {Table[
       Table[SMSONATOR[values[[1 + imsublen, rsi + rsublen*ri]],
         values[[1 + imsublen, rsi + rsublen*ri + 1]], density], {rsi,
          1, rsublen}], {ri, 0, rni - 1}]}, {Partition[hintsb,
       rsublen]}]];
  vertints =
   If[NumberQ[vintsr] == True,
    Table[Table[
      SMSONATOR[values[[csi + ci*imsublen, 1 + rr*rsublen]],
       values[[csi + ci*imsublen + 1, 1 + rsublen*rr]],
       density], {csi, 1, imsublen}], {ci, 0, ini - 1}, {rr, 0, rni}],
     Transpose[
     Insert[Join[{Partition[vintsl, imsublen]}, {Partition[vintsr,
         imsublen]}],
      Table[Table[
        SMSONATOR[values[[csi + ci*imsublen, 1 + rsublen]],
         values[[csi + ci*imsublen + 1, 1 + rsublen]], density], {csi,
          1, imsublen}], {ci, 0, ini - 1}], 2]]];
  Flatten[
   Table[{-Total[vertints[[ii, jj]]] + Total[horizints[[ii, jj]]] +
      Total[vertints[[ii, jj + 1]]] -
      Total[horizints[[ii + 1, jj]]], {Re[
       values[[1 + (ii - 1) imsublen, 1 + (jj - 1) rsublen]]],
      Re[values[[1 + ii imsublen, 1 + jj rsublen]]], 2,
      Im[values[[1 + (ii - 1) imsublen, 1 + (jj - 1) rsublen]]],
      Im[values[[1 + ii imsublen, 1 + jj rsublen]]], 2, iterations,
      density, vertints[[ii, jj]], vertints[[ii, jj + 1]],
      horizints[[ii, jj]], horizints[[ii + 1, jj]]}}, {jj, 1,
     rni}, {ii, 1, ini}], 1]]
```

This code peforms numerical integration over the specified area, in the same way as `fastgrid` but its output will be a table where each element will have the the form

```
{<value of the integration>,{ rmin,rmax,2,immin,immax,2,density,
                             {<lefthandside integrals>},
                             {<righthandsideintegrals>},
                             {<bottom inegrals>},{<top integrals>}}}
```

(6.3)

Each left, one of the right and top bottom integrals will be a table of values such that, when summed, will be the numerical integral of the respective parts of that contour given by the preceding `rmin`, `rmax` ,`immin` `immax` values with the desired number of points as specified by the `density` parameter in the beginning. These four tables will then be subsequently subdivided on subsequent iterations to give the top bottom left and right integrals of the contours whose lengths shrink by a factor of two on each iteration. The number of entries will be determined by the `iterations` parameter in the beginning. These can then be fed back into the module on subsequent iteration eliminating the need for numerically integrating eight line segments on each contour each time. This gives a huge advantage over the previous procedure; unnecessary, repeated numerical elimination of the same line segment is completely eliminated!

```
fastresonancefinder[rmin_, rmax_, rni_, immin_, immax_, ini_,
  density_, iterations_] :=
 Module[{fastintsiloc, intsiloc, fastrfoutput, rfoutput,
   fastiterations},
  $HistoryLength = 1;
  fastiterations =
   IntegerPart[N[Log[2, density*20*(rmax - rmin)/rni] - 1]];
  Print[fastiterations, " fast iterations"];
  fastintsiloc = {{2 I, {rmin, rmax, rni, immin, immax, ini, density,
      fastiterations, 0, 0, 0, 0}}};
  fastrfoutput = {{2 I, {rmin, rmax, rni, immin, immax, ini, density,
      fastiterations, 0, 0, 0, 0}}};
  Do[fastintsiloc = Flatten[fastergrid @@@ fastintsiloc[[All, 2]], 1];
    fastintsiloc = Select[fastintsiloc, Abs[Im[#[[1]]]] > 1 &];
   Print[fastintsiloc];
   If[NumberQ[fastintsiloc[[1, 1]]] == True,
    fastrfoutput = fastintsiloc, Break[]];
   Print["fast iteration complete"], {it, 0, fastiterations - 1}];

  intsiloc =
   Flatten[fastgrid @@@
     Drop[Drop[Partition[Flatten[fastrfoutput], 13], None, {9, 13}],
      None, {1}], 1];
  intsiloc = Select[intsiloc, Abs[Im[#[[1]]]] > 0.5 &];
  rfoutput = intsiloc;
  Print[intsiloc];
  Do[intsiloc = Flatten[fastgrid @@@ intsiloc[[All, 2]], 1];
   intsiloc = Select[intsiloc, Abs[Im[#[[1]]]] > 0.5 &];
   If[NumberQ[intsiloc[[1, 1]]] == True, rfoutput = intsiloc, Break[]];
   Print["normal iteration complete"], {it, 0,
    iterations - fastiterations - 2}];
  rfoutput]
```

This code takes a search area given by `rmin`- the minimum real value, `rmax`, the maximum real value `immin` and `immax` the minimum and maximum imaginary values respectively. The `rni` and `ini` parameters give the number of subdivisions of the search area on the first iteration; for subsequent iterations this will be 2. The `rni` and `ini` can be used to ensure that on the first, (and consequently all the others) iteration, the sub-contours will be square They can also be used to cut computation time if some information is known beforehand about the location of the resonances. The density parameter has been discussed before will simply be passed to the `FUNC` module that makes use of it when performing the integration. The `iterations` parameter is self-explanatory; the number of times that contours containing resonances should be sub-divided and integrated over. Each iteration produces an array of tables in the form of (6.3); those with a contour integral whose imaginary part is less than a threshold (which we have set to 1 here) are discarded, the rest will be fed back into the `fastergridgrid` routine. Whilst the numerical contour integral around resonances should be an integer miultiple of 2π, we would have to take more interpolation points than is practical to get close to this number; experience has taught us that the imaginary part of the numerical integral around contours containing no resonance is very close to zero, hence we discard them if it is less than 1. The user can control these parameters and make a tradeoff between speed and accuracy. As the contours become very small, the number of interpolation points per unit length will have to increase to ensure that there are any points for numerical integration at all, this is built into the `SMSONATOR` function. Because the `fastergrid` module works backwards, computing the numerical integral for a line segment, by computing the numerical integral for the smallest possible line segments that would result from successive subdivision of the original line segment, storing them for later use, then adding them up, this would cause the number of interpolation points used for the integration of the original line segment to be larger than desired if too many iterations/subdivisions are requested. To avoid this, after a certain number of iterations/subdivisions, the normal `fastgrid` is used instead, and each iteration will then produce an array of tables in the form of (6.2). At this point in time, the contours will be so small that numerically integrating around them will be acceptably fast anyway.

```
NRM[a_, mcontrol_, iterations_, repeat_,modes_evals_] :=
  Module[{temp},
  $HistoryLength = 1;
  temp = ConstantArray[0, 10];
  temp[[1]] = a;
  For[nn = 2, nn <= iterations, nn++,
   temp[[nn]] =
    temp[[nn - 1]] -
     1/Tr[(Inverse[
          TSM[temp[[nn - 1]], evals, modes, 1, mcontrol][[1]]]).(TSM[
           temp[[nn - 1]], evals, modes, 2, mcontrol][[2]])];
   If[Abs[Det[TSM[temp[[nn]], 200, 20, 1, mcontrol][[1]]]] == 0,
    Break[]]
   ];
  {temp[[nn - 1]],
   Abs[Det[TSM[temp[[nn - 1]], evals, modes, 1, mcontrol][[1]]]]}]
```

As we can see, this is an implementation of Newton's method. `a` is the start point and `mcontrol` is the sheet of our Riemann surface. `iterations` controls how many iterations of Newton's method we use. As usual; `modes` and `evlas` control how many modes of Γ and eigenvalues of X we use to make the Neumann to Dirichlet map.

```
Clear[NRMSQ]
NRMSQ[rmin_, rmax_, immin_, immax_, evals_, modes_, attempts_,
  iterations_, mcontrol_] := Module[{temp, output},
  $HistoryLength = 1;
  temp = ConstantArray[((rmax + rmin)/2 ) + ((immax + immin)/2) I,
    attempts];
  output = ConstantArray[Null, attempts];
  Do[
   Catch[Do[
      If[Abs[Det[TSM[temp[[aa]], evals, modes, 1, mcontrol][[1]]]] ==
        0, output[[aa]] = temp[[aa]]; Throw[temp[[aa]]]];
      temp[[aa]] =
       temp[[aa]] -
        1/(Tr[(Inverse[
              TSM[temp[[aa]], evals, modes, 1, mcontrol][[1]]]).(TSM[
               temp[[aa]], evals, modes, 2, mcontrol][[2]])] -
           Sum[1/(temp[[aa]] - temp[[jj]]), {jj, 1, aa - 1}]);
      If[Re[temp[[aa]]] < rmin || Re[temp[[aa]]] > rmax ||
        Im[temp[[aa]]] < immin || Im[temp[[aa]]] > immax ||
        NumberQ[temp[[aa]]] == False, output[[aa]] = Null;
       temp[[aa]] = Break[]];
      output[[aa]] = temp[[aa]];
      , {nn, 2, iterations}]];
   , {aa, 1, attempts}];
  DeleteCases[output, Null]]
```

The `resonancefinder` procedure has given us a contour; this may contain more than one resonance. We can apply Newton's method repeatedly in this case. After finding the exact value of a resonance, λ_0 say, the determinant of the scattering matrix (or indeed any function we wish to find the roots of) can be multiplied by $\frac{1}{\lambda-\lambda_0}$ and Newton's method applied to the product [47]. We may repeat this procedure as many times as we wish. This can be used to determine the order of the resonances as, Newton's method will converge to the same number after n repetitions of this procedure for a resonance of order n. The `NRMSQ` procedure is similar to the `NRM` procedure, except that it takes as an input the minimum and maximum real and imaginary values of a search area: `rmin rmax immin immax` and starts Newton's method in the midpoint between these real and imaginary maxima and minima. `atemts` controls how many times to repeat the above procedure and `iterations` controls the number of iterations of Newton's method each time. If during iterations the value strays beyond the minima and maxima real and imaginary values the code will stop. The output will be an array of length less or equal to the `attempts` parameter or even an empty array if no zeros/resonances are found within the search area.

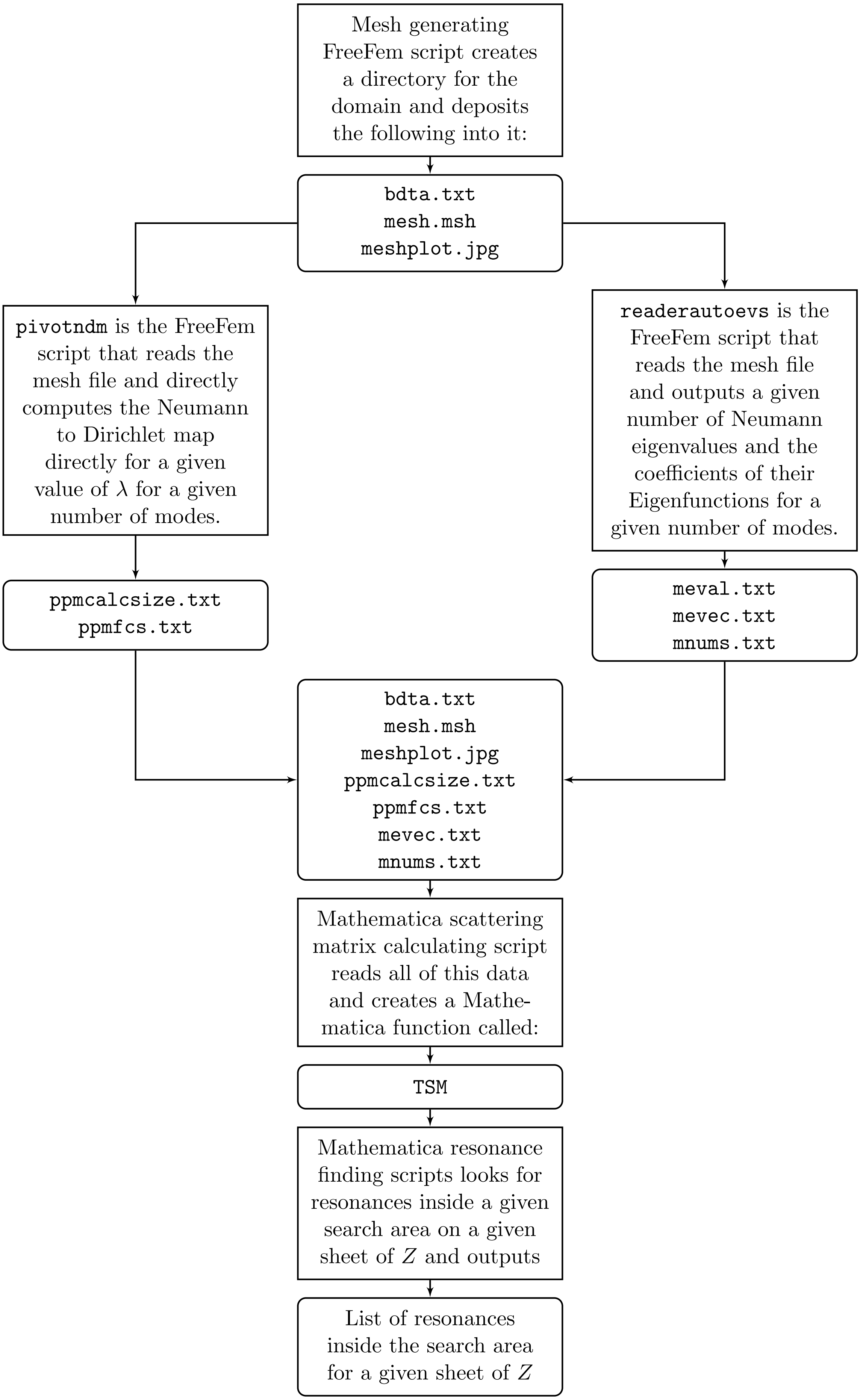
6.2.4 A sample of the implementation of this code for a search area

We believe it would be beneficial to the reader to give an example of these procedures set up to find resonances of a given search area.

```
BatchDomains={<an array of paths to domain directories
containing Neumann to Dirichlet
  data in string form>};
modetable={<An array of mode information>};

Do[
 modecontrol = modetable[[ii]];
 FUNC[a_] :=
  With[{temp = TSM[a, 600, 20, 2, modecontrol]},
   Tr[(Inverse[temp[[1]]]).temp[[2]]]];
   Do[
Print[HQBatchDomains[[nnnn]]];
Print[modecontrol];

{CDIM,tempLAMBDA,ENDSc}=
Flatten[Import[StringJoin[BatchDomains[[nnnn]],
                   "/ppmcalcsize.txt"],"Table"]];

tempndm = Transpose[Partition[ToExpression[First[Import[StringJoin[
BatchDomains[[nnnn]],    "/ppmndfcs.txt"], "Table"]]], ENDSc*CDIM]];

H=Flatten[ToExpression[Import[StringJoin[BatchDomains[[nnnn]],
               "/mnums.txt"], "Table"]]];
          {leneval,lenfcs} = {H[[2]],H[[3]]};

Widths = Table[First[H[[i]]], {i, 4, ENDSc + 3}];
Clear[H];

NEVEC  = Partition[ToExpression[Flatten[Import[StringJoin[BatchDomains[[nnnn]],
 "/mevec.txt"],
               "Table"]]],lenfcs*ENDSc];

EVAL =ToExpression[Flatten[Import[StringJoin[BatchDomains[[nnnn]],
       "/meval.txt"], "Table"]]];


   finderinput =
   Parallelize[
     fastresonancefinder[0.1, 15, 15, 0.0000001, 3, 3, 8, 10],
     Method -> "FinestGrained"][[All, 2]];
  finderinput = Drop[finderinput, None, {3}];
  finderinput[[All, 5]] = 1000;
  finderinput[[All, 6]] = 20;
  finderinput =
   Transpose[
    Join[Transpose[finderinput],
     Transpose[
      ConstantArray[{4, 25, modecontrol},
       Length[finderinput[[All, 1]]]]]]];
  output = Flatten[DeleteCases[NRMSQ @@@ finderinput, {}]];
  output = Join[output, Conjugate[output]];
  output = DeleteDuplicates[output];
  Export[StringJoin[HQBatchDomains[[nnnn]], "/resdata_",
    ToString[modetable[[ii]]]], output, "Table"];
  Print[output];
] ,{nnnn,1,Length[BatchDomains]} ], {ii, 1, Length[modetable]}]
```

This takes an array of paths to various user inputted domains, an array of mode control tables (we have described how these work at the start of 6.1) and calculates the resonances for them in a given search area ($[0.1 - 0.7] \times [0.0000001i, 3i]$ in this example), before saving them, in table format, in the same directory as the domain data. The output of the `resonancefinder` module, is changed slightly so that it can be inputted into the `NRMSQ` module to obtain the actual location of the resonances found. Experience has taught us that if a multi-core processor, or remote Mathematica kernels are available, it is better to parallelise at this point and not before. The output will be a series of tables of resonances within the defined search area for the various domains and sheets of Z requested. Within each table of resonances, each resonance will appear as often as its multiplicity.

6.2.5 A visual summary of all of this



Mesh generating FreeFem script creates a directory for the domain and deposits the following into it:

`bdta.txt`
`mesh.msh`
`meshplot.jpg`

`pivotndm` is the FreeFem script that reads the mesh file and directly computes the Neumann to Dirichlet map directly for a given value of λ for a given number of modes.

`ppmcalcsize.txt`
`ppmfcs.txt`

`readerautoevs` is the FreeFem script that reads the mesh file and outputs a given number of Neumann eigenvalues and the coefficients of their Eigenfunctions for a given number of modes.

`meval.txt`
`mevec.txt`
`mnums.txt`

`bdta.txt`
`mesh.msh`
`meshplot.jpg`
`ppmcalcsize.txt`
`ppmfcs.txt`
`mevec.txt`
`mnums.txt`

Mathematica scattering matrix calculating script reads all of this data and creates a Mathematica function called:

`TSM`

Mathematica resonance finding scripts looks for resonances inside a given search area on a given sheet of Z and outputs

List of resonances inside the search area for a given sheet of Z

6.3 3d Domains

Everything thus far has been implemented for domains in two dimensions, but this can extend to three dimensions with only minor modification to the file format and some knowledge of creating 3d objects in FreeFem with tetgen [48], or some other add-on to FreeFem++ which allows for the creation of 3-D mesh. Although no results have been shown here, this program has been extended to three dimensions with full functionality. There are some slight differences, which we shall document here.

6.3.1 Cuboidal ends

Once a mesh generating template script has been run, the mesh containing directory will contain two files. Unfortunately FreeFem's plot feature is unable to save 3d plots.

```
bdata.txt
mash.msh
```

Within the `bdata.txt` file the first number is still the number of ends. The subsequent numbers are the x, y and z coordinates of the first start point, the x and y and z components of the first width vector and the x and the y and z components of the first height vector. Subsequent ends follow a similar pattern. After this comes the width of the first end, then the width of subsequent ends, then the same for height. This will be stored in exactly the same way as for 2d domains. When we wish to generate a directly computed Neumann to dirichlet map and eigendata, we have created the scripts `pivotndm3drect` and `readerautoevs3drect`, which work exactly the same as for 2d domains. The modes on the boundary will be ordered as in Example 68. There is also Mathematica code that works in the same way for such domains to obtain scattering matrices and their derivatives.

6.3.2 Cylindrical, circular ends

A mesh generating template script will create the `bdata.txt` and `mesh.msh` files as above. The structure of the of the `bdata.txt` file is as follows: The first entry will be the number of ends, the next three entries will be the x, y and z coordinates of the centre of the first end, the next three entries will be the x, y and z components of a unit normal vector from the first end. This pattern is repeated for the number of ends in the domain. The entries after will be the radii of the first then subsequent ends.

In the FreeFem directory there will contain a comma separated list containing Neumann Bessel function zeros which we have labelled `neumbes0s.txt`. These are used in preference to Mathematica's internal library of such values. The first entry will be the number of Bessel functions we have provided zeros for and the second will be the Number of zeros for each Bessel function. The rest of the table will be the zeros; care must be taken to make sure that this table is large enough to provide sufficient data for the number of modes required. As for the cuboidal case, the modes on the boundary will be ordered as in Example 69.

The scripts `pivotndm3dcyl` and `readerautoevs3dcyl` will process the data. Separate mathematica code will process this to calculate the scattering matrices..

6.3.3 Cylindrical ends with arbitrary cross section

No code currently exists for this although there is no practical impediment to it being created. We would need to compute an orthonormal basis for each cross section beforehand then proceed as before.

Chapter 7

Concluding remarks

7.1 Future directions

The program is currently limited in that it is restricted to Neumann or Acoustic waveguides in Euclidian space: extending these techniques and computer programs to those with Dirichlet or mixed boundary conditions will require some more work, as will allowing for non-Euclidian metrics on either the internal domain or the ends. There is also the possibility to look at potentials supported on the internal domain and ends with arbitrary cross section. A natural application of this program now, will be the study of domain optimisation questions involving resonances.

7.2 Acknowledgements

Besides friends and family, I wish to thank my supervisor Dr Alexander Strohmaier for all the inspiration, advice and support during the time of writing and Dr Keith Watling for going out of his way to supply me with computers, without which the calculations would be a fraction of what they are.

Bibliography

[1] M. Reed and B. Simon, *Methods of modern mathematical physics. III.* Academic Press [Harcourt Brace Jovanovich, Publishers], New York-London, 1979. Scattering theory.

[2] T. Christiansen, "Some upper bounds on the number of resonances for manifolds with infinite cylindrical ends," *Annales Henri Poincar*, vol. 3, no. 5, pp. 895–920, 2002.

[3] M. Levitin and A. Strohmaier, "Computations of eigenvalues and resonances on perturbed hyperbolic surfaces with cusps,"

[4] W. Müller and A. Strohmaier, "Scattering at low energies on manifolds with cylindrical ends and stable systoles," *Geometric and Functional Analysis*, vol. 20, no. 3, pp. 741–778, 2010.

[5] R. Melrose, *Geometric Scattering Theory.* Stanford Lectures: Distinguished Visiting Lecturers in Mathematics, Cambridge University Press, 1995.

[6] D. Yafaev, *Mathematical Scattering Theory: General Theory.* Mathematical scattering theory: general theory, American Mathematical Soc., 1992.

[7] M. Levitin and M. Marletta, "A simple method of calculating eigenvalues and resonances in domains with infinite regular ends," *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, vol. 138, pp. 1043–1065, 10 2008.

[8] F. Hecht, "New development in freefem++," *J. Numer. Math.*, vol. 20, no. 3-4, pp. 251–265, 2012.

[9] A. Aslanyan, L. Parnovski, and D. Vassiliev, "Complex resonances in acoustic waveguides," *The Quarterly Journal of Mechanics and Applied Mathematics*, vol. 53, no. 3, pp. 429–447, 2000.

[10] W. Müller, "On the analytic continuation of rank one eisenstein series," *Geometric & Functional Analysis GAFA*, vol. 6, no. 3, pp. 572–586, 1996.

[11] M. Reed and B. Simon, *I: Functional Analysis.* Methods of Modern Mathematical Physics, Elsevier Science, 1981.

[12] M. Taylor, *Partial Differential Equations I: Basic Theory.* Applied Functional Analysis: Applications to Mathematical Physics, Springer, 1996.

[13] J. Lee, *Introduction to Smooth Manifolds.* Graduate Texts in Mathematics, Springer, 2003.

[14] W. Rudin, *Real and complex analysis.* Mathematics series, McGraw-Hill, 1987.

[15] E. B. Davies, *Spectral Theory and Differential Operators.* Cambridge University Press, 1995. Cambridge Books Online.

[16] L. Hörmander, *The Analysis of Linear Partial Differential Operators II: Differential Operators with Constant Coefficients.* Classics in Mathematics, Springer, 2004.

[17] "NIST Digital Library of Mathematical Functions." http://dlmf.nist.gov/, Release 1.0.10 of 2015-08-07. Online companion to [18].

[18] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, eds., *NIST Handbook of Mathematical Functions.* New York, NY: Cambridge University Press, 2010. Print companion to [17].

[19] Z. M. Christiansen, Tanya, "Spectral asymptotics for manifolds with cylindrical ends," *Annales de l'institut Fourier*, vol. 45, no. 1, pp. 251–263, 1995.

[20] T. J. Christiansen, *Scattering theory on compact manifolds with boundary.* PhD thesis, Massachusetts Institute of Technology. Dept. of Mathematics.

[21] R. B. Melrose, *The Atiyah-Patodi-Singer index theorem*, vol. 4 of *Research Notes in Mathematics.* A K Peters, Ltd., Wellesley, MA, 1993.

[22] L. Guillopé, "Théorie spectrale de quelques variétés à bouts," *Ann. Sci. École Norm. Sup. (4)*, vol. 22, no. 1, pp. 137–160, 1989.

[23] R. Mazzeo and A. Vasy, "Analytic continuation of the resolvent of the laplacian on symmetric spaces of noncompact type," *Journal of Functional Analysis*, vol. 228, no. 2, pp. 311 – 368, 2005.

[24] A. Strohmaier, "Analytic continuation of resolvent kernels on noncompact symmetric spaces," *Mathematische Zeitschrift*, vol. 250, no. 2, pp. 411–425, 2005.

[25] G. Jones and D. Singerman, *Complex Functions: An Algebraic and Geometric Viewpoint.* Cambridge University Press, 1987.

[26] M. Reed and B. Simon, *Methods of modern mathematical physics: Analysis of operators.* Methods of Modern Mathematical Physics, Academic Press, 1978.

[27] A. Girouard and I. Polterovich, "Spectral Geometry of the Steklov Problem.".

[28] P. Ŝolín, *Partial Differential Equations and the Finite Element Method.* Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts, Wiley, 2005.

[29] P. Tong and J. Rossettos, *Finite-element method: basic technique and implementation.* MIT Press, 1977.

[30] D. C. Sorensen, "Implicitly restarted arnoldi/lanczos methods for large scale eigenvalue calculations," 1996.

[31] R. B. Lehoucq, D. C. Sorensen, and C. Yang, "Arpack users guide: Solution of large scale eigenvalue problems by implicitly restarted arnoldi methods.," 1997.

[32] G. Golub and C. Van Loan, *Matrix Computations.* Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 1996.

[33] D. V. Evans, M. Levitin, and D. Vassiliev, "Existence theorems for trapped modes," *Journal of Fluid Mechanics*, vol. 261, pp. 21–31, 2 1994.

[34] E. B. Davies and L. Parnovski, "Trapped modes in acoustic waveguides," *Quart. J. Mech. Appl. Math.*, vol. 51, no. 3, pp. 477–492, 1998.

[35] H. Hawkins and L. Parnovski, "Trapped modes in a waveguide with a thick obstacle," *Mathematika*, vol. 51, no. 1-2, pp. 171–186 (2005), 2004.

[36] E. R. Johnson, M. Levitin, and L. Parnovski, "Existence of eigenvalues of a linear operator pencil in a curved waveguide—localized shelf waves on a curved coast," *SIAM J. Math. Anal.*, vol. 37, no. 5, pp. 1465–1481 (electronic), 2006.

[37] R. B. Melrose, "Polynomial bound on the distribution of poles in scattering by an obstacle," *Journes quations aux drives partielles*, pp. 1–8, 1984.

[38] M. Zworski, "Sharp polynomial bounds on the number of scattering poles of radial potentials," *Journal of Functional Analysis*, vol. 82, no. 2, pp. 370 – 403, 1989.

[39] L. Guillope and M. Zworski, "Scattering asymptotics for riemann surfaces," *Annals of Mathematics*, vol. 145, no. 3, pp. pp. 597–660, 1997.

[40] A. M. Aslanyan and E. B. Davies, "Separation of variables in perturbed cylinders," Tech. Rep. math.SP/0012113, Dec 2000.

[41] D. Borthwick, "Distribution of resonances for hyperbolic surfaces," *Exp. Math.*, vol. 23, no. 1, pp. 25–45, 2014.

[42] L. N. Trefethen and T. Betcke, "Computed eigenmodes of planar regions," in *Recent advances in differential equations and mathematical physics*, vol. 412 of *Contemp. Math.*, pp. 297–314, Amer. Math. Soc., Providence, RI, 2006.

[43] T. Betcke and L. N. Trefethen, "Reviving the method of particular solutions," *SIAM Rev.*, vol. 47, no. 3, pp. 469–491 (electronic), 2005.

[44] S. A. Nazarov, K. Ruotsalainen, and P. Uusitalo, "Bound states of waveguides with two right-angled bends," *J. Math. Phys.*, vol. 56, no. 2, pp. 021505, 24, 2015.

[45] E. P. Wigner, "Lower limit for the energy derivative of the scattering phase shift," *Phys. Rev.*, vol. 98, pp. 145–147, Apr 1955.

[46] L. Eisenbud, *Dissertation, Unpublished.* PhD thesis, Princeton University.

[47] A. Strohmaier, "Private conversation."

[48] H. Si, "Tetgen, a delaunay-based quality tetrahedral mesh generator," *ACM Trans. Math. Softw.*, vol. 41, pp. 11:1–11:36, Feb. 2015.