Vibration of Elastic Plates Submerged in a Fluid with a Free Surface

November 2009

Michael J. A. Smith

Supervised by

Dr. Michael H. Meylan

A dissertation submitted for the degree of
Bachelor of Science (Honours) in Applied Mathematics
at the University of Auckland, New Zealand

Abstract

The vibration of an elastic plate by floating and submerged bodies, and the associated problem of wave scattering, is a problem of great interest in the fields of engineering and mathematics, with applications in offshore platform construction and shipbuilding design, as well as in the vibration of icebergs and sea ice. This field is generally known as hydroelasticity.

We investigate the response of floating elastic plates, subject to a wave forcing, using linearised governing equations. Using Green's functions, we formulate the problem in terms of integral equations, which are then solved with the use of numerical methods, including boundary element methods.

A solution is presented for the vibration of an elastic plate of variable stiffness and mass, which floats on the surface of a fluid, with negligible submergence. The solution method is based closely on the equivalent problem for a floating elastic plate of constant properties. We show that the solution for a plate of variable properties can be calculated from the solution of a uniform plate, by expressing the natural modes for the variable plate in terms of the modes for a constant plate.

A solution is also presented for the problem of a submerged thin beam. This is analogous to solving the problem of wave scattering by a submerged thin structure when the beam is rigid. The solution method for this problem is also closely related to the problem for a floating uniform plate, but with the additional consideration of hypersingular integral equations.

Acknowledgements

The author would like to acknowledge the support and dedication of Dr Mike Meylan, an excellent supervisor from whom I have learnt a great deal.

The author would also like to thank the University of Auckland for the financial support given by a University of Auckland PGDip/Honours/Masters Scholarship.

Acknowledgements also to Dr Howie Cohl for his editorial assistance.

Contents

List of Figures

Nomenclature

Φ	-	fluid potential
ϕ	-	potential in the frequency domain
ϕ^{I}	-	incident potential
ϕ^{S}	-	scattered potential
ϕ^{D}	-	diffraction/diffracted potential
ϕ^{R}	-	radiated potential
$\partial_x^j \phi$	-	denotes $\frac{\partial^j \phi}{\partial x^j}$
∂_n	-	denotes $\frac{\partial}{\partial n}$
Re	-	real part of an expression
χ	-	displacement of the beam
ζ	-	displacement in the frequency domain
ρ	-	density of medium
g	-	gravitational constant
P	-	pressure
D	-	stiffness function for a beam
β	-	non-dimensional stiffness
m	-	linear mass density function for a beam
γ	-	non-dimensional linear mass density
H	-	dimensional depth of medium
h	-	non-dimensional depth of medium
L_1	-	dimensional length of beam
L	-	non-dimensional length of beam
ω	-	frequency of incident wave
α	-	ω^2
k_n	-	positive real solution of dispersion relation $k\tanh(kh) = \alpha$
k_0	-	first imaginary solution of dispersion relation $k\tanh(kh) = \alpha$
R	-	reflection coefficient
T	-	transmission coefficient
X_n	-	eigenfunctions for uniform floating beam
μ_n	-	eigenvalues for uniform floating beam
$\epsilon[v]$	-	energy functional

1 Introduction

The problem of a floating elastic plate, subject to linear wave forcing has been studied for a considerable time and is discussed in some detail by Stoker (1957) , where the solution for shallow water is given for both a finite and semi-infinite plate. For non-shallow water, the solution for a semi-infinite plate was first found by Fox & Squire (1994), and for a finite plate by Meylan & Squire (1994); Newman (1994). Recently, the floating plate problem has received considerable research attention, motivated largely by the construction, or proposed construction, of very large floating structures (VLFS). These are used for industrial space, airports, storage facilities and habitation. We do not discuss in detail the numerous methods developed to solve these problems, instead readers are referred to the excellent review articles by Watanabe *et al.* (2004); Squire (2007).

The most straightforward method to solve the linear response of a variable hydroelastic body is to generalize the method used to solve for a rigid body. In this method the body motion is solved by an expansion in the rigid body modes, and the effect of the fluid is found by solving an integral equation over the wetted surface of the body, using the Green's function for the fluid (Mei, 1989; Newman, 1977). To include the effect of the elasticity, the elastic modes, as well as the rigid body modes, need to be included. This method is the basis of the solution we present here.

Current computational methods allow for the solution of very complicated hydroelastic problems, for example, to calculate the response of a container ship. However, there still exists a need for methods to solve simple hydroelastic problems. The case of a floating plate in two-dimensions is exactly one such well studied problem. The solution for a plate of variable thickness has been developed by Bennetts *et al.* (2007) and by Meylan & Sturova (2009).

To consider the problem of wave scattering in general, we suspend an object in a medium, and subject it to an incident forcing wave. This incident wave will be scattered by the structure, and the object itself will vibrate, causing the propagation of waves into the medium.

This work primarily focuses around the use of linearised models for the fluid flow and beam response. Although the actual problem is non-linear, we assume that the amplitude of the waves is much smaller than their wavelength, which admits the use of linearised equations for fluid flow. We can also assume that the displacement of the mean is small compared to these waves, which allows the use Euler-Bernoulli beam theory, as opposed to more complicated theories, such as Timoshenko beam theory (Rao, 1986).

More specifically, this work applies the ideas on uniform floating beams from Stoker (1957) to the more complicated problem of a floating elastic plate of variable stiffness and mass. The technique for computing the response of a floating body focuses primarily on the use of integral equations (particularly Green's second identity). An additional extension is made to this theory for the case of a submerged thin plate of varying stiffness and mass, however, the integral equation formulation gives rise to added complications, including hypersingular integrals. The solution for the problem of wave scattering by soft and hard bodies is discussed in detail by Yang (2002).

The primary goal of this dissertation is to solve the response of a submerged variable plate, subject to an incident wave forcing. This solution is considered in a number of stages. Firstly, the solution method for a finite elastic plate of constant properties floating on water is considered. Using concepts from this established theory, the solution for a floating variable plate is then considered. The final goal is to consider the vibration of a submerged thin beam of variable stiffness and mass. Unfortunately, while the theory has been fully developed, we have only developed computational methods for a rigid plate (which contains most numerical difficulties).

2

Solution for a uniform free beam floating on water

We begin with the problem of a two dimensional uniform elastic beam, which is floating on the surface of a fluid, where the domain is of constant finite depth, H. The solution presented here is closely related to the solution method presented by Newman (1994).

2.1 Outline of beam theory

Our beam is modelled using the Euler-Bernoulli beam equation:

$$D\partial_x^4\chi + m\partial_t^2\chi = P, \tag{2.1}$$

where χ is the displacement, D is the stiffness, m is the mass per unit length of the beam, P denotes pressure, and the notation $\partial_x^j\chi$ denotes the partial derivative $\frac{\partial^j\chi}{\partial x^j}$. We also have conditions at the ends of the beam, which are assumed free here (although other edge conditions could be easily included in the current formulation). The free conditions are

$$\partial_x^2\chi = \partial_x^3\chi = 0, \;\; x = \pm L_1. \tag{2.2}$$

2.2 Equations in the time domain

We express our problem in terms of a set of linearised equations, assuming that the fluid flow is irrotational and inviscid, and that the amplitude of the fluid and structure motion is small. This gives rise to the equations for the problem as follows:

$$\Delta\Phi = 0, \; -H < z < 0, \tag{2.3a}$$

$$\partial_z\Phi = 0, \; z = -H, \tag{2.3b}$$

$$\partial_z\Phi = \partial_t\chi, \quad z = 0, \quad \text{and} \tag{2.3c}$$

$$-\rho(g\chi + \partial_t\Phi) = P, \quad z = 0, \tag{2.3d}$$

where Φ is the velocity potential for the fluid, Δ is the Laplace Operator ($\Delta\Phi = \partial_x^2\Phi + \partial_z^2\Phi$), ρ is the fluid density, and g is the acceleration due to gravity. Note that equation (2.3a) is Laplace's equation for an irrotational, inviscid medium, (2.3b) is the condition that there is no flow through the seabed (at depth H), (2.3c) is the kinematic condition, and (2.3d) is the dynamic condition for the plate.

We assume that the pressure is zero except under the plate which occupies the region $-L_1 \leq x \leq L_1$. Under the plate the dynamic condition, from the Euler-Bernoulli beam equation, is

$$D\partial_x^4\chi + m\partial_t^2\chi = -\rho(g\chi + \partial_t\Phi), \; x \in (-L_1, L_1), \quad z = 0. \tag{2.4}$$

A diagram outlining the problem is shown below:

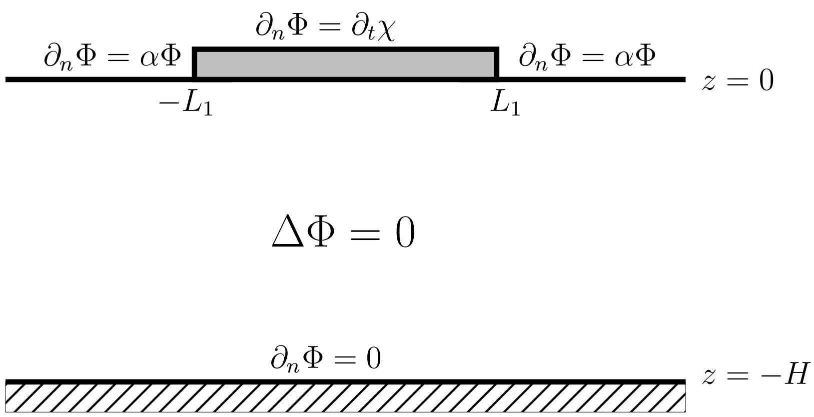

Figure 2.1: Floating elastic beam in a semi-infinite domain

The condition (2.3b) can be alternatively expressed in the form $\partial_z \Phi\big|_{z=-H} = 0$, as can expressions (2.3c) – (2.3d).

2.3 Non-dimensionalisation and transformation to the frequency domain

We non-dimensionalise all lengths with respect to an arbitrary length parameter L^*, and non-dimensionalise time with respect to $\sqrt{L^*/g}$ (so that gravity is unity). L denotes the non-dimensional length of the beam ($L = L_1/L^*$), and h denotes the non-dimensional depth ($h = H/L^*$). We now consider the equations in the frequency domain and assume that all quantities are proportional to $e^{i\omega t}$ so that we can write

$$\chi(x,t) = \mathrm{Re}\left\{\zeta \mathrm{e}^{i\omega t}\right\}, \tag{2.5a}$$

$$\Phi(x,t) = \mathrm{Re}\left\{\phi \mathrm{e}^{i\omega t}\right\}, \tag{2.5b}$$

where Re denotes the real part of the expression.

Under this assumption, equations (2.3) and (2.4) become

$$\Delta\phi = 0,\ -h < z < 0, \tag{2.6a}$$

$$\partial_z \phi = 0,\ z = -h, \tag{2.6b}$$

$$\partial_z \phi = i\omega\zeta,\ x \in (-L, L), \quad z = 0, \tag{2.6c}$$

$$\partial_z \phi = \alpha\phi,\ x \notin (-L, L), \quad z = 0, \quad \text{and} \tag{2.6d}$$

$$\beta\partial_x^4\zeta - (\gamma\alpha - 1)\,\zeta - \alpha\phi = 0,\ x \in (-L, L), \quad z = 0, \tag{2.6e}$$

where $\alpha = \omega^2$, $\beta = D/\rho g L^{*4}$, and $\gamma = m/\rho L^*$. The edge conditions (2.2) also apply. Note that expression (4.1) can be found in Tayler (1986).

2.4 Boundary conditions at $x \to \pm\infty$

Equations (2.6a) to (4.1) also require boundary conditions as $x \to \pm\infty$. To investigate the wave scattering from our structure, we consider an incident wave of unit amplitude (in potential) incoming from the left. This gives rise to boundary conditions involving an incident wave, plus corresponding reflected and transmitted waves from our plate. These imply that

$$\lim_{x\to-\infty}\phi = \frac{\cosh(k(z+h))}{\cosh(kh)}e^{-ikx} + R\frac{\cosh(k(z+h))}{\cosh(kh)}e^{ikx}, \tag{2.7}$$

where k is the first positive, real solution of the dispersion relation: $k\tanh(kh) = \alpha$, and R is the reflection coefficient. We also have

$$\lim_{x\to\infty}\phi = T\frac{\cosh(k(z+h))}{\cosh(kh)}e^{-ikx}, \tag{2.8}$$

where T is the transmission coefficient. These expressions for the potential, as $|x| \to \infty$, come from separation of variables. Details for this can be found in Linton & McIver (2001).

2.5 Modes of vibration for a uniform beam

To determine the modes of vibration for our plate, we now expand the displacement of the beam in terms of the modes of vibration, and obtain the following series:

$$\zeta = \sum_{n=0}^{N} \zeta_n X_n, \tag{2.9}$$

where the beam eigenfunctions satisfy the eigenvalue equation

$$\partial_x^4 X_n = \mu_n^4 X_n, \tag{2.10}$$

with the free edge conditions

$$\partial_x^2 X_n = \partial_x^3 X_n = 0,\ \ x = \pm L. \tag{2.11}$$

These modes can be determined analytically, and are given by

$$X_0 = \frac{1}{\sqrt{2L}}, \tag{2.12a}$$

$$X_1 = \sqrt{\frac{3}{2L^3}}x, \tag{2.12b}$$

$$X_{2n}(x) = \frac{1}{\sqrt{2L}}\left(\frac{\cos(\mu_{2n}x)}{\cos(\mu_{2n}L)} + \frac{\cosh(\mu_{2n}x)}{\cosh(\mu_{2n}L)}\right), \tag{2.12c}$$

and

$$X_{2n+1}(x) = \frac{1}{\sqrt{2L}} \left(\frac{\sin(\mu_{2n+1}x)}{\sin(\mu_{2n+1}L)} + \frac{\sinh(\mu_{2n+1}x)}{\sinh(\mu_{2n+1}L)} \right), \tag{2.12d}$$

where $\mu_0 = \mu_1 = 0$ and μ_n for $n \geq 2$ are the roots of

$$(-1)^n \tan(\mu_n L) + \tanh(\mu_n L) = 0. \tag{2.12e}$$

Note that we have defined the beam eigenfunctions so that they satisfy the following normalising condition

$$\int_{-L}^{L} X_m(x) X_n(x) dx = \delta_{mn}, \tag{2.13}$$

where δ_{mn} is the Kronecker delta.

2.6 Radiated, diffracted, and incident potentials

For the purposes of computing the potential ϕ, we make use of the fact that all equations are linear, and split the computation of the potential into two parts:

$$\phi = \phi^{\mathrm{D}} + \phi^{\mathrm{R}},$$

where ϕ^{D} denotes the diffraction potential, and ϕ^{R} denotes the radiation potential. The diffraction potential relates to the wave scattering problem, as this is the potential assuming that the structure is completely rigid, whereas the radiation potential considers the response of the structure, and the resulting propagation of waves into the medium.

Note that ϕ^{D} also incorporates the incident potential, and can therefore be split into scattered and incident potentials ($\phi^{\mathrm{D}} = \phi^{\mathrm{I}} + \phi^{\mathrm{S}}$).

After expanding the displacement of the plate in terms of the modes (2.9), and applying this to equations (2.6a) – (2.6c) we obtain the following equations for the potential

$$\Delta\phi = 0, \ -h < z < 0, \tag{2.14a}$$

$$\partial_z \phi = 0, \ z = -h, \tag{2.14b}$$

$$\partial_z \phi = \alpha\phi, \ x \notin (-L, L), \quad z = 0, \quad \text{and} \tag{2.14c}$$

$$\partial_z \phi = i\omega \sum_{n=0}^{N} \zeta_n X_n, \ x \in (-L, L), \quad z = 0. \tag{2.14d}$$

From linearity the potential can be further expanded as

$$\phi = \phi^{\mathrm{D}} + \sum_{n=0}^{N} \zeta_n \phi_n^{\mathrm{R}}. \tag{2.15}$$

The radiation potentials satisfy the conditions:

$$\Delta \phi_n^{\mathrm{R}} = 0, \ -h < z < 0, \tag{2.16a}$$

$$\partial_z \phi_n^{\mathrm{R}} = 0, \ z = -h, \tag{2.16b}$$

$$\partial_z \phi_n^{\mathrm{R}} = \alpha \phi_n^{\mathrm{R}}, \ x \notin (-L, L), \ z = 0, \quad \text{and} \tag{2.16c}$$

$$\partial_z \phi_n^{\mathrm{R}} = i\omega X_n, \ x \in (-L, L), \ z = 0, \tag{2.16d}$$

plus the Sommerfeld radiation condition that:

$$\frac{\partial \phi_n^{\mathrm{R}}}{\partial x} \pm ik\phi_n^{\mathrm{R}} = 0, \ \text{as } x \to \pm\infty. \tag{2.17}$$

The diffracted potential satisfies

$$\Delta \phi^{\mathrm{D}} = 0, \ -h < z < 0, \tag{2.18a}$$

$$\partial_z \phi^{\mathrm{D}} = 0, \ z = -h, \tag{2.18b}$$

$$\partial_z \phi^{\mathrm{D}} = \alpha \phi^{\mathrm{D}}, \ x \notin (-L, L), \ z = 0, \quad \text{and} \tag{2.18c}$$

$$\partial_z \phi^{\mathrm{D}} = 0, \ x \in (-L, L), \ z = 0, \tag{2.18d}$$

with an additional radiation condition that:

$$\frac{\partial}{\partial x}\left(\phi^{\mathrm{D}} - \phi^{\mathrm{I}}\right) \pm ik\left(\phi^{\mathrm{D}} - \phi^{\mathrm{I}}\right) = 0, \ \text{as } x \to \pm\infty. \tag{2.19}$$

The incident potential is given to be

$$\phi^{\mathrm{I}} = \frac{\cosh(k(z+h))}{\cosh(kh)} e^{-ikx}, \tag{2.20}$$

which is unit amplitude with respect to potential, and travelling towards positive infinity.

Incidentally, the scattered potential satisfies

$$\Delta\phi^{\mathrm{S}} = 0,\ -h < z < 0, \tag{2.21a}$$

$$\partial_z\phi^{\mathrm{S}} = 0,\ z = -h, \tag{2.21b}$$

$$\partial_z\phi^{\mathrm{S}} = \alpha\phi^{\mathrm{S}},\ x \notin (-L, L),\ z = 0, \quad \text{and} \tag{2.21c}$$

$$\partial_z\phi^{\mathrm{S}} = -\partial_z\phi^{\mathrm{I}},\ x \in (-L, L),\ z = 0. \tag{2.21d}$$

We aim to determine the radiation and diffraction potential through integral equations, which require the use of Green's functions.

2.7 Solution for radiation and diffracted potentials using Green's functions

For this problem, we can use a special Green's function, which is given by the following system of equations

$$\Delta G(\mathbf{x}, \mathbf{x}') = \delta(x - x'),\ -h < z < 0, \tag{2.22a}$$

$$\partial_z G = \alpha G,\ z = 0, \quad \text{and} \tag{2.22b}$$

$$\partial_z G = 0,\ z = -h, \tag{2.22c}$$

plus the Sommerfeld radiation condition of no incoming waves. In our case, we denote our field points by $\mathbf{x} = (x, z)$ and our source points by $\mathbf{x}' = (x', z')$. We express the solution to our system above as the following series

$$G(\mathbf{x}, \mathbf{x}') = \sum_{n=0}^{\infty} a_n(x) f_n(z), \tag{2.23}$$

where $f_n(z)$ is chosen in such a way as to satisfy the boundary conditions above:

$$f_n(z) = \frac{\cos(k_n(z + h))}{N_n}. \tag{2.24}$$

Note that the normalization coefficient N_n is:

$$N_n = \sqrt{\frac{\cos(k_n h)\sin(k_n h) + k_n h}{2k_n}}. \tag{2.25}$$

The Green's function can then be expressed as

$$(\partial_x^2 + \partial_z^2)G = \delta(x - x')\delta(z - z'), \tag{2.26}$$

and the delta function can be expanded as follows

$$\delta(z - z') = \sum_{n=0}^{\infty} f_n(z) f_n(z'), \tag{2.27}$$

which gives rise to the expression

$$\sum_{n=0}^{\infty} (\partial_x^2 - k_n^2) a_n(x) f_n(z) = \delta(x - x') \sum_{n=0}^{\infty} f_n(z) f_n(z'). \tag{2.28}$$

Therefore in order to find the coefficients a_n, we must solve

$$(\partial_x^2 - k_n^2) a_n(x) = \delta(x - x') f_n(z'), \tag{2.29}$$

which has the solution

$$a_n(x) = \frac{e^{-|x-x'|k_n} f_n(z')}{2k_n}. \tag{2.30}$$

Consequently, we can derive the Green's function

$$G(\mathbf{x}, \mathbf{x}') = \sum_{n=0}^{\infty} -\frac{e^{-|x-x'|k_n}}{\cos(k_n h)\sin(k_n h) + k_n h} \cos(k_n(z+h))\cos(k_n(z'+h)). \tag{2.31}$$

We can write the solution to the Greens function (restricted to the surface) as

$$G(\mathbf{x}, \mathbf{x}') = \sum_{n=0}^{N} -\frac{e^{-k_n|x-x'|}}{\tan(k_n h) + h k_n \sec^2(k_n h)}. \tag{2.32}$$

Using this Green's function with Green's second identity

$$\int_{\Omega} (u\Delta v - v\Delta u)\, dV = \int_{\partial\Omega} (u\partial_n v - v\partial_n u)\, dS, \tag{2.33}$$

we can obtain integral equations for the radiated and diffracted potentials.

Substituting the Green's function and ϕ_n^{R} into (2.33) allows us to obtain

$$\phi_n^{\mathrm{R}}(x) = \int_{-L}^{L} G(x,x') \left(\alpha\phi_n^{\mathrm{R}}(x') - i\omega X_n(x')\right) dx', \quad \text{and} \tag{2.34a}$$

$$\phi^{\mathrm{D}}(x) = \phi^{I}(x) + \alpha \int_{-L}^{L} G(x,x')\phi^{\mathrm{D}}(x')dx'. \tag{2.34b}$$

The second expression, (2.34b), is obtained from using Green's second identity with the Green's function and ϕ^{S}, then adding ϕ^{I}. One way to evaluate integrals of this type is to use boundary element methods.

2.8 Boundary element methods

Boundary element methods are concerned principally with the numerical solution of field equations, and in particular, provide a technique for solving field equations which have been recast as boundary integral equations (where these integral equations often involve Green's functions, Cartwright (2001)). We consider boundary element methods in a general sense in this section, whereas the previous method is a kind of boundary element method with a very special Green's function.

Unlike finite element methods, we only recover our unknown on the boundary of the domain, which is more efficient for a floating structure, as the potential under the plate is all that is required.

The boundary element method is a particularly useful tool for solving elliptic problems such as Laplace's equation. Fundamentally, we divide our boundary into panels where ϕ is assumed constant over each panel, and G is integrated exactly over each panel

$$\int_{\mathbf{x}_i - l/2}^{\mathbf{x}_i + l/2} \phi(\mathbf{x}')G(\mathbf{x},\mathbf{x}')d\mathbf{s}' \approx \phi(\mathbf{x}_i) \int_{\mathbf{x}_i - l/2}^{\mathbf{x}_i + l/2} G(\mathbf{x}_i,\mathbf{x}')d\mathbf{s}', \tag{2.35}$$

where $\mathbf{x}_i$ denotes a panel midpoint, and l is the length of a given panel. This then generalises as follows

$$\int_{\partial\Omega} \phi(\mathbf{x}')G(\mathbf{x},\mathbf{x}')ds' = \sum_i \phi(\mathbf{x}_i) \int_{\mathbf{x}_i-l/2}^{\mathbf{x}_i+l/2} G(\mathbf{x}_i,\mathbf{x}')ds' = \mathbb{G}\vec{\phi}, \tag{2.36}$$

where

$$\mathbb{G}_{ij} = \int_{\mathbf{x}_i-l/2}^{\mathbf{x}_i+l/2} G(\mathbf{x}_i,\mathbf{x}_j')ds'. \tag{2.37}$$

From this, the integral equations (2.34a) and (2.34b) can be expressed in matrix form, admitting the solutions

$$\vec{\phi}_n^{\mathrm{R}} = -i\omega \left[\mathrm{I} - \alpha\mathbb{G}\right]^{-1} \mathbb{G}\vec{X}_n, \quad \text{and} \tag{2.38}$$

$$\vec{\phi}^{\mathrm{D}} = \left[\mathrm{I} - \alpha\mathbb{G}\right]^{-1} \vec{\phi}^{\mathrm{I}}. \tag{2.39}$$

Having determined the diffraction and radiation potential underneath our floating plate, we need to determine the displacement coefficients for our beam.

2.9 Solution for the beam

Previously we had a dynamic condition for the displacement of the plate, expressed in terms of the fluid potential. Substituting the solution for the radiated and diffracted potentials into equation (4.1) allows us to obtain the expression:

$$\sum_{n=0}^{N} \left(\beta\mu_n^4 - \alpha\gamma + 1\right)\zeta_n X_n = -i\omega\left(\phi^{\mathrm{D}} + \sum_{n=0}^{N}\zeta_n\phi_n^{\mathrm{R}}\right), \qquad \text{for all } m. \tag{2.40}$$

After multiplying by X_m, and integrating over the plate, we obtain:

$$\sum_{n=0}^{N} \left(\beta\mu_n^4 - \alpha\gamma + 1\right)\zeta_n = -i\omega\left(\int_{-L}^{L}\phi^{\mathrm{D}}X_m dx + \sum_{n=0}^{N}\zeta_n\int_{-L}^{L}\phi_n^{\mathrm{R}}X_m dx\right). \tag{2.41}$$

We now introduce the following notation defining the diagonal matrix $\mathbf{D}$ with elements

$$d_{mm} = \mu_m^4, \tag{2.42}$$

and $\mathbf{\Lambda}$ with elements

$$\Lambda_{mn} = i\omega\int_{-L}^{L}\phi_n^{\mathrm{R}}(x)X_m(x)dx, \tag{2.43}$$

as well as the vector $\vec{\mathbf{f}}$, with elements given by

$$f_n = -i\omega \int_{-L}^{L} \phi^{\mathrm{D}} X_n dx. \tag{2.44}$$

Then we can write equation (2.41) as

$$\left(\beta\mathbf{D} - (\alpha\gamma - 1)\,\mathbf{I} + \mathbf{\Lambda}\right)\vec{\zeta} = \vec{\mathbf{f}}, \tag{2.45}$$

from which we can easily obtain the displacement coefficients. Note that the real part of the matrix $\mathbf{\Lambda}$ is called the added mass matrix, and the imaginary part is known as the damping matrix.

2.10 Reflection and transmission coefficients

The reflection and transmission coefficients represent the ratio of the amplitude of a reflected or transmitted wave to the amplitude of the incident wave. They hold the property that $|R|^2+|T|^2 = 1$. The reflection and transmission coefficients can be computed using Green's second identity (2.33), in the same manner as equation (2.34a), but with the solution and ϕ^{I}, to obtain

$$R = \frac{\int_{-L}^{L} e^{k_0 x} \left(\alpha\phi(x) - \partial_n \phi(x)\right) dx}{2k_0 \int_{-h}^{0} \left(\phi_0(z)\right)^2 dz}. \tag{2.46}$$

Similarly, if we use a wave incident from the right, we obtain

$$T = -1 + \frac{\int_{-L}^{L} e^{-k_0 x} \left(\alpha\phi(x) - \partial_n \phi(x)\right) dx}{2k_0 \int_{-h}^{0} \left(\phi_0(z)\right)^2 dz}, \tag{2.47}$$

where $\phi^I(\mathbf{x}) = \phi_0(z)e^{-ikx}$.

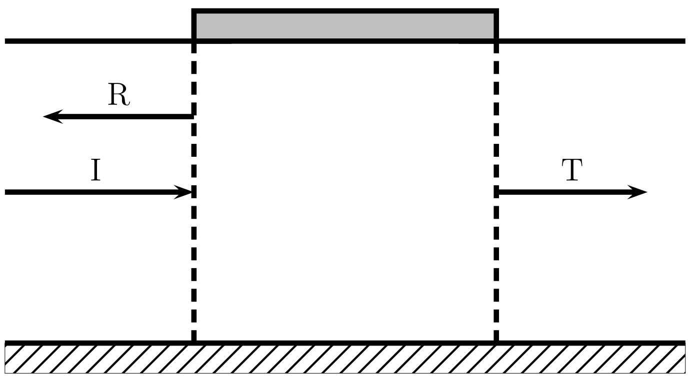

Figure 2.2: Diagram demonstrating reflection and transmission

2.11 Numerical example

To demonstrate the theory outlined in this chapter, we consider modelling a VLFS as a floating uniform beam, using parameters from Kyoung *et al.* (2005). These were taken to be $\alpha = 7.5$, $L_1 = 150$m, $H = 58.5$m, draft=0.5m, $\rho = 1000$ kg m^{-3}, $D = 8.75 * 10^9$ N-m, and 5 modes of vibration.

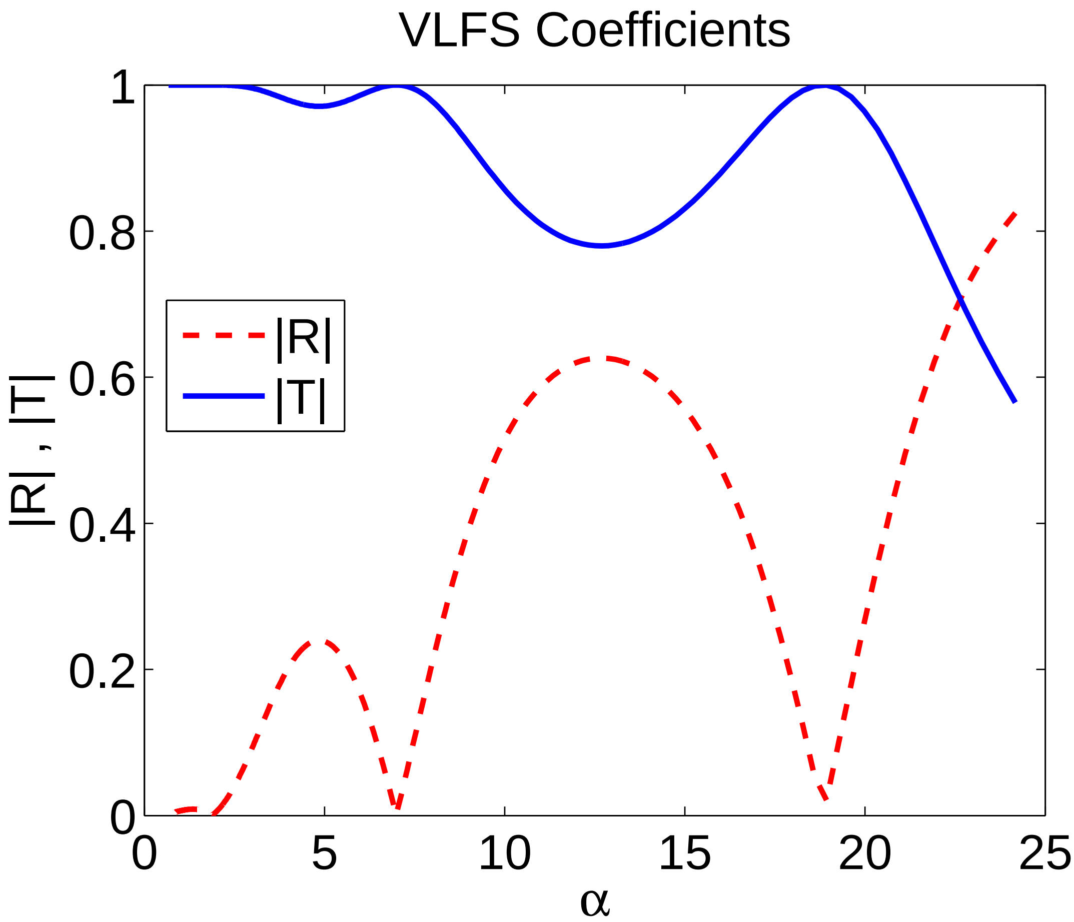

Figure 2.3: Reflection and transmission coefficients for a uniform beam where a VLFS is considered

Observe that for Figure (2.3), we observe near perfect transmission (zero reflection) near particular values of α, such as $\alpha = 2, 7.1$ and 19. In Figure (2.4), plots of the displacement for our case example are shown.

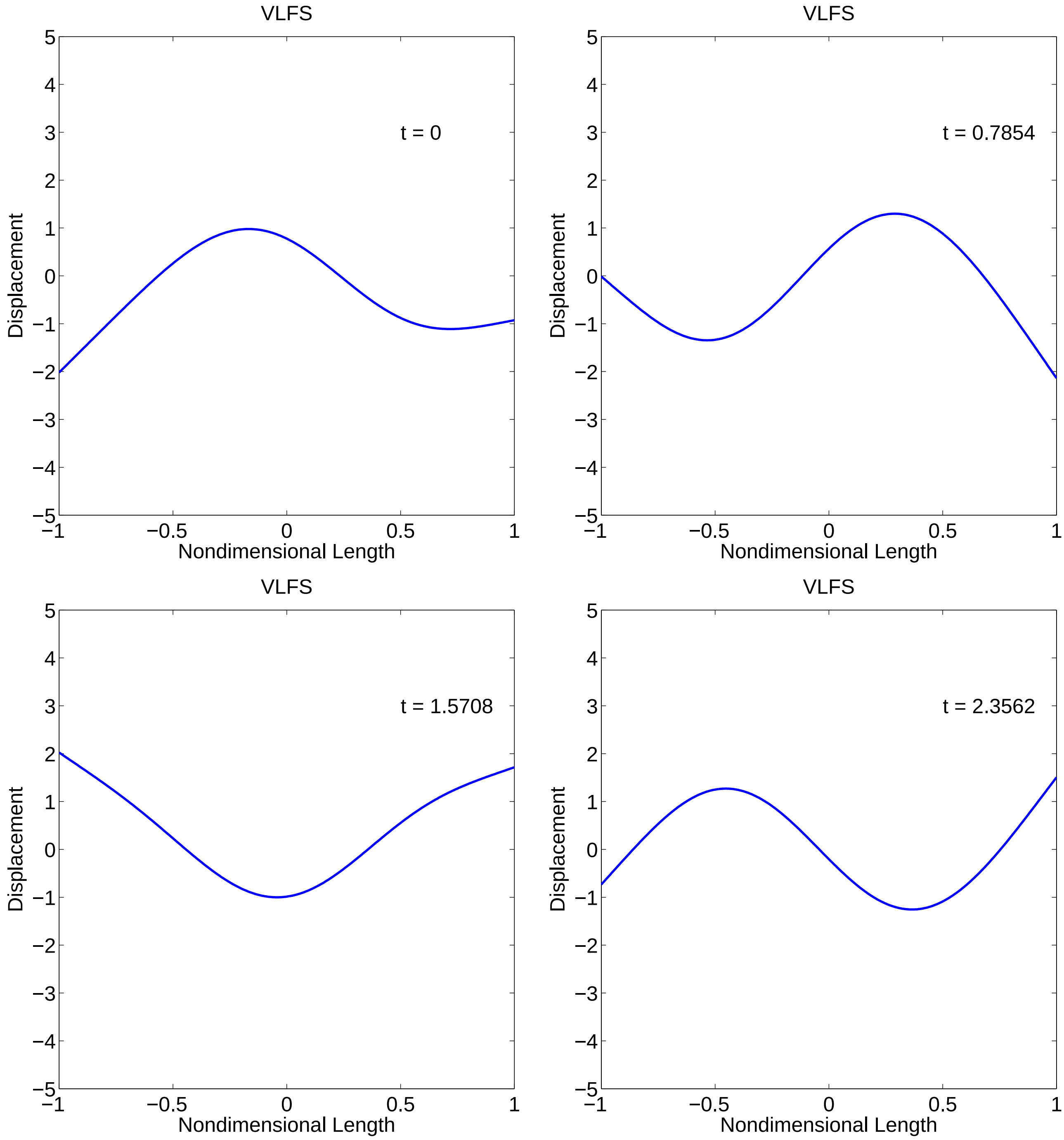

Figure 2.4: Hydroelastic response of a VLFS

3

Non-uniform free beam floating on water

As before, we consider a two dimensional beam floating on water (with negligible submergence) in a region of finite constant depth. However, we now extend our theory to the case of a non-uniform, or variable, elastic plate floating on water. The case of a non-uniform free beam has many practical applications, particularly in the modelling of sea ice, or large floating structures. The previous chapter on uniform beams was not original work, but the following solution method is original research, and has been submitted for publication (Meylan & Smith, 2009).

3.1 Eigenvalue problem

In the case of a non-uniform beam, we now use a more generalized form of the Euler-Bernoulli beam equation

$$\partial_x^2 \left(\beta(x)\partial_x^2\zeta\right) + \gamma(x)\partial_t^2\zeta = 0, \tag{3.1}$$

plus the edge conditions (2.2). The eigenfunctions for the non-uniform beam we denote by $\widehat{X}_n$ and the corresponding eigenvalues we denote by $\widehat{\mu}_n$. They satisfy the generalised eigenvalue problem

$$\partial_x^2 \left(\beta(x)\partial_x^2\widehat{X}_n\right) = \gamma(x)\widehat{\mu}_n^4\widehat{X}_n. \tag{3.2}$$

3.2 Variational principles

If we have a complicated eigenvalue problem, which is difficult to solve, we can express the problem in an equivalent variational form.

For example, the eigenvalue problem for a uniform beam (2.10), has the equivalent variational problem

$$\min \qquad R[X_n] = \frac{\beta}{2}\int_{-L}^{L} \left(\partial_x^2 X_n\right)^2 dx, \tag{3.3}$$

$$\text{subject to} \qquad \int_{-L}^{L} \gamma X_n^2 dx = 1. \tag{3.4}$$

This is obtained by minimising the Rayleigh quotient involving an energy functional for our uniform beam, as discussed in Linton & McIver (2001) and Lanczos (1949).

The stationery points of the variational problem above correspond to the eigenfunctions for the eigenvalue problem (2.10).

For the eigenfunction problem (3.2), we have the equivalent variational problem

$$J[\zeta] = \frac{1}{2}\int_{-L}^{L} \left\{\beta(x)(\partial_x^2\zeta)^2 - \widehat{\mu}^4\gamma(x)\zeta^2\right\}dx, \tag{3.5}$$

where $\widehat{\mu}$ also doubles as the Lagrange multiplier. We can find these non-uniform eigenfunctions using the Rayleigh-Ritz method, expanding in terms of the eigenfunctions for a free beam.

3.3 Rayleigh-Ritz method

The Rayleigh-Ritz method is a widely used method in mechanics, used to simplify the problem of finding the stationery points of a variational problem (as in expression (3.5)). We essentially replace the variational problem of finding a $y(x)$ that extremises $J[y]$ to finding a set of constants $a_1, a_2, ..., a_N$ that extremise $J[a_1, a_2, ..., a_N]$, then solving

$$\frac{\partial}{\partial a_k} J[a_1, a_2, ..., a_N] = 0, \tag{3.6}$$

for all $k = 1, 2, ..., N$. The Rayleigh-Ritz method gives a close approximation to the exact natural modes of a variable plate, providing that a suitable ansatz function has been chosen. In our example of a non-uniform beam, we write

$$\widehat{X}_m = \sum_{n=0}^{N} a_{mn} X_n(x), \tag{3.7}$$

and obtain:

$$J[a_{mn}] = \frac{1}{2} \int_{-L}^{L} \left\{ \beta(x) \left[\sum_{m=0}^{M} a_{mn} X_n'' \right]^2 - \widehat{\mu}_m^4 \gamma(x) \left[\sum_{m=0}^{M} a_{mn} X_n \right]^2 \right\} dx. \tag{3.8}$$

Then extremising $J[a_{mn}]$ allows us to obtain

$$\frac{\partial}{\partial a_m} J[a_{mn}] \quad = \quad \sum_{m=0}^{M} (k_{mn} - \widehat{\mu}_m^4 m_{mn}) a_{mn} = 0, \tag{3.9}$$

(for all $m = 0, 1, ...M$) where we define

$$k_{mn} = \int_{-L}^{L} \beta(x) X_m'' X_n'' dx, \quad \text{and} \tag{3.10a}$$

$$m_{mn} = \int_{-L}^{L} \gamma(x) X_m X_n dx, \tag{3.10b}$$

which can be expressed in the following form with matrices $\mathbf{K}$ and $\mathbf{M}$:

$$\mathbf{K}\mathbf{a}_m = \widehat{\mu}_m^4 \mathbf{M}\mathbf{a}_m. \tag{3.11}$$

The a_{mn} coefficients are found by solving the generalised eigenvalue equation above, from which we can form the matrix $\mathbf{A}$ with columns given by the generalised eigenvectors $\mathbf{a}_m$. Note that this method is analogous to a finite element method in one dimension.

MATLAB is used to implement the solution for this problem.

3.4 Numerical integration

To evaluate each element of the matrices **K** and **M**, we can employ composite Simpson's rule (Chapra, 2008)

$$\begin{aligned}\int_a^b f(x)dx &\approx \frac{h}{3}\left[f(x_0)+2\sum_{j=1}^{n/2-1} f(x_{2j})+4\sum_{j=1}^{n/2} f(x_{2j-1})+f(x_n)\right]\\ &\approx \frac{h}{3}\left[f(x_0)+4f(x_1)+2f(x_2)+4f(x_3)+2f(x_4)+...+4f(x_{n-1})+f(x_n)\right].\end{aligned}$$

This allows us to express k_{mn} as follows (with weights w_h)

$$\begin{aligned}k_{mn}=\int_{-L}^{L}\beta(x)X_m''X_n''dx &\approx \sum\beta(x_h)X_m''(x_h)X_n''(x_h)w_h\\ &\approx \vec{X}_m''H\vec{X}_n''^{T}, \end{aligned}\tag{3.12}$$

where

$$\begin{aligned}\vec{X}_m'' &= [X_m''(x_1),X_m''(x_2),...,X_m''(x_h)],\\ H &= \begin{pmatrix}\gamma(x_1)w_1 & 0 & ... & 0\\ 0 & \gamma(x_2)w_2 & ... & 0\\ \vdots & & \ddots & \vdots\\ 0 & ... & ... & \gamma(x_h)w_h\end{pmatrix},\end{aligned}$$

and the weights are defined as: $w_1=h/3, w_2=4h/3,...,w_h=h/3$.

We can extend this concept to the the full matrix **K** if we form

$$\mathbf{K} = X_{mat}''HX_{mat}''^{T},\tag{3.13}$$

where

$$X_{mat}'' = \begin{pmatrix}X_1''(x_1) & X_1''(x_2) & ... & X_1''(x_h)\\ X_2''(x_1) & X_2''(x_2) & ... & X_2''(x_h)\\ \vdots & \vdots & & \vdots\\ X_N''(x_1) & X_N''(x_2) & ... & X_N''(x_h)\end{pmatrix}.\tag{3.14}$$

For $\mathbf{M}$ we use an identical approach

$$\mathbf{M} = X_{mat} J X_{mat}^T, \tag{3.15}$$

where

$$J = \begin{pmatrix} \gamma(x_1)w_1 & 0 & ... & 0 \\ 0 & \gamma(x_2)w_2 & ... & 0 \\ \vdots & & \ddots & \vdots \\ 0 & ... & ... & \gamma(x_h)w_h \end{pmatrix}, \quad \text{and} \tag{3.16}$$

$$X_{mat} = \begin{pmatrix} X_1(x_1) & X_1(x_2) & ... & X_1(x_h) \\ X_2(x_1) & X_2(x_2) & ... & X_2(x_h) \\ \vdots & \vdots & & \vdots \\ X_N(x_1) & X_N(x_2) & ... & X_N(x_h) \end{pmatrix}. \tag{3.17}$$

Having determined the matrices $\mathbf{K}$ and $\mathbf{M}$, as well as solving the eigenvalue problem (3.11), we can determine the non-uniform modes of vibration. This allows us to expand the displacement in terms of the non-uniform modes to obtain

$$\zeta = \sum_{n=0}^{N} \widehat{\zeta}_n \widehat{X}_n, \tag{3.18}$$

and as the displacement was previously defined as

$$\widehat{X}_m = \sum_{n=0}^{N} a_{mn} X_n(x), \tag{3.19}$$

we can obtain the following relationship

$$\zeta_n = \mathbf{A}\widehat{\zeta_n} \quad \text{and} \quad \widehat{\zeta_n} = \mathbf{A}^{-1}\zeta_n. \tag{3.20}$$

In other words, we can express the displacement coefficients for a non-uniform beam in terms of the displacement coefficients for a uniform beam, allowing us to connect with established theory in the previous chapter.

3.5 Equations for a floating non-uniform beam

The equations for a non-uniform beam, floating on the fluid surface, are virtually the same as those given for a uniform beam (2.6a) to (4.1). If we apply these results, the governing equations can be expressed as

$$\Delta\phi = 0, \ -h < z < 0, \tag{3.21a}$$

$$\partial_z\phi = 0, \ z = -h, \tag{3.21b}$$

$$\partial_z\phi = \alpha\phi, \ x \notin (-L, L), \quad z = 0, \tag{3.21c}$$

$$\partial_z\phi = i\omega\sum_{n=0}^{\infty}\zeta_n\widehat{X}_n, \ x \in (-L, L), \quad z = 0, \quad \text{and} \tag{3.21d}$$

$$\partial_x^2\left(\beta(x)\sum_{n=0}^{\infty}\zeta_n\partial_x^2\widehat{X}_n\right) - (\gamma(x)\alpha - 1)\sum_{n=0}^{N}\zeta_n\widehat{X}_n + i\omega\phi = 0, \ x \in (-L, L), \ z = 0. \tag{3.21e}$$

From linearity the potential can be written as

$$\phi = \phi^{\mathrm{D}} + \sum_{n=0}^{N}\zeta_n\widehat{\phi}_n^{\mathrm{R}}, \tag{3.22}$$

where

$$\widehat{\phi}_n^{\mathrm{R}} = \sum_m a_{mn}\phi_n^{\mathrm{R}}. \tag{3.23}$$

Substituting this expression into equation (3.21e) gives the following

$$\sum_{n=0}^{N}\left(\widehat{\mu}_n^4\gamma(x) - \gamma(x)\alpha + 1\right)\zeta_n\widehat{X}_n = -i\omega\left(\phi^{\mathrm{D}} + \sum_{n=0}^{N}\zeta_n\widehat{\phi}_n^{\mathrm{R}}\right), \ x \in (-L, L), \ z = 0. \tag{3.24}$$

If we now multiply by $\widehat{X}_m$ and integrate from $-L$ to L, we obtain

$$\sum_{n=0}^{N}\left(\widehat{\mu}_n^4 - \alpha\right)\left[\int_{-L}^{L}\gamma(x)\widehat{X}_n\widehat{X}_m dx + \int_{-L}^{L}\widehat{X}_n\widehat{X}_m dx + i\omega\int_{-L}^{L}\widehat{\phi}_n^{\mathrm{R}}\widehat{X}_m dx\right]\zeta_n =$$
$$-i\omega\int_{-L}^{L}\phi^{\mathrm{D}}\widehat{X}_m dx, \ x \in (-L, L), \ z = 0. \tag{3.25}$$

Using the series expansion for $\widehat{X}_n$ at (3.7), and (3.20), we obtain the following equation

$$\left[(\widehat{\mathbf{D}} - \alpha\mathbf{I})\mathbf{A}^{\mathbf{T}}\mathbf{M}\mathbf{A} + \mathbf{A}^{\mathbf{T}}\mathbf{A} + \mathbf{A}^{\mathbf{T}}\mathbf{\Lambda}\mathbf{A}\right]\vec{\zeta} = \mathbf{A}\vec{\mathbf{f}}, \tag{3.26}$$

where the elements of the diagonal matrix $\widehat{\mathbf{D}}$ are given by

$$\widehat{d}_{mm} = \widehat{\mu}_m^4. \tag{3.27}$$

Equation (3.26) is closely related to equation (2.45).

3.6 Numerical examples

To demonstrate the theory outlined in this chapter, we consider the $\beta(x)$ and $\gamma(x)$ profiles outlined in the following graph. Using these profiles, we are able to compute the reflection and transmission coefficients for different multiples of γ and β. The reflection and transmission coefficients provide a useful way of comparing the behaviour of the beam from different density and mass profiles.

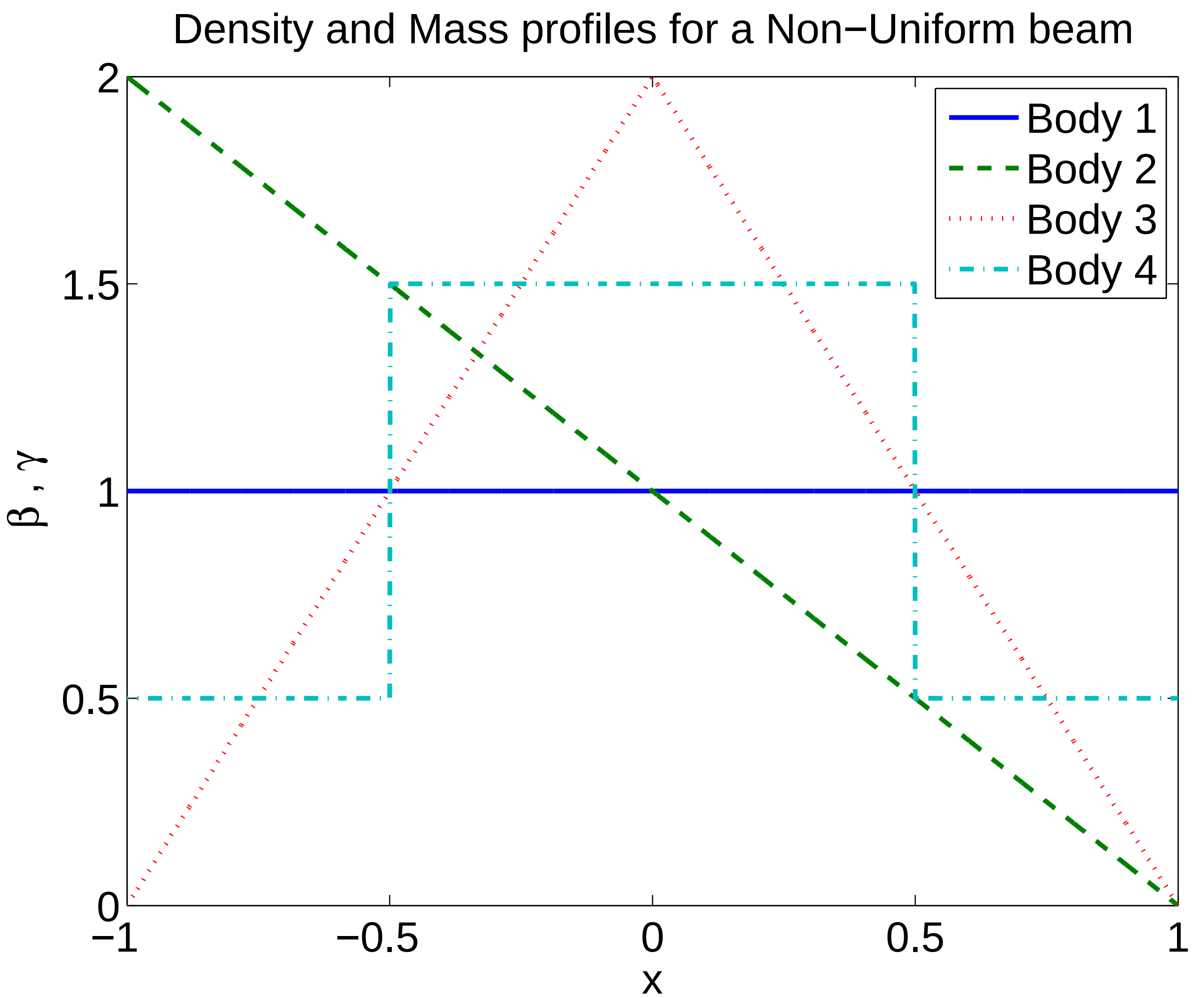

Figure 3.1: Density and mass function profiles for Bodies 1 to 4

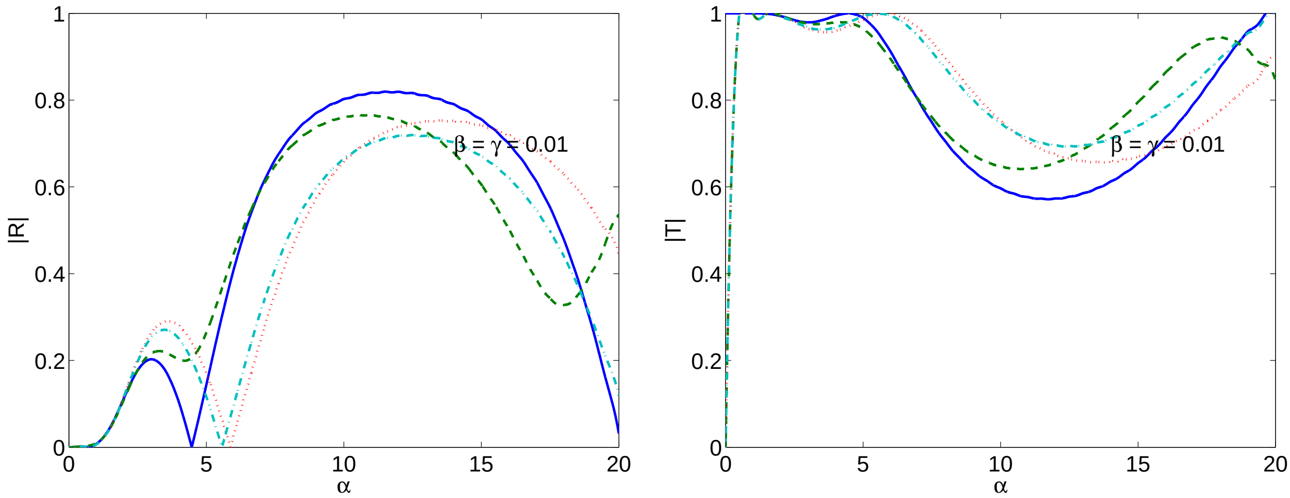

Figure 3.2: All bodies for $\beta = \gamma = 0.01$

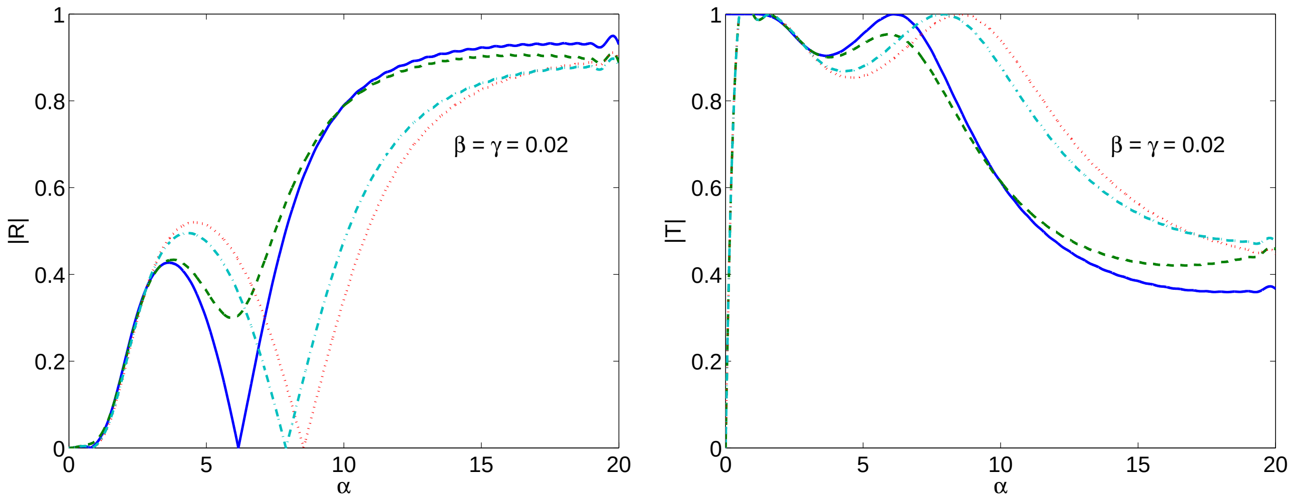

Figure 3.3: All bodies for $\beta = \gamma = 0.02$

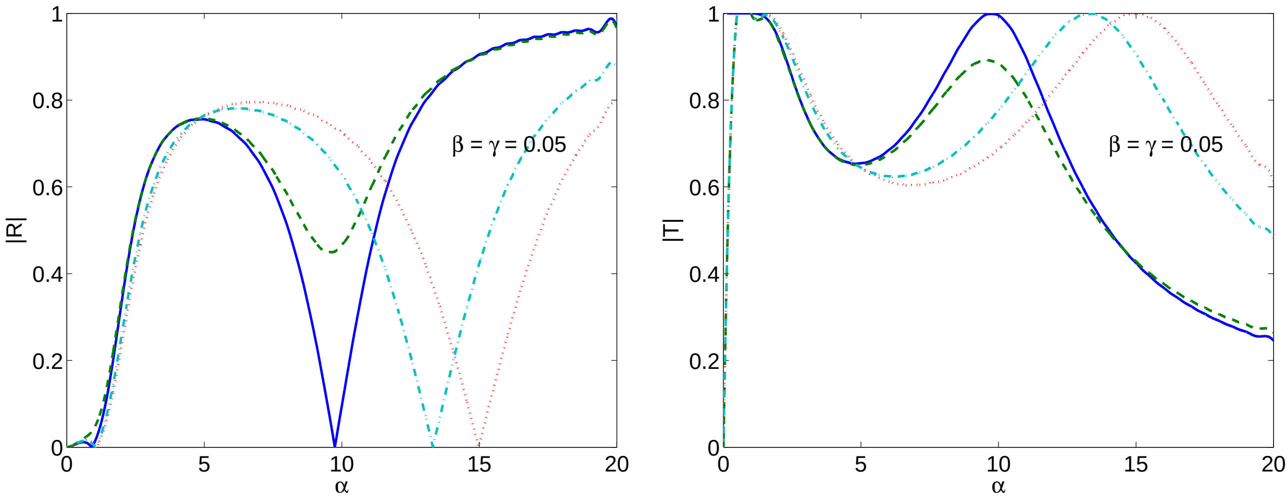

Figure 3.4: All bodies for $\beta = \gamma = 0.05$

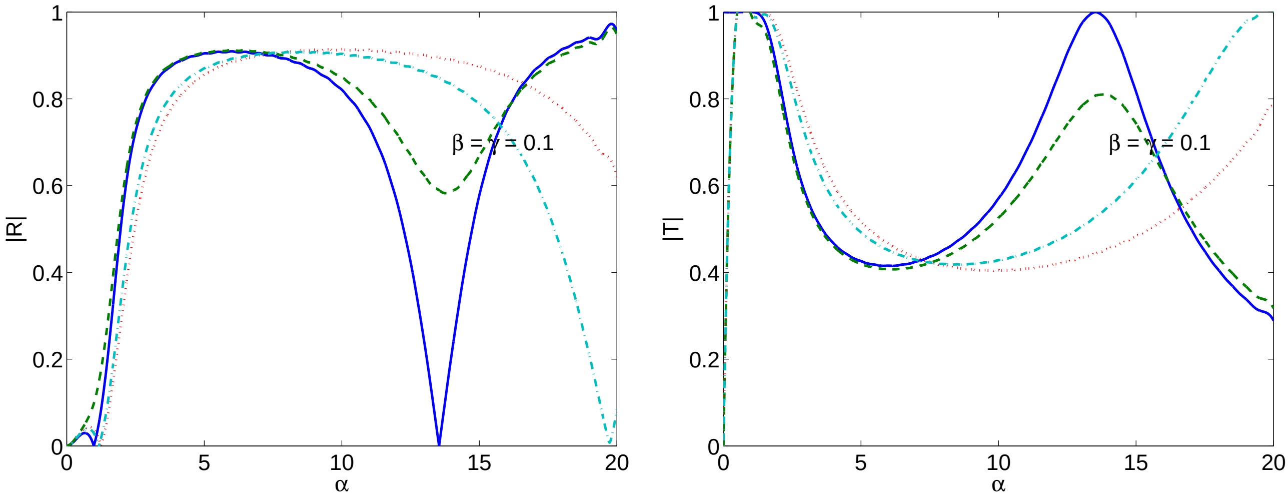

Figure 3.5: All bodies for $\beta = \gamma = 0.1$

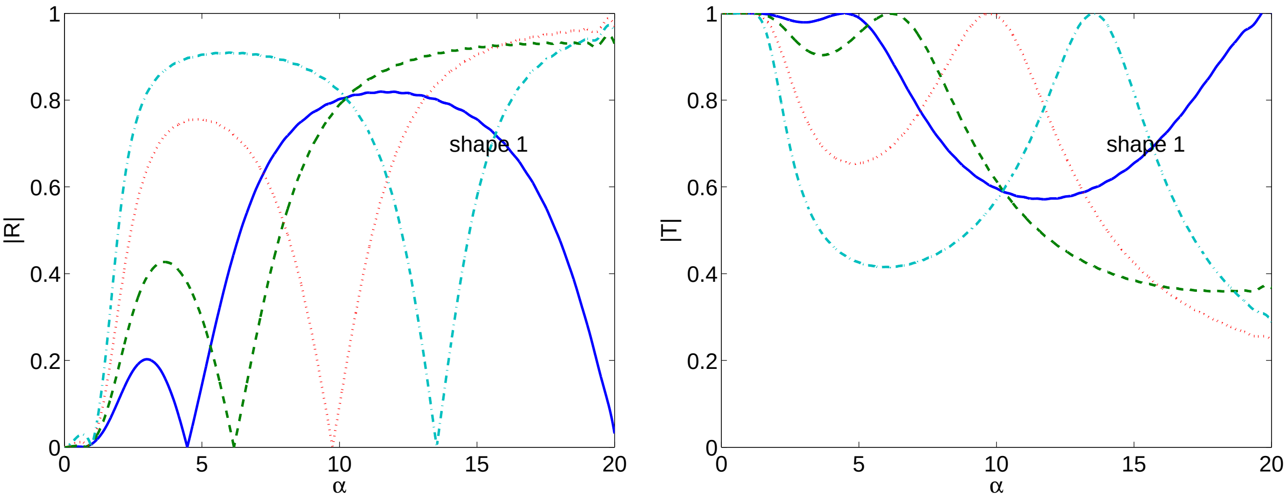

Figure 3.6: Body 1 for $\beta = \gamma = 0.01, 0.02, 0.05, 0.1$

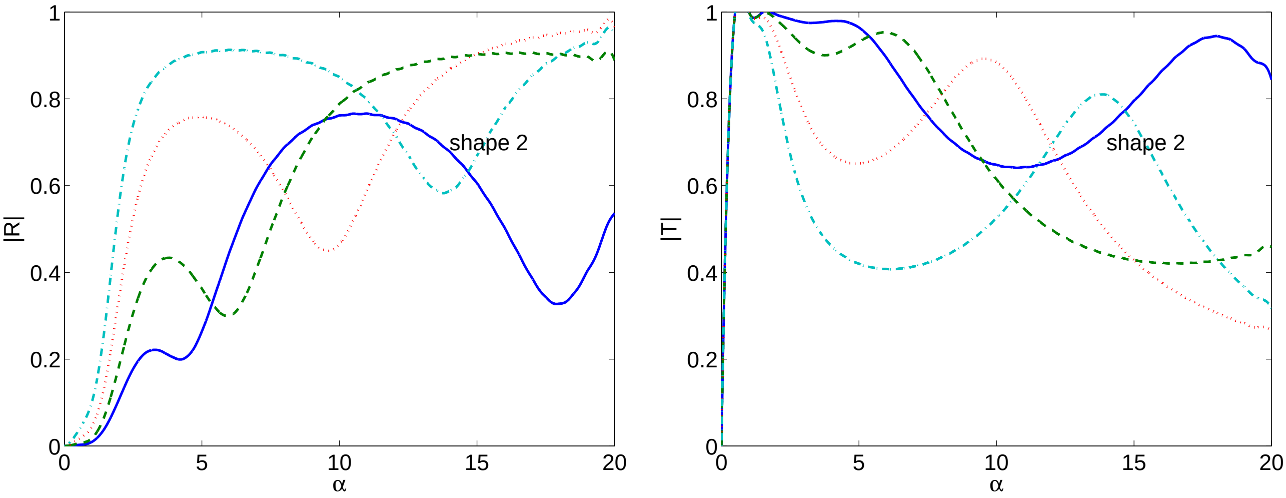

Figure 3.7: Body 2 for $\beta = \gamma = 0.01, 0.02, 0.05, 0.1$

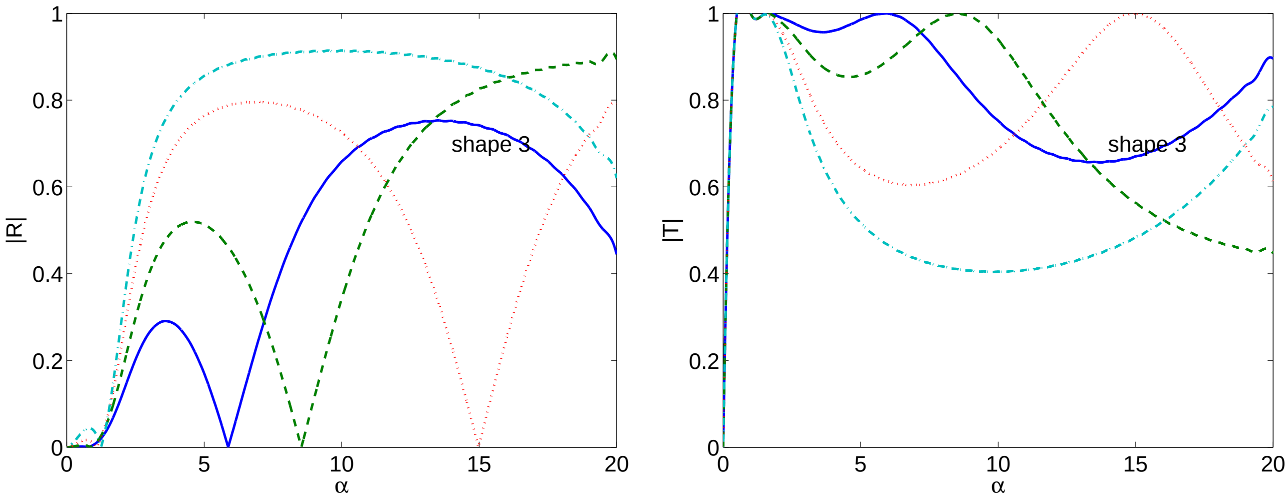

Figure 3.8: Body 3 for $\beta = \gamma = 0.01, 0.02, 0.05, 0.1$

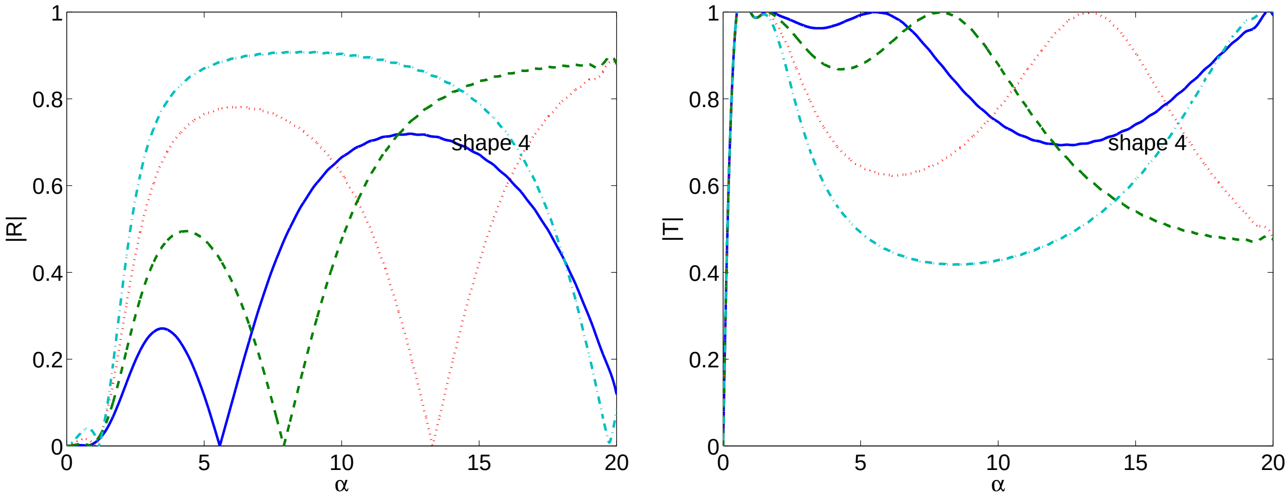

Figure 3.9: Body 4 for $\beta = \gamma = 0.01, 0.02, 0.05, 0.1$

4
Non-uniform submerged beam

The problem of determining the vibration of a submerged beam is a problem of great theoretical and practical importance, and is the ultimate aim of the project. Our beam under consideration is assumed to have negligible thickness (considered to be a thin body). This simple problem is, in fact, particularly challenging, as when formulating the integral equation form of the problem, we encounter hypersingular integral equations.

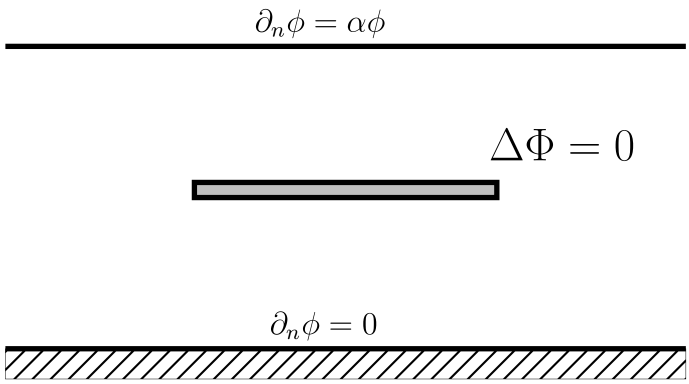

Figure 4.1: Submerged thin structure

We consider a two-dimensional beam in a semi-infinite region ($-\infty < x < \infty$ and $-h < z < 0$), completely submerged in a fluid. To the best knowledge of the author, this problem has never been solved.

4.1 Thin structures

When submerging a beam in a semi-infinite region, we need to impose artificial boundaries in order to utilise boundary element methods, which assume a closed domain. This effectively splits our domain into three parts. Our equations for the fluid flow in the central domain, Ω (immediately surrounding the structure), are as follows:

$$\Delta\phi = 0,\ (x,z) \in\ \Omega, \tag{4.1a}$$
$$\partial_z\phi = 0,\ z = -h, \tag{4.1b}$$
$$\partial_z\phi = \alpha\phi,\ z = 0, \tag{4.1c}$$
$$\phi = \tilde{\phi}_1(z),\ x = a_1, \quad \text{and} \tag{4.1d}$$
$$\phi = \tilde{\phi}_2(z),\ x = a_2, \tag{4.1e}$$

plus radiation conditions:

$$\frac{\partial}{\partial x}\phi \pm ik\phi = 0,\ \text{as } x \to \pm\infty. \tag{4.2}$$

Additionally, there is a kinematic condition for the response of the beam, which is submerged at constant depth

$$\partial_n\phi = g(x),\ x \in (-L, L),\ z = -d, \tag{4.3}$$

as well as a dynamic condition for the plate:

$$\sum_{n=0}^{N} \left(\widehat{\mu}_n^4\gamma(x) - \gamma(x)\alpha\right)\zeta_n\widehat{X}_n = -i\omega\left(\left[\ \phi^{\mathrm{D}}\ \right] + \sum_{n=0}^{N}\zeta_n\left[\ \phi_n^{\mathrm{R}}\ \right]\right), \quad x \in (-L, L), \quad z = -d. \tag{4.4}$$

This expression is closely related to (3.21e) for the case of a non-uniform floating beam, and for the case of a uniform floating beam (there is no restoring force for the case of a submerged beam).

For a non-uniform submerged beam, we can use:

$$\partial_n \phi = i\omega \widehat{X}_n,\ x \in (-L, L),\ z = -d, \tag{4.5}$$

and in the case of a rigid structure, we specify $g(x) = 0$.

Our problem can be illustrated by the following image:

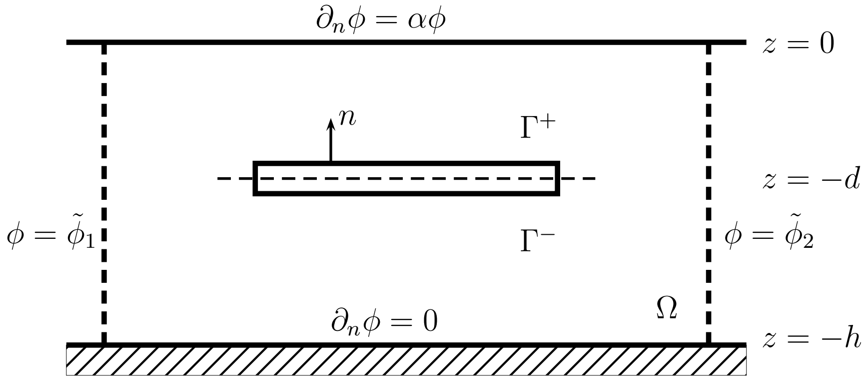

Figure 4.2: Submerged thin structure in a finite region

As the obstacle is thin, we need to split the object (denoted by Γ) into two regions ($\Gamma^{\pm}$), as well as distinguish their normal derivatives (n into $n^{\pm}$).

The boundary condition is now written as

$$\partial_{n^{\pm}} \phi = \pm g(x), \tag{4.6}$$

with the additional constraint that the normal derivatives are related in the following manner

$$\partial_{n^-} = -\partial_{n^+}. \tag{4.7}$$

The notation ∂_n and $\partial_{n'}$ represents the normal derivatives on the boundary of our domain. If we lie on the outer boundary $\partial\Omega$, then the normal derivative is always outwards pointing relative to Ω, whereas if we lie on the inner boundary Γ, the normal derivative is always in the positive z direction.

As a consequence of the relationship (4.7), we can express everything in terms of Γ^+, and omit the notation in the future (in particular, $\partial_n = \partial_{n^+}$). Therefore, Green's second identity for our submerged structure, integrated along both intervals Γ^+ and Γ^-, becomes

$$\int_\Gamma (\phi \partial_{n'} G - G \partial_{n'} \phi) ds' = \int_\Gamma [\ \phi\ ] \partial_{n'} G ds', \tag{4.8}$$

where $[\ \phi\ ] = \phi(x^+) - \phi(x^-)$. This function represents the difference in potential above and below the beam, and is known as the jump in potential across Γ (Linton & McIver, 2001).

4.2 Integral equations for the submerged beam problem

In this section, we aim to formulate our problem in terms of integral equations. In doing this, we define our Green's function to be the fundamental solution of Laplace's equation in $\mathbb{R}^2$ (not the Green's function in the case of a floating plate). From Greenberg (1971),

$$G(\mathbf{x}, \mathbf{x}') = -\frac{1}{4\pi} \ln\left((x - x')^2 + (z - z')^2\right). \tag{4.9}$$

Employing Green's second identity, (2.33), around Ω and Γ, in connection with the well known result from Zwillinger (1992)

$$\int_\Omega \delta(\mathbf{x}, \mathbf{x}') \phi(\mathbf{x}') d\mathbf{s}' = \begin{cases} 0 & \text{if } \mathbf{x} \notin \Omega \text{ and } \mathbf{x} \notin \partial\Omega \\ \frac{1}{2}\phi(\mathbf{x}) & \text{if } \mathbf{x} \in \partial\Omega \\ \phi(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega \end{cases}, \tag{4.10}$$

we can obtain the following relationship for the total boundary of our domain (where $\mathbf{x}$ and $\mathbf{x}'$ are the source and field points in our region, restricted to lie on the boundary, with $\mathbf{n}$ and $\mathbf{n}'$ their respective normals)

$$\frac{1}{2}\phi(\mathbf{x}) = \int_\Gamma (\phi \partial_{n'} G - G \partial_{n'} \phi) ds' + \int_{\partial\Omega} (\phi \partial_{n'} G - G \partial_{n'} \phi) ds'. \tag{4.11}$$

Using our result from the previous section, we can express this as

$$\frac{1}{2}\phi(\mathbf{x}) = \int_\Gamma [\ \phi\ ] \partial_{n'} G ds' + \int_{\partial\Omega} (\phi \partial_{n'} G - G \partial_{n'} \phi) ds', \tag{4.12}$$

To evaluate the second integral in the above expression, we revisit boundary element methods.

4.3 Boundary element methods

As mentioned earlier, the boundary element method is a useful tool for solving elliptic problems such as Laplace's equation, in a closed region. Providing we correctly discretise our boundary into suitably small elements, we can express our integral as a sum over these elements, evaluating ϕ and ϕ_n at the panel midpoints.

For example,

$$\int_{\partial\Omega} (\phi \partial_{n'n} G - \partial_n G \partial_{n'} \phi) ds',$$

can be expressed in the form

$$\mathbb{G}^{(1)} \vec{\phi} - \mathbb{G}^{(2)} \vec{\phi}_{n'},$$

where

$$\mathbb{G}^{(1)}_{ij} = \int_{x_i - l/2}^{x_i + l/2} \partial_{n'n} G(x_i, x'_j) dh, \quad \text{and} \tag{4.13}$$

$$\mathbb{G}^{(2)}_{ij} = \int_{x_i - l/2}^{x_i + l/2} \partial_n G(x_i, x'_j) dh. \tag{4.14}$$

Recall that we have restricted $\mathbf{x}$ to be on Γ, and that $\mathbf{x}'$ is restricted to be on $\partial\Omega$. As a consequence of this, $\mathbb{G}^{(1)}$ and $\mathbb{G}^{(2)}$ are matrices of dimension $(\Gamma, \partial\Omega)$, and ϕ and $\phi_{n'}$ are vectors of dimension $(\partial\Omega, 1)$. We now consider in more detail, the computation of the fluid potential.

4.4 Radiation and diffraction potentials

Recall that the potential can be expressed as the sum

$$\phi = \phi^{\mathrm{D}} + \sum_m \widehat{\zeta}_m \phi^{\mathrm{R}}_m. \tag{4.15}$$

For our example of a submerged beam, we first consider computing the diffraction potential (with $\phi = \phi^{\mathrm{D}}$ to avoid notation), which is equivalent to solving the original system of equations ((4.1a) – (4.69d)), assuming that the structure is completely rigid:

$$\partial_n \phi = 0 \text{ on } \Gamma. \tag{4.16}$$

In order to compute $[\ \phi\ ]$, we will restrict all $\mathbf{x}$ points in (4.12) to be on Γ, and then take the normal derivative of that expression, to obtain:

$$⨍_{\Gamma} [\ \phi\ ] \partial_{n'n} G ds' + \int_{\partial\Omega} (\phi \partial_{n'n} G - \partial_n G \partial_{n'} \phi) ds' = 0. \tag{4.17}$$

where the first integral is known as a Hadamard finite part integral (discussed shortly). To avoid any possible confusion, it is important to note that for the first integral of (4.17), $(\mathbf{x}, \mathbf{x}') \in (\Gamma, \Gamma)$, and for the second integral, $(\mathbf{x}, \mathbf{x}') \in (\Gamma, \partial\Omega)$.

Note that when considering the radiated potentials ($\phi = \phi_m^{\mathrm{R}}$), the kinematic condition becomes

$$\partial_n \phi_m^{\mathrm{R}} = i\omega \widehat{X}_m. \tag{4.18}$$

However, for the moment, we will consider $\phi = \phi^{\mathrm{D}}$

4.5 Hypersingular integral equations

Hypersingular integral equations are integral equations whose kernel has a singularity of order greater than one. They typically arise in the study of problems in fluid dynamics.

We can obtain hypersingular integral equations from reducing Neumann boundary value problems for Laplace's equation to integral equations by means of the double-layer potential (Lifanov *et al.*, 2004). In other words, they effectively arise from deriving the log singularity of the Green's function twice.

We define a Cauchy principle value of a singular integral to be

$$\mathcal{P}\!\!\!\int_a^b f(x) dx = \lim_{\epsilon \to 0} \left[\int_a^{x_0 - \epsilon} f(x) dx + \int_{x_0 + \epsilon}^b f(x) dx \right], \quad \epsilon > 0,$$

providing the limit exists (Kwok, 2002). It can be shown that the derivative of a Cauchy principle value integral gives rise to a Hadamard principal value integral (Aliabadi, 2002)

$$\frac{\partial}{\partial x} \mathcal{P}\!\!\!\int_a^b \frac{f(x)}{x - x'} dx = ⨍_a^b \frac{f(x)}{(x - x')^2} dx.$$

Using the Green's function defined earlier:

$$G(\mathbf{x},\mathbf{x}') = -\frac{1}{4\pi}\ln\left((x-x')^2+(z-z')^2\right), \tag{4.19}$$

we can express the first term of equation (4.12) in the following form

$$\int_\Gamma [\ \phi\ ]\partial_{n'}G ds' = \frac{1}{2\pi}\int_\Gamma [\ \phi\ ]\frac{(z-z')}{(x-x')^2+(z-z')}ds', \tag{4.20}$$

which has an interior singularity in Γ. Consequently, this is a Cauchy principal value integral. Therefore, when we take the derivative of this, we obtain a Hadamard integral of the form

$$\frac{\partial}{\partial n}\int_\Gamma [\ \phi\ ]\partial_{n'}G ds' = \frac{1}{2\pi}\fint_\Gamma \frac{[\ \phi\ ]}{(x-x')^2}dx'. \tag{4.21}$$

In order to evaluate this integral, we express $[\ \phi(x')\ ]$ as a series of Chebyshev polynomials of the second kind

$$[\ \phi(x')\ ] = (1-x'^2)^{1/2}\sum_{m=0}^{M} b_m U_m(x'), \tag{4.22}$$

and use the result from Linton & McIver (2001)

$$\fint_{-1}^{1}\frac{(1-v^2)^{1/2}U_m(v)}{(u-v)^2}dv = -\pi(m+1)U_m(u). \tag{4.23}$$

From this we can obtain

$$-\frac{1}{2}\sum_{m=0}^{M} b_m(m+1)U_m(x) + \int_{\partial\Omega}\partial_{n'n}G\phi ds' - \int_{\partial\Omega}\partial_n G\partial_{n'}\phi ds' = 0, \tag{4.24}$$

providing that the non-dimensionalised length L is unitary.

Multiplying this expression by $(1-x^2)^{1/2}U_n(x)$ and integrating yields

$$\tilde{M}\vec{b} + \mathbb{Y}[\mathbb{G}^{(1)}\vec{\phi} - \mathbb{G}^{(2)}\vec{\phi}_{n'}] = 0, \tag{4.25}$$

where

$$\mathbb{Y} = \mathbb{U}^{\mathrm{T}}\mathbb{W}^{(1)}\mathbb{W}^{(2)}, \tag{4.26}$$

$$\mathbb{W}^{(1)} = \mathrm{diag}((1-x_k^2)^{1/2}), \tag{4.27}$$

$$\mathbb{U} = [U_1(x), U_2(x), \ldots, U_m(x)], \tag{4.28}$$

$\tilde{M}_{mm} = -\frac{\pi}{4}m$, and $\mathbb{W}^{(2)}$ is a weights matrix associated with the choice of a numerical integration technique. The matrices $\mathbb{G}^{(1)}$ and $\mathbb{G}^{(2)}$ denote the integration of our appropriate Green's function expressions, and the matrix $\mathbb{U}$ has columns of Chebyshev polynomials of the second kind.

This result arises through the use of the orthogonal relation (Mason, 2003)

$$\int_{-1}^{1} (1-x^2)^{1/2} U_m(x) U_n(x) dx = \frac{\pi}{2}\delta_{mn}. \tag{4.29}$$

We now focus our attention to the original expression (4.12)

$$\frac{1}{2}\phi(\mathbf{x}) = \int_{\Gamma} [\ \phi\ ] \partial_{n'} G ds' + \int_{\partial\Omega} (\phi \partial_{n'} G - G \partial_{n'} \phi) ds'. \tag{4.30}$$

Using the series approximation for $[\ \phi\ ]$ as at (4.22), we obtain

$$\frac{1}{2}\vec{\phi} = \mathbb{G}^{(3)}\mathbb{W}^{(1)}\mathbb{U}\vec{b} + \mathbb{G}^{(4)}\vec{\phi} - \mathbb{G}^{(5)}\vec{\phi}_{n'}, \tag{4.31}$$

where

$$\begin{aligned} \mathbb{G}_{ij}^{(3)} &= \int_{x_i - l/2}^{x_i + l/2} \partial_{n'} G(x_i, x_j') dh, & (4.32) \\ \mathbb{G}_{ij}^{(4)} &= \int_{x_i - l/2}^{x_i + l/2} \partial_{n'} G(x_i, x_j') dh, \quad \text{and} & (4.33) \\ \mathbb{G}_{ij}^{(5)} &= \int_{x_i - l/2}^{x_i + l/2} G(x_i, x_j') dh. & (4.34) \end{aligned}$$

Note that this time, $\mathbb{G}^{(3)}$ has $(\mathbf{x},\mathbf{x}') \in (\partial\Omega, \Gamma)$, whereas $\mathbb{G}^{(4)}$ and $\mathbb{G}^{(5)}$ are both restricted to $\partial\Omega$. Practically speaking, it is important to take note of line elements when integrating over $\partial\Omega$, and the direction of integration.

This leaves us with the following two equations

$$\begin{aligned} \tilde{M}\vec{b} + \mathbb{Y}\mathbb{G}^{(1)}\vec{\phi} - \mathbb{Y}\mathbb{G}^{(2)}\vec{\phi}_{n'} &= 0, \quad \text{and} & (4.35) \\ \mathbb{G}^{(3)}\mathbb{W}^{(1)}\mathbb{U}\vec{b} + \mathbb{G}^{(4)}\vec{\phi} - \mathbb{G}^{(5)}\vec{\phi}_{n'} &= \frac{1}{2}\vec{\phi}. & (4.36) \end{aligned}$$

In order to evaluate this, we need to find a way to express our boundary conditions in Neumann form, so that we can remove $\vec{\phi}_{n'}$ from the expressions above. This is discussed in the following section.

4.6 Neumann form

In order to utilise boundary element methods, we previously split our domain into three parts, and Dirichlet boundary conditions were imposed at the artificial boundaries of Ω. We now attempt to express these boundary conditions in Neumann form by examining this separation in more detail.

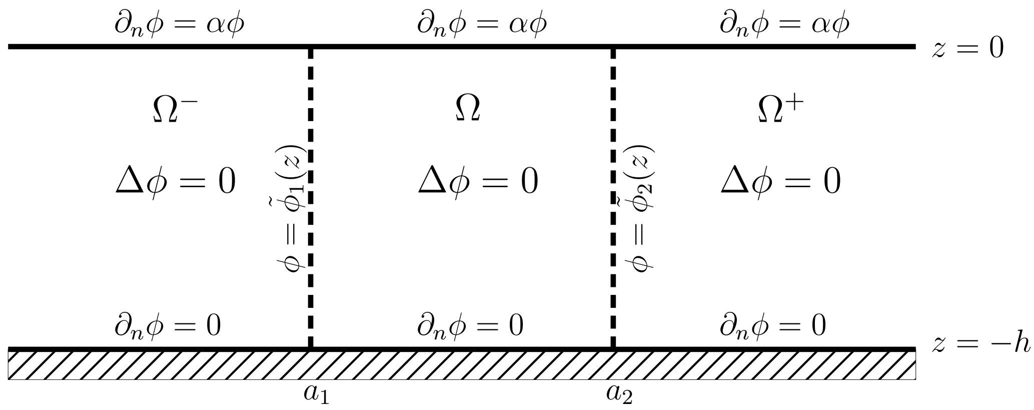

Figure 4.3: Semi-infinite domain split into 3 regions

In the Ω^- region ($-\infty < x < a_1$, $-h < z < 0$), the following equations model the potential of the fluid

$$\Delta\phi = 0,\ -h < z < 0, \tag{4.37a}$$

$$\partial_n\phi = \alpha\phi,\ z = 0, \tag{4.37b}$$

$$\partial_n\phi = 0,\ z = -h, \quad \text{and} \tag{4.37c}$$

$$\phi = \tilde{\phi}_1(z),\ x = a_1, \tag{4.37d}$$

plus a radiation condition

$$\lim_{x\to-\infty} \phi = \phi_0(z)e^{-k_0x} + R\phi_0(z)e^{k_0x}. \tag{4.38}$$

Note that we use the same expression for the incident potential as (2.20)

$$\phi^{\mathrm{I}} = \phi_0(z)e^{ikx} = \frac{\cosh(k(z+h))}{\cosh(kh)}e^{ikx} = \frac{\cos(k_0(z+h))}{\cos(k_0h)}e^{-k_0x}, \tag{4.39}$$

and that $a_1 < 0$.

For our problem in Ω^-, separation of variables can be used

$$\phi = \sum_{m=0}^{\infty} C_m X_m(x) Z_m(z), \tag{4.40}$$

with the following general solutions

$$\begin{aligned} Z_n(z) &= \frac{\cos(k_n(z+h))}{\cos(k_n h)}, \quad \text{and} & (4.41)\\ X_n(x) &= e^{k_n(x-a_1)}. & (4.42)\end{aligned}$$

A normalisation factor can be computed as follows

$$\int_{z=-h}^{0} Z_n(z) Z_m(z) dz = A_{mn}\delta_{mn}, \tag{4.43}$$

where

$$A_n = \frac{1}{2}\frac{\cos(k_n h)\sin(k_n h) + k_n h}{k_n \cos^2(k_n h)}. \tag{4.44}$$

Using condition (4.37), and orthogonality, allows us to determine that

$$C_m = \frac{1}{A_m}\int_{-h}^{0} \tilde{\phi}_1 Z_m(z) dz = \frac{1}{A_m}\left\langle \tilde{\phi}_1, Z_m(z)\right\rangle. \tag{4.45}$$

Employing the radiation condition finally yields the potential in the region Ω^-

$$\begin{aligned} \phi &= \phi_0(z) e^{-k_0 x} + \sum_{m=0}^{\infty} \frac{1}{A_m}\left\langle \tilde{\phi}_1, Z_m(z)\right\rangle X_m(x) Z_m(z) \\ &= \phi_0(z) e^{-k_0 x} + \sum_{m=0}^{\infty} \frac{1}{A_m}\left\langle \tilde{\phi}_1, \frac{\cos(k_n(z+h))}{\cos(k_n h)}\right\rangle \left[\frac{\cos(k_n(z+h))}{\cos(k_n h)}\right] e^{k_n(x-a_1)}. & (4.46)\end{aligned}$$

From this expression, we can form a Neumann boundary condition for our artificial boundary at $x = a_1$. For convenience, we define

$$\mathrm{Q_L}\tilde{\phi}_1(z) = -\sum_{m=0}^{\infty} \frac{k_m}{A_m}\left\langle \tilde{\phi}_1, Z_m(z)\right\rangle Z_m(z), \tag{4.47}$$

and evaluating (4.46) at $x = a_1$ yields the following

$$\phi\big|_{x=a_1} = \phi_0(z) e^{-a_1 k_0} + \sum_{m=0}^{\infty} C_m Z_m(z). \tag{4.48}$$

Using expressions (4.37) and (4.48), it can be shown that

$$\tilde{\phi}_1(z) = \phi_0(z)e^{-a_1 k_0} + \sum_{m=0}^{\infty} C_m Z_m(z). \tag{4.49}$$

To determine our Dirichlet boundary condition, we first multiply this expression by $Z_0(z)$ and integrate (then multiply the expression by $Z_n(z)$ and integrate), to obtain

$$\begin{aligned}
\left\langle \tilde{\phi}_1, Z_0 \right\rangle &= A_0 e^{-a_1 k_0} + A_0 C_0 \quad \text{(for } m = 0\text{)}, \quad \text{and} \\
\left\langle \tilde{\phi}_1, Z_n \right\rangle &= A_n C_n \qquad\qquad\quad \text{for } m \geq 1.
\end{aligned}$$

Therefore we can say that

$$\begin{aligned}
\partial_n \phi\big|_{x=a_1} &= -\partial_x \phi\big|_{x=a_1} \qquad (4.50) \\
&= k_0 \phi_0(z) e^{-k_0 a_1} - \sum_{m=0}^{\infty} k_m C_m Z_m(z) \\
&= k_0 \phi_0(z) e^{-k_0 a_1} - k_0 C_0 Z_0(z) - \sum_{m=1}^{\infty} k_m C_m Z_m(z) \\
&= k_0 \phi_0(z) e^{-k_0 a_1} - \left[\frac{k_0}{A_0} \left\langle \tilde{\phi}_1, Z_0 \right\rangle Z_0(z) - k_0 Z_0(z) e^{-k_0 a_1} \right] - \sum_{m=1}^{\infty} k_m C_m Z_m(z) \\
&= 2k_0 \phi_0(z) e^{-k_0 a_1} - \sum_{m=0}^{\infty} \frac{k_m}{A_m} \left\langle \tilde{\phi}_1, Z_m \right\rangle Z_m(z).
\end{aligned}$$

Consequently, the normal derivative at $x = a_1$ can be expressed as follows

$$\partial_n \phi(a_1, z) = \mathrm{Q_L} \tilde{\phi}_1(z) + 2k_0 \phi_0(z) e^{k_0 a_1}. \tag{4.51}$$

Observe that for our particular case, as the incident wave potential is unit amplitude in potential, we have $\phi_0(z) = Z_0(z)$. Using the approach above for the region Ω^+, with equations

$$\Delta\phi = 0, \ -h < z < 0, \tag{4.52a}$$

$$\partial_n \phi = \alpha\phi, \ z = 0, \tag{4.52b}$$

$$\partial_n \phi = 0, \ z = -h, \quad \text{and} \tag{4.52c}$$

$$\phi = \tilde{\phi}_2(z), \ x = a_2, \tag{4.52d}$$

plus a radiation condition

$$\lim_{x\to\infty} \phi = T\phi_0(z)e^{-k_0 x}, \tag{4.53}$$

it is straightforward to show that

$$\phi_n(a_2, z) = \mathrm{Q_R}\tilde{\phi}_2(z). \tag{4.54}$$

However, all this still leaves us with the unknown quantities, $\mathrm{Q_L}$ and $\mathrm{Q_R}$. Recall that we earlier defined

$$\mathrm{Q_L}\tilde{\phi}_1(z) = -\sum_{m=0}^{\infty} \frac{k_m}{A_m} \left\langle \tilde{\phi}_1, Z_m(z) \right\rangle Z_m(z), \qquad \text{at } x = a_1. \tag{4.55}$$

If we discretise the integral, assuming that $\tilde{\phi}_1(z)$ is piecewise constant over each element, it is clear to see that

$$\left\langle \tilde{\phi}_1, Z_m(z) \right\rangle = \int_{-h}^{0} \tilde{\phi}_1(p) Z_m(p) dp = \sum_i \tilde{\phi}_1(\mathbf{x}_i) \int_{\mathbf{x}_i - l/2}^{\mathbf{x}_i + l/2} Z_m(p) dp. \tag{4.56}$$

After truncating the number of modes, the following expression for $\mathrm{Q_L}$ can be obtained

$$\mathrm{Q_L}\tilde{\phi}_1(z) = -\sum_{m=0}^{M} \sum_i \left[\frac{1}{A_m} Z_m \right] \left[k_m \int_{\mathbf{x}_i - l/2}^{\mathbf{x}_i + l/2} Z_m(p) dp \right] \tilde{\phi}_1(\mathbf{x}_i), \tag{4.57}$$

which can be expressed in the form

$$\mathrm{Q_L}\tilde{\phi}_1 = \mathrm{SR}\tilde{\phi}_1, \tag{4.58}$$

where

$$s_{im} = -\frac{1}{A_m} \frac{\cos(k_m(\mathbf{x}_i + h)}{\cos(k_m h))}, \quad \text{and} \tag{4.59}$$

$$r_{mi} = k_m \int_{\mathbf{x}_i - l/2}^{\mathbf{x}_i + l/2} \frac{\cos(k_m(p+h))}{\cos(k_m h)} dp = \left[\frac{\sin(k_m(p+h))}{\cos(k_m h)} \right]_{p = \mathbf{x}_i - l/2}^{\mathbf{x}_i + l/2}. \tag{4.60}$$

We now have Neumann boundary conditions along our entire outer boundary. This allows us to express the boundary conditions as

$$\vec{\phi}_{n'} = \mathrm{A}\vec{\phi} - \vec{f}, \tag{4.61}$$

where

$$\mathrm{A} = \begin{bmatrix} \mathrm{Q_L} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mathrm{Q_R} & 0 \\ 0 & 0 & 0 & \alpha \mathrm{I} \end{bmatrix}, \quad \text{and} \quad \vec{f} = \begin{bmatrix} -2k_0\phi_0 e^{-k_0 a_1} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.62}$$

when orientating anticlockwise around the outer boundary from $(a_1, 0)$.

After converting all the boundary conditions to Neumann form, we can express our system (4.36) in the following form

$$\begin{bmatrix} \tilde{M} & \mathbb{Y}(\mathbb{G}^{(1)} - \mathbb{G}^{(2)}\mathrm{A}) \\ \mathbb{G}^{(3)}\mathbb{W}^{(1)}\mathbb{U} & (\mathbb{G}^{(4)} - \mathbb{G}^{(5)}\mathrm{A} - \frac{1}{2}\mathrm{I}) \end{bmatrix} \begin{bmatrix} \vec{b} \\ \vec{\phi} \end{bmatrix} = \begin{bmatrix} -\mathbb{Y}\mathbb{G}^{(2)}\vec{f} \\ -\mathbb{G}^{(5)}\vec{f} \end{bmatrix}. \tag{4.63}$$

Solving this allows us to recover the potential around the outer boundary, $\partial\Omega$, as well as $[\ \phi\ ]$ (or $[\ \phi^{\mathrm{D}}\ ]$ to be precise).

4.7 Solution for the beam

For the case of the radiated potentials ($\phi = \phi_m^{\mathrm{R}}$) we use an identical process, which yields the system of equations

$$\tilde{M}\vec{b} + \mathbb{Y}\mathbb{G}^{(1)}\vec{\phi} - \mathbb{Y}\mathbb{G}^{(2)}\vec{\phi}_{n'} = i\omega\mathbb{Y}\mathbf{A}X_{mat}, \quad \text{and} \tag{4.64}$$

$$\mathbb{G}^{(3)}\mathbb{W}^{(1)}\mathbb{U}\vec{b} + \mathbb{G}^{(4)}\vec{\phi} - \mathbb{G}^{(5)}\vec{\phi}_{n'} = \frac{1}{2}\vec{\phi}, \tag{4.65}$$

(for all m). This system can be expressed in the following form

$$\begin{bmatrix} \tilde{M} & \mathbb{Y}(\mathbb{G}^{(1)} - \mathbb{G}^{(2)}\mathrm{A}) \\ \mathbb{G}^{(3)}\mathbb{W}^{(1)}\mathbb{U} & (\mathbb{G}^{(4)} - \mathbb{G}^{(5)}\mathrm{A} - \frac{1}{2}\mathrm{I}) \end{bmatrix} \begin{bmatrix} \vec{b} \\ \vec{\phi} \end{bmatrix} = \begin{bmatrix} i\omega\mathbb{Y}\mathbf{A}X_{mat} \\ 0 \end{bmatrix}. \tag{4.66}$$

Solving this system gives us the critical $[\ \phi_m^{\mathrm{R}}\ ]$. Having found $[\ \phi\ ]$ for both the radiated and diffraction potentials, the displacement coefficients for our submerged beam can now be determined, providing we solve the dynamic condition

$$\sum_{n=0}^{N} \left(\widehat{\mu}_n^4 \gamma(x) - \gamma(x)\alpha\right) \zeta_n \widehat{X}_n = -i\omega \left([\ \phi^{\mathrm{D}}\ ] + \sum_{n=0}^{N} \zeta_n [\ \phi_n^{\mathrm{R}}\ ] \right), \quad x \in (-L, L), \quad z = -d. \tag{4.67}$$

After multiplying by $\widehat{X}_n$ and integrating over the submerged plate, we obtain

$$\sum_{n=0}^{N}\left(\widehat{\mu}_n^4-\alpha\right)\left[\int_{-L}^{L}\widehat{X}_n\widehat{X}_m dx+i\omega\int_{-L}^{L}[\ \phi_n^{\mathrm{R}}\ ]\widehat{X}_m dx\right]\zeta_n=-i\omega\int_{-L}^{L}[\ \phi^{\mathrm{D}}\ ]\widehat{X}_m dx.$$

This can be expressed in the following matrix form

$$\left[(\widehat{\mathbf{D}}-\alpha\mathbf{I})+\mathbf{A^T A}+\mathbf{A^T \Lambda A}\right]\vec{\zeta}=\mathbf{A}\vec{\mathbf{f}}, \tag{4.68}$$

where the matrices and vectors are defined analogously to the case of a submerged beam. This system is closely related to (3.26).

Unfortunately we do not have the numerical solution for the elastic plate, so for the purposes of this dissertation, we only consider the case of wave scattering from a rigid beam. This is equivalent to only considering the diffraction potential for our elastic beam (and ignoring the radiated potentials entirely).

In doing this, we can use the method of eigenfunction matching to determine whether our results are consistent with established theory.

4.8 Eigenfunction matching for a submerged rigid body

We use the established technique of eigenfunction matching, as discussed by Heins (1950), to determine whether our integral equation formulation and use of boundary element methods is producing consistent results with the calculation of the diffraction potential. For the purposes of the eigenfunction matching, we model our problem in the frequency domain, as previously

$$\Delta\phi = 0, \ -h < z < 0, \tag{4.69a}$$

$$\partial_z\phi = 0, \ z = -h, \tag{4.69b}$$

$$\partial_z\phi = \alpha\phi, \ z = 0, \quad \text{and} \tag{4.69c}$$

$$\partial_z\phi = 0, \ z = -d, \ x \in (-L, L), \tag{4.69d}$$

plus the Sommerfeld radiation condition

$$\frac{\partial}{\partial x}\phi \pm ik\phi = 0, \ \text{as } x \to \pm\infty. \tag{4.70}$$

We have 4 primary regions, $x < -L$, $x > L$, and $x \in (-L, L)$ for $z \in (-d, 0)$ and $z \in (-h, -d)$ (referred to as regions 1 to 4 respectively). In the first three regions, we use the free-surface eigenfunctions (due to their contact with the free surface itself), and in the last, we use the structure eigenfunctions.

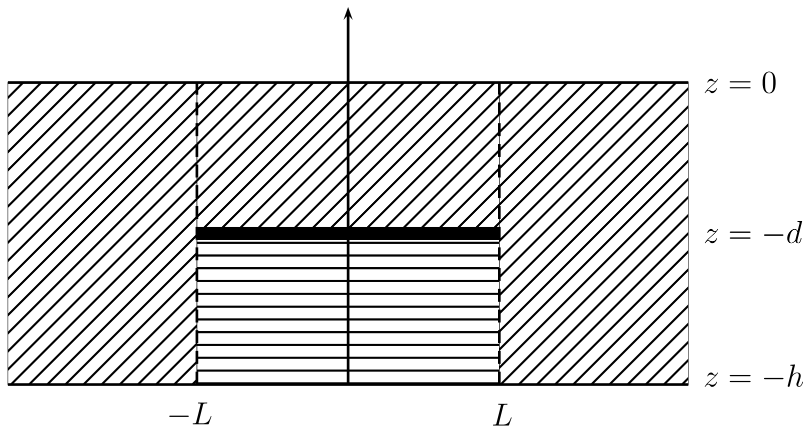

Figure 4.4: Regions for eigenfunction matching method

In section (4.6) earlier, the free surface eigenfunctions were derived in the case of depth h

$$Z_n(z) = \frac{\cos(k_n^h(z+h))}{\cos(k_n^h h)}, \tag{4.71}$$

where the new notation k_n^h denotes the roots of the dispersion relation for depth h.

Separation of variables for a floating structure yields the eigenfunctions

$$Z_n(z) = \cos(\kappa_n(z+h)), \tag{4.72}$$

where $\kappa_n = \frac{n\pi}{h}$. Adjusting for the fact the beam is submerged gives rise to $\kappa_n = \frac{n\pi}{h-d}$ (the eigenfunction is unchanged). To avoid confusion, we define

$$\phi_m^h = \frac{\cos(k_n^h(z+h))}{\cos(k_n^h h)}, \tag{4.73}$$

$$\phi_m^d = \frac{\cos(k_n^d(z+d))}{\cos(k_n^d d)}, \quad \text{and} \tag{4.74}$$

$$\psi_m = \cos(\kappa_m(z+h)), \tag{4.75}$$

then expand the potential as follows

$$\phi_1(x,z) = e^{-k_0^h(x+L)}\phi_0^h(z) + \sum_{m=0}^{\infty} a_m e^{k_m^h(x+L)}\phi_m^h(z), \tag{4.76a}$$

$$\phi_2(x,z) = \sum_{m=0}^{\infty} b_m e^{-k_m^d(x+L)}\phi_m^d(z) + \sum_{m=0}^{\infty} c_m e^{k_m^d(x-L)}\phi_m^d(z), \tag{4.76b}$$

$$\phi_3(x,z) = d_0\frac{L-x}{2L} + \sum_{m=1}^{\infty} d_m e^{-\kappa_m(x+L)}\psi_m(z) + e_0\frac{x+L}{2L} + \sum_{m=1}^{\infty} e_m e^{\kappa_m(x-L)}\psi_m(z), \quad \text{and} \tag{4.76c}$$

$$\phi_4(x,z) = \sum_{m=0}^{\infty} f_m e^{-k_m^h(x-L)}\phi_m^h(z), \tag{4.76d}$$

where ϕ_1 denotes the potential in the first region, and so forth. The potential and its derivative must be continuous between the regions, that is, the potentials and their derivatives must match at $x = \pm L$:

$$\phi_1 = \phi_2 + \phi_3, \qquad \text{at } x = -L, \quad \text{and} \tag{4.77}$$

$$\phi_4 = \phi_2 + \phi_3, \qquad \text{at } x = L, \tag{4.78}$$

plus their respective derivatives. Multiplying these equations by ϕ_l^h, and integrating from $-h$ to 0, allows us to find our required coefficients, when the sums are suitably truncated.

4.9 Numerical results

Using MATLAB, we are able to reconstruct $[\ \phi\ ]$, and the potential around the boundary for a rigid structure, for an incident wave of a given frequency. The solution obtained from the method outlined in this dissertation is compared with the solution from eigenfunction matching.

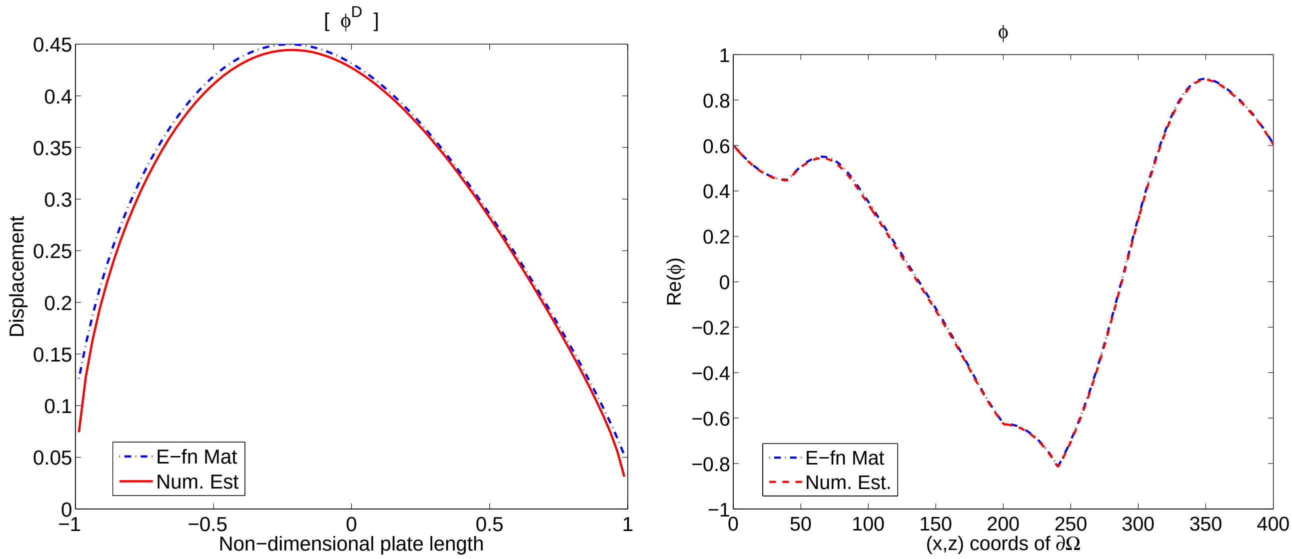

Figure 4.5: $[\ \phi\ ]$ and ϕ for $\alpha = 0.5$

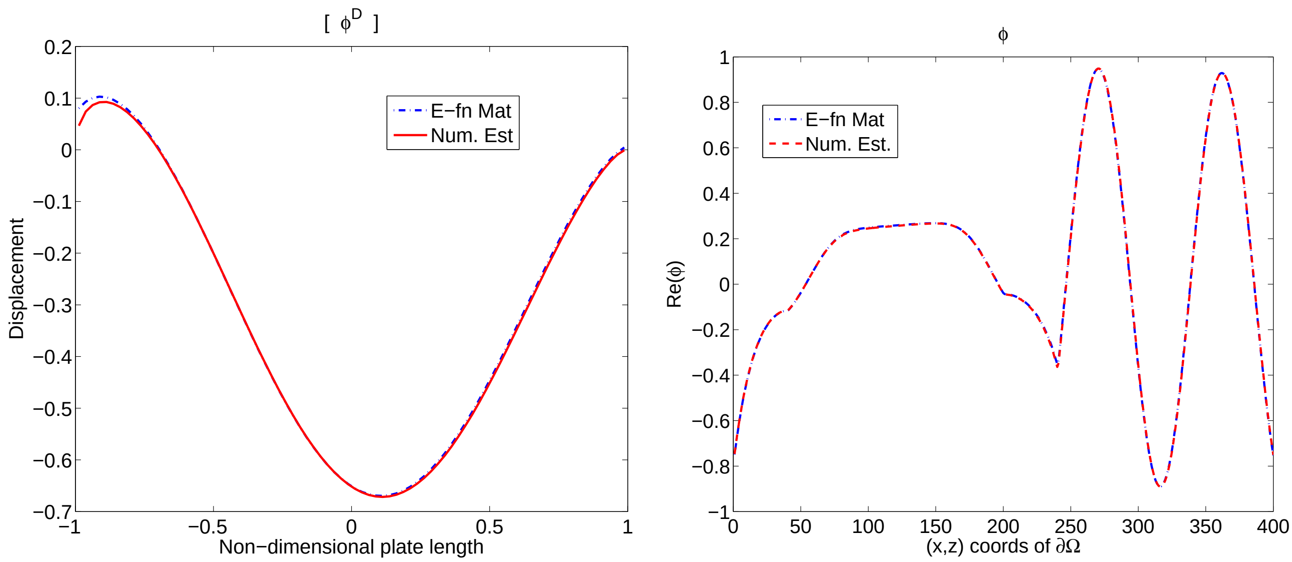

Figure 4.6: $[\ \phi\ ]$ and ϕ for $\alpha = 2.5$

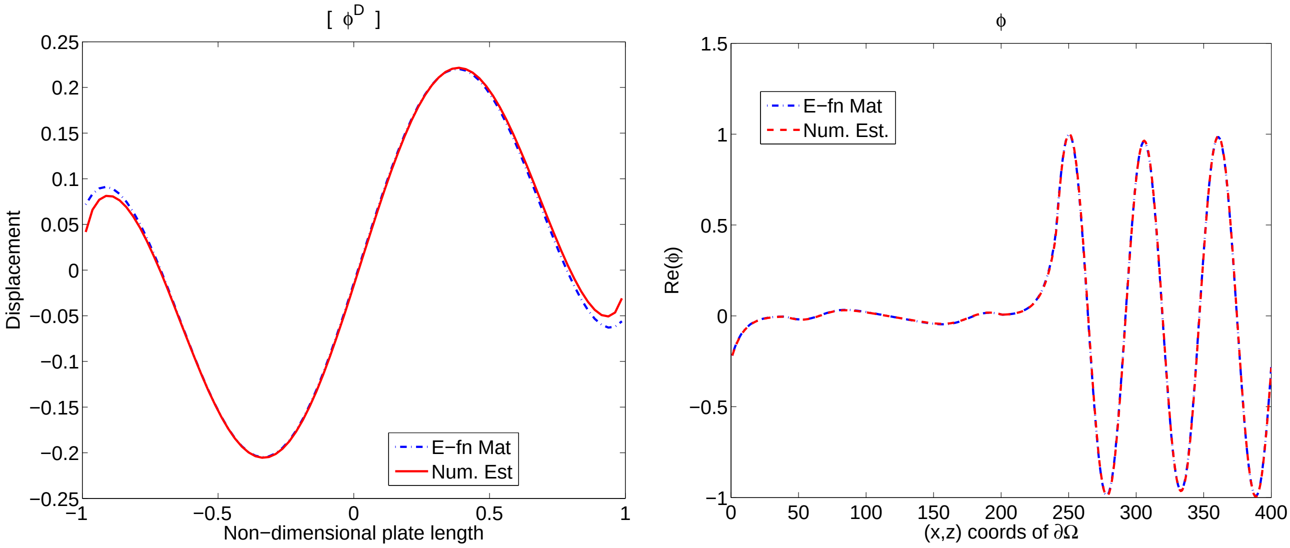

Figure 4.7: $[\ \phi\ ]$ and ϕ for $\alpha = 4.5$

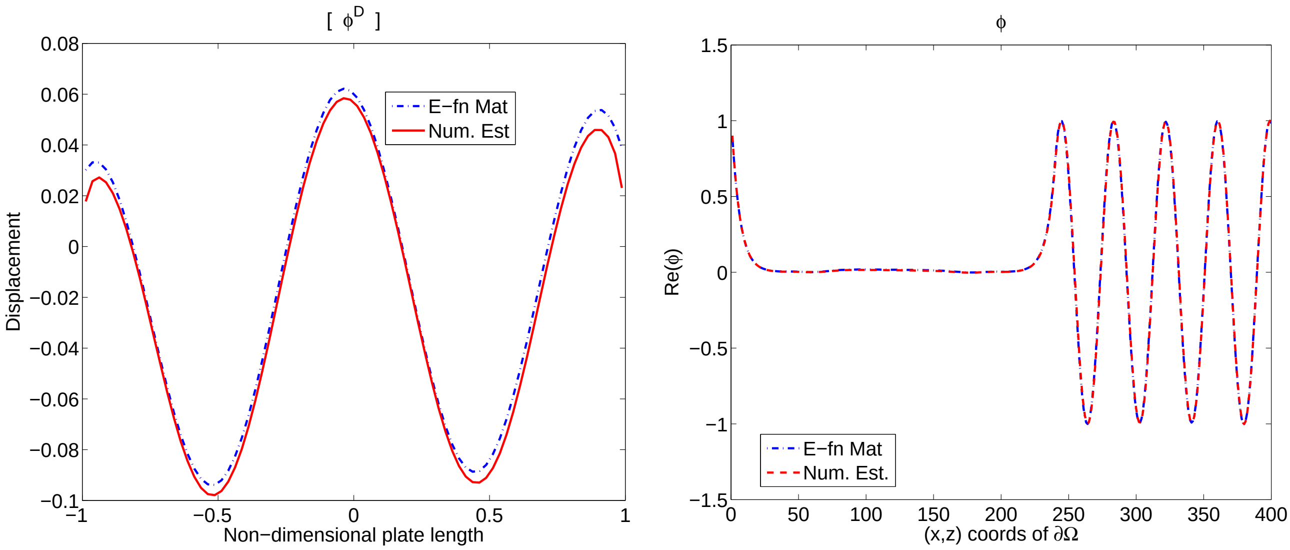

Figure 4.8: $[\ \phi\ ]$ and ϕ for $\alpha = 6.5$

The examples shown above demonstrate good agreement for reconstructing the potential at the boundary, and also show that our solution is well matched at the plate.

5
Discussion

This dissertation examines three problems, specifically, the displacement of a uniform floating beam, the displacement of a variable floating beam, and the response from a submerged plate.

The first part relating to uniform beam theory was an implementation of an established solution by Newman (1994). However, the solution method for variable floating plates is original research. The solution method for variable plates involves an expansion of the non-uniform modes of vibration in terms of the uniform modes using Rayleigh-Ritz methods, so as to determine the displacement of the beam. Numerical methods have been able to produce results consistent with the solution for uniform beams.

A solution method is also presented for the problem of a submerged plate. The solution method for this problem is also similar to the technique for floating beams, involving the separation of the potential. This is also original work, as previous authors (Yang, 2002), have only considered wave scattering from submerged rigid structures.

The solution method for submerged beams had the additional complication of hyper-singular integral equations. This problem was overcome by expanding an unknown in terms of Chebyshev polynomials of the second kind.

Theoretically, a full solution method is shown for the case of an submerged variable beam, however, we have only been able to numerically demonstrate the theory for the case of a rigid body. These results were compared against the established approach of eigenfunction matching, and results were found to be consistent with this.

Details of this dissertation, and MATLAB code, can be found online at `www.wikiwaves.org`, in particular

- `http:// ... /Boundary_Element_Method_for_a_Fixed_Body_in_Finite_Depth`
- `http:// ... /Eigenfunction_Matching_for_a_Submerged_Finite_Dock`
- `http:// ... /Free-Surface_Green_Function`
- `http:// ... /Green_Function_Methods_for_Floating_Elastic_Plates`
- `http:// ... /Wave_Scattering_by_Submerged_Thin_Bodies`
- `http:// ... /Waves_on_a_Variable_Beam`

Bibliography

ALIABADI, M. H., ed. 2002 *The Boundary Element Method Volume II: Applications in Solids and Structures*. New York: Wiley.

BENNETTS, L., BIGGS, N. & PORTER, D. 2007 A multi-mode approximation to wave scattering by ice sheets of varying thickness. *Journal of Fluid Mechanics* **579**, 413–443.

CARTWRIGHT, D. J., ed. 2001 *Underlying Principles of the Boundary Element Method*. Southampton: WIT Press.

CHAPRA, S. C., ed. 2008 *Applied numerical methods with MATLAB for engineers and scientists*. Boston: McGraw-Hill.

FOX, C. & SQUIRE, V. A. 1994 On the oblique reflexion and transmission of ocean waves at shore fast sea ice. *Phil. Trans. R. Soc. Lond. A.* **347**, 185–218.

GREENBERG, M. D., ed. 1971 *Applications of Green's Functions in Science and Engineering*. Englewood Cliffs, NJ: Prentice-Hall.

HEINS, A. E. 1950 Water waves over a channel of finite depth with a submerged plane barrier. *Can. J. Math.* **2**, 210–222.

KWOK, Y. K., ed. 2002 *Applied Complex Variables for Scientists and Engineers*. New York: Cambridge University Press.

KYOUNG, J. H., HONG, S. Y., KIM, B. W. & CHO, S. K. 2005 Hydroelastic response of a very large floating structure over a variable bottom topography. *Ocean Engineering* **32**, 2040–2052.

LANCZOS, C., ed. 1949 *The Variational Principles of Mechanics*. Toronto: University of Toronto Press.

LIFANOV, I. K., POLTAVSKII, L. N. & VAINIKKO, G., ed. 2004 *Hypersingular integral equations and their applications*. Boca Raton, FL: Chapman & Hall.

LINTON, C. M. & MCIVER, P., ed. 2001 *Handbook of Mathematical Techniques for Wave/Structure Interactions*. Boca Raton, FL: Chapman and Hall.

MASON, J. C., ed. 2003 *Chebyshev Polynomials*. Boca Raton, FL: Chapman & Hall.

MEI, C. C. 1989 *The Applied Dynamics of Ocean Surface Waves*. World Scientific.

MEYLAN, M. & STUROVA, I. 2009 Time-dependent motion of a two-dimensional floating elastic plate. *Journal of Fluid Mechanics* **25**, 445–460.

MEYLAN, M. H. & SMITH, M. J. 2009 Wave scattering by a floating elastic plate of variable stiffness and mass. *In Revision.*

MEYLAN, M. H. & SQUIRE, V. A. 1994 The response of ice floes to ocean waves. *J. Geophys. Research* **99** (C1), 891–900.

NEWMAN, J. N. 1977 *Marine hydrodynamics*. MIT Press.

NEWMAN, J. N. 1994 Wave effects on deformable bodies. *Appl. Ocean Res.* **16**, 45 – 101.

RAO, S., ed. 1986 *Mechanical Vibrations*. Reading, MA: Addison-Wesley.

SQUIRE, V. A. 2007 Of ocean waves and sea-ice revisited. *Cold Regions Science and Technology* **49** (2), 110–133.

STOKER, J. J. 1957 *Water Waves: The Mathematical Theory with Applications*. Interscience, New York.

TAYLER, A. B., ed. 1986 *Mathematical Models in Applied Mechanics*. Oxford: Clarendon Press.

WATANABE, E., UTSUNOMIYA, T. & WANG, C. M. 2004 Hydroelastic analysis of pontoon-type VLFS: a literature survey. *Eng. Struct.* **26** (2), 245–256.

YANG, S. 2002 A numerical method for scattering from acoustically soft and hard thin bodies in two dimensions. *J. of Fluid Mechanics.* **250** (5), 773–793.

ZWILLINGER, D., ed. 1992 *Handbook of Differential Equations*. Boston: Academic Press.

6
Appendix

Details of the MATLAB code produced over the course of the dissertation are included here for reference. Some of the code produced relies on programs that can be found on wikiwaves (at the pages listed in the discussion section).

6.1 elastic_plate_modes_variable.m

```
function [zeta,displacement,wavepotential,R,T]= ...
    elastic_plate_modes_variable(alpha,beta,gamma,h,L,number_modes,N)

%[zeta,displacement,wavepotential,R,T]= ...
%elastic_plate_modes_variable(alpha,beta,gamma,h,L,number_modes,N)
%
% Program to determine the modes of a floating nonuniform beam.
% This function file determines the following:
%
%            zeta = coefficients of displacement
%            displacement = complex displacement
%            wavepotential = complex potential
%            R = reflection coefficient
%            T = transmission coefficient
%
%and requires:
%
%            gamma=non dimensionalised m (linear mass density function)
%            beta= non dimensionalied D (flexural rigidity function)
%            alpha=frequency of the incident wave
%            h=depth of water
%            L=half the length of the beam
%            number_modes=number of modes
%            N=number of points
%
%defaults listed here:
% if nargin ==0, L=1/2;   end
% if nargin <=1, N = 200; end
% if nargin <=2, beta=ones(size(x)); end
% if nargin <=3, gamma=ones(size(x)); end
% if nargin <=4, alpha=1;  end
% if nargin <=5, h=1;  end
% if nargin <=6, number_modes=10;  end

%------ ------ ------ ------ ------ ------ ------ ------ ------ ------
```

```
%computing non uniform eigenvalues and eigenvectors
N_beam = (length(beta) - 1)/2;
x = linspace(-L,L,2*N_beam+1);

eigvals = beam_ev(number_modes,L);    %vector of eigenvalues for uniform
eigmodes=beam_em(eigvals,L,x);         %matrix of uniform eigenfunctions
eigmodes_dd=beam_em_dd(eigvals,L,x);  %matrix of second derivatives

weights=2*ones(1,length(x));           %vector of weights
weights(2:2:2*N_beam) = 4;weights(1) = 1;weights(end) = 1;
weights = L/(3*N_beam)*weights;

J_mat=diag(gamma.*weights);
M=eigmodes*J_mat*eigmodes';

H_mat=diag(beta.*weights);
K=eigmodes_dd*H_mat*eigmodes_dd';
[A,D] = eig(K,M);  %D contains the non-uniform eigvals

%------ ------ ------ ------ ------ ------ ------ ------ ------ ------
%Moving onto potential problem (different number of pts optional)

x=linspace(-L,L,2*N+1);
eigmodes=beam_em(eigvals,L,x);

weights=2*ones(1,length(x));              %vector of weights
weights(2:2:2*N) = 4;weights(1) = 1;weights(end) = 1;
weights = L/(3*N)*weights;


G = matrix_G_surface(alpha,h,L,N);       %free-surface Green function
k = dispersion_free_surface(alpha,0,h); %solves alpha=-k*tan(k*h)s

phi_i = exp(-k*x);                        %unit amplitude in potential
phi_d = inv(eye(length(G))-alpha*G)*phi_i.';
phi_r = inv(eye(size(G))-alpha*G)*G*(-1i*sqrt(alpha))*eigmodes.';

```

```
ab = transpose(eigmodes*diag(weights)*phi_r);
diff = eigmodes*diag(weights)*phi_d;

zeta_hat = (D*transpose(A)*M*A - alpha*transpose(A)*M*A +  ...
    transpose(A)*A + i*sqrt(alpha)*transpose(A)*ab*A)\ ...
    (-i*sqrt(alpha)*transpose(A)*diff);

zeta = A*zeta_hat;  %expressing in terms of uniform problem

displacement = transpose(zeta)*eigmodes;
wavepotential = transpose(zeta)*transpose(phi_r) + transpose(phi_d);

%reflection/transmission coefficients
R = -alpha/(tan(k*h)+k*h*sec(k*h)^2)*((exp(-k*x)).* ...
    (wavepotential - 1i/sqrt(alpha)*displacement))*weights';
T = 1 - alpha/(tan(k*h) + k*h*sec(k*h)^2)*((exp(k*x)).* ...
    (wavepotential - 1i/sqrt(alpha)*displacement))*weights';
%Note this program is based on existing code from wikiwaves.org
```

6.2 submerged_beam_code.m

```
%This file computes the jump in potential for a submerged structure,
%comparing the solution with eigenfunction matching
clear all
close all
% %%%%alter these three to test%%%%%%%%%%%
alpha=0.5;
h=1;
h1=0.5;
stepsize=0.025;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N_Q=10; %number of modes used to compute Q matrix
N=10;   %number of modes for Dr Mike Meylan's code
L=1;
[phi_eig,phin_eig,phij_eig]=eigmatch(alpha,h,h1,stepsize,N);
kh = dispersion_free_surface(alpha,N,h);% roots of the dispersion equation
k0h=kh(1);
kd = dispersion_free_surface(alpha,N,h1);
%%%for exterior boundary%%%
[outerboundary_panels,count] = outerpanels(stepsize,h,-2,2);
xmidpts=(outerboundary_panels(:,1)+outerboundary_panels(:,3))/2;
zmidpts=(outerboundary_panels(:,2)+outerboundary_panels(:,4))/2;
dOmega_midpts=cat(2,xmidpts,zmidpts);
%need these thresholds so sizes match correctly
count2a=count(1)+(count(2)-count(1))/4;
count2b=count(1)+(count(2)-count(1))*3/4;
count3a=count(3)+(count(4)-count(3))/4;
count3b=count(3)+(count(4)-count(3))*3/4;
%%%for the plate%%%
gamma_x=linspace(-L,L,count3b-count3a+1);
gamma_z=-h1*ones(size(gamma_x));
xmidpts=gamma_x(1:end-1)+diff(gamma_x)/2;
zmidpts=-h1*ones(size(xmidpts));
```

```
gamma_midpts=cat(1,xmidpts,zmidpts);gamma_midpts=transpose(gamma_midpts);
boundary=cat(2,gamma_x',gamma_z');
gamma_panels=cat(2,boundary(1:(end-1),:),boundary(2:end,:));

%%
%this section computes the Q matrix
panelsL = outerboundary_panels(1:count(1),:);
[Q_left,F_left]=Q_matrix(panelsL,alpha,N_Q);
Q_right=flipud(fliplr(Q_left)); %due to symmetry of rectangular domain

A = blkdiag(Q_left,zeros(count(2)-count(1)), ...
    Q_right,alpha*eye(count(4)-count(3)));
F = exp(-k0h)*[F_left.',zeros(length(phi_eig)-length(F_left),1).']; F=F(:);

%%
x=gamma_midpts(:,1);
z=gamma_midpts(:,2);

matrix_G1 = G1(outerboundary_panels,gamma_midpts);
matrix_G2 = G2(outerboundary_panels,gamma_midpts,count);
matrix_G3 = G3(dOmega_midpts,gamma_panels);
[matrix_G5,matrix_G4]=bem_constant_panel(outerboundary_panels);

% testing Green's 2nd identity expression
% phi_test1=2*(matrix_G3*phij_eig+(matrix_G4*phi_eig-matrix_G5*phin_eig));
% plot(real(phi_test1),'bo')
% hold on
% plot(real(phi_eig),'.r')
% hold off

U = chebyshevII(10,x);
W1=diag((1-x.^2).^(1/2));
W2 = abs(x(1)-x(2)).*eye(size(W1));
Y = U.'*W1* W2;
M_tilde=-pi/4*diag(1:1:10);

% test code for chebyshev coefficients, original integral eqn,
```

```
% and its derivative:
%
% b_eig=2/pi*abs(x(1)-x(2)).*U.'*phij_eig;
% %this should give zeros:
% roughly_zeros = M_tilde*b + Y*(matrix_G1*phi_eig - matrix_G2*phin_eig);
% %this should give close estimate to phi_eig:
% phi_test2 = 2*(matrix_G3*W1*U*b+(matrix_G4*phi_eig-matrix_G5*phin_eig));
% plot(real(phi_eig),'b');
% hold on
% plot(real(phi_test2),'go');
% hold off
% figure
% plot(real(roughly_zeros))

%defining block matrices and solving
block_matrix = vertcat([M_tilde, Y*(matrix_G1 - matrix_G2 *A)], ...
    [matrix_G3*W1*U, matrix_G4 - matrix_G5*A - 1/2*eye(size(A))]);
block_vector = vertcat(-Y*matrix_G2*F, -matrix_G5 *F);
block_solution = block_matrix\block_vector;
b_coeffs=block_solution(1:10);  %these are tested against b_eig
phi = block_solution(11:end);   %potential around the outer boundary
phij=W1*U*b_coeffs;             %this is the jump in potential

```

6.3 eigmatch.m

```

function [phi,phin,phij]=eigmatch(alpha,h,h1,stepsize,N)
%function [phi,phin,phij]=eigmatch(alpha,h,h1,stepsize,N)
%This function file uses Dr Mike Meylan's existing code,
%submerged_finite_dock.m, to obtain the coefficients of the potential
%using eigenfunction matching.  Using this code, an expression for the
%potential around the boundary, and the jump in potential at the submerged
%plate, can be obtained ( the normal derivative of the potential is also
%computed).
%
%Input Parameters:  alpha    -- frequency of incoming wave
%                   h        -- constant depth of fluid medium
%                   h1       -- depth plate submerged at
%                   stepsize --
%                   N        -- number of modes
%
%This code has been tailored to compute the potential for a rectangular
%region [-2 2] x [0 -h]. For the purposes of our program, the submerged
%beam is assumed to have (non-dimensionalised) length [-1,1].
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%         (8)                 (7)                 (6)
%-----.-------------------------------------------------.-------
%     |            |                       |            |           z=0
%     |            |                       |            |
%     |            |                       |            |
%  (1)|            |________________|             | (5)
%     |            |                       |            |
%     |            |                       |            |
%     |            |                       |            |           z=-h
%-----.-------------------------------------------------.--------
%    -2   (2)     -1          (3)          1      (4)   2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% an outline of this method can be found at www.wikiwaves.org
```

```
%http:// ... /Eigenfunction_Matching_for_a_Submerged_Finite_Dock

L=1;theta=0;    %theta denotes the angle the incident wave hits the beam
[a,b,c,d,e,f] = submerged_finite_dock(alpha,h,h1,L,theta,N);

% a consequence of using submerged_finite_dock.m is introducing variables:
N_above = round(h1/h*(N+1)) -1; % number of modes above is N_above + 1;
N_below = N-1-N_above; % number of modes below is N_below +1;
if N_above <  0
    N_above = 0;
    N_below = N-1;
elseif N_below < 0
    N_above = N-1;
    N_below = 0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% roots of the dispersion equation for the water
kh = dispersion_free_surface(alpha,N,h);
kd = dispersion_free_surface(alpha,N,h1);
kappa = [0:N_below]*pi/(h-h1);

%%%for exterior boundary%%%
[outerboundary_panels,count] = outerpanels(stepsize,h,-2,2);
xmidpts=(outerboundary_panels(:,1)+outerboundary_panels(:,3))/2;
zmidpts=(outerboundary_panels(:,2)+outerboundary_panels(:,4))/2;
dOmega_midpts=cat(2,xmidpts,zmidpts);

%need these thresholds so everything all the right size
count2a=count(1)+(count(2)-count(1))/4;
count2b=count(1)+(count(2)-count(1))*3/4;
count3a=count(3)+(count(4)-count(3))/4;
count3b=count(3)+(count(4)-count(3))*3/4;
%%%for the plate%%%
gamma_x=linspace(-L,L,count3b-count3a+1);
gamma_z=-h1*ones(size(gamma_x));
xmidpts=gamma_x(1:end-1)+diff(gamma_x)/2;
```

```
zmidpts=-h1*ones(size(xmidpts));
gamma_midpts=cat(1,xmidpts,zmidpts);gamma_midpts=transpose(gamma_midpts);
boundary=cat(2,gamma_x',gamma_z');
gamma_panels=cat(2,boundary(1:(end-1),:),boundary(2:end,:));
%%
%for side (1)
k0h=kh(1);
x=dOmega_midpts(1:count(1),1);
z=dOmega_midpts(1:count(1),2);
phi0 = cos(kh(1)*(z+h))/cos(kh(1)*h);
for i=1:length(a)
    out(i,:)  = exp(kh(i)*(x+L)).*(cos(kh(i)*(z+h))/cos(kh(i)*h));
    out1(i,:) = kh(i)*exp(kh(i)*(x+L)).*(cos(kh(i)*(z+h))/cos(kh(i)*h));
end
phi1=(phi0.*exp(-kh(1)*(x+L))).'+a.'*out;
phi1n = (k0h*phi0.*exp(-kh(1)*(x+L))).'-a.'*out1;
clear out out1
%%
%for side (2)
x=dOmega_midpts(count(1)+1:count2a,1);
z=dOmega_midpts(count(1)+1:count2a,2);
for i=1:length(a)
    out(i,:) = exp(kh(i)*(x+L)).*cos(kh(i)*(z+h))/cos(kh(i)*h);
    out1(i,:) = exp(kh(i)*(x+L)).*( kh(i)*sin(kh(i)*(z+h)) )/cos(kh(i)*h);
end
phi2 = (exp(-k0h*(x+L)).*cos(k0h*(z+h))/cos(k0h*h)).'+a.'*out;
phi2n = (exp(-k0h*(x+L)).*( k0h*sin(k0h*(z+h))   )/cos(k0h*h)).'+a.'*out1;
clear out out1
%%
%for side (3)
```

```
x=dOmega_midpts(count2a+1:count2b,1);
z=dOmega_midpts(count2a+1:count2b,2);

for i=2:length(d)
    out1(i-1,:) = exp(-kappa(i)*(x+L)).*cos(kappa(i)*(z+h));
    out3(i-1,:) = exp(-kappa(i)*(x+L)).*(kappa(i)*sin(kappa(i)*(z+h)));
end

for i=2:length(e)
    out2(i-1,:) = exp(kappa(i)*(x-L)).*cos(kappa(i)*(z+h));
    out4(i-1,:) = exp(kappa(i)*(x-L)).*(kappa(i)*sin(kappa(i)*(z+h)));
end

if length(d)==1
    temp1=zeros(length(x),1);
    temp3=zeros(length(x),1);
else
    temp1=d(2:end).'*out1;
    temp3=d(2:end).'*out3;
end

if length(e)==1
    temp2=zeros(length(x),1);
    temp4=zeros(length(x),1);
else
    temp2=e(2:end).'*out2;
    temp4=e(2:end).'*out4;
end
temp1=temp1(:).';  % all row vectors
temp2=temp2(:).';
temp3=temp3(:).';
temp4=temp4(:).';

phi3 = (d(1)*(L-x)/(2*L)).'+ temp1  +(e(1)*(x+L)/(2*L)).' + temp2;
phi3n = temp3  +temp4 ;
```

```
clear out1 out2 out3 out4 temp1 temp2 temp3 temp4
%%
%for side (4)
x=dOmega_midpts(count2b+1:count(2),1);
z=dOmega_midpts(count2b+1:count(2),2);

for i=1:length(f)
    out(i,:) = exp(-kh(i)*(x-L)).*cos(kh(i)*(z+h))/cos(kh(i)*h);
    out1(i,:) = exp(-kh(i)*(x-L)).*(kh(i)*sin(kh(i)*(z+h))/cos(kh(i)*h));
end

phi4=f.'*out;
phi4n=f.'*out1;
clear out out1
%%
%for side (5)
x=dOmega_midpts(count(2)+1:count(3),1);
z=dOmega_midpts(count(2)+1:count(3),2);

for i=1:length(f)
    out(i,:) = exp(-kh(i)*(x-L)).*cos(kh(i)*(z+h))/cos(kh(i)*h);
    out1(i,:) = exp(-kh(i)*(x-L)).*(-kh(i)*cos(kh(i)*(z+h))/cos(kh(i)*h));
end

phi5 = f.'*out;
phi5n = f.'*out1;
clear out out1
%%
%for side (6)
x=dOmega_midpts(count(3)+1:count3a,1);
z=dOmega_midpts(count(3)+1:count3a,2);

for i=1:length(f)
    out(i,:) = exp(-kh(i)*(x-L)).*cos(kh(i)*(z+h))/cos(kh(i)*h);
    out1(i,:) = exp(-kh(i)*(x-L)).*(-kh(i)*sin(kh(i)*(z+h))/cos(kh(i)*h));
end

```

```
phi6 = f.'*out;
phi6n = f.'*out1;
clear out out1
%%
%for side (7)
x=dOmega_midpts(count3a+1:count3b,1);
z=dOmega_midpts(count3a+1:count3b,2);

for i=1:length(b)
    out1(i,:) = exp(-kd(i)*(x+L)).*cos(kd(i)*(z+h1))/cos(kd(i)*h1);
    out3(i,:) = exp(-kd(i)*(x+L)).*( -kh(i)*sin(kh(i)*(z+h)))/cos(kh(i)*h);
end

for i=1:length(c)
    out2(i,:) = exp(kd(i)*(x-L)).*cos(kd(i)*(z+h1))/cos(kd(i)*h1);
    out4(i,:) = exp(kd(i)*(x-L)).*( -kh(i)*sin(kh(i)*(z+h)) )/cos(kh(i)*h);
end

phi7 = b.'*out1+c.'*out2;
phi7n = b.'*out3+c.'*out4;
clear out1 out2 out3 out4
%%

%for side (8)
x=dOmega_midpts(count3b+1:end,1);
z=dOmega_midpts(count3b+1:end,2);

for i=1:length(a)
    out(i,:) = exp(kh(i)*(x+L)).*cos(kh(i)*(z+h))/cos(kh(i)*h);
    out1(i,:) = exp(kh(i)*(x+L)).*( -kh(i)*sin(kh(i)*(z+h)) )/cos(kh(i)*h);
end
phi8 = (exp(-k0h*(x+L)).*cos(k0h*(z+h))/cos(k0h*h)).'+a.'*out;
phi8n = (exp(-k0h*(x+L)).*( -k0h*sin(k0h*(z+h))   )/cos(k0h*h)).'+a.'*out1;

clear out out1
%%
phi = [phi1,phi2,phi3,phi4,phi5,phi6,phi7,phi8];
```

```
phin = [phi1n,phi2n,phi3n,phi4n,phi5n,phi6n,phi7n,phi8n];
phi = phi(:);
phin = phin(:);

%%
%This section deals with computing the jump in potential
x=gamma_midpts(:,1);
z=gamma_midpts(:,2);

for i=1:length(b)
    out1(i,:) = exp(-kd(i)*(x+L)).*cos(kd(i)*(z+h1))/cos(kd(i)*h1);
end

for i=1:length(c)
    out2(i,:) = exp(kd(i)*(x-L)).*cos(kd(i)*(z+h1))/cos(kd(i)*h1);
end

phi2j = b.'*out1+c.'*out2;

clear out1 out2

for i=2:length(d)
    out1(i-1,:) = exp(-kappa(i)*(x+L)).*cos(kappa(i)*(z+h));
end

for i=2:length(e)
    out2(i-1,:) = exp(kappa(i)*(x-L)).*cos(kappa(i)*(z+h));
end

if length(d)==1
    temp1=zeros(size(x));
else
    temp1=d(2:end).'*out1;
end

if length(e)==1
    temp2=zeros(size(x));
```

```
else
    temp2=e(2:end).'*out2;
end
temp1=temp1(:).';   % row vector
temp2=temp2(:).';

phi3j = (d(1)*(L-x)/(2*L)).'+ temp1+(e(1)*(x+L)/(2*L)).' + temp2;

clear out1 out2

phij = phi2j-phi3j;
phij=phij(:);   % column vector

```

6.4 Q_matrix.m

```
function [Q,F]=Q_matrix(panelsL,alpha,number_modes)
%
%This program returns the matrix Q and the vector F, which represent
%respectively the boundary conditions over all the segments, and the
%incident wave.
% Inputs
% panelsL - panels on LHS of domain,
% alpha  - relates to frequency of incident wave
% number_modes  - number of modes (truncation parameter)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

NL=size(panelsL,1);
l = -panelsL(1,1); % position of the left hand edge is -l
h = - panelsL(end,4);

%We define the calculation points in the middle of each panel
mid_points_z = (panelsL(1:end,2) + panelsL(1:end,4))/2;

%Obtaining of the roots of the dispersion equation
```

```
K=dispersion_free_surface(alpha,number_modes,h);
M=length(K);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Calculation of the matrix Q :

% M is the number of evanescent modes
A=zeros(1,M);
for j=1:M,
    A(j)=0.5*(cos(K(j)*h)*sin(K(j)*h)+K(j)*h)/(K(j)*(cos(K(j)*h)^2));
end

%We begin with the boundary condition on the left hand of the domain
S = zeros(NL,M);
R = zeros(M,NL);

for k = 1 : NL,
    for j = 1 : M,
        S(k,j)=-K(j)*cos(K(j)*(mid_points_z(k)+h))/(cos(K(j)*h));
    end
end

for j = 1 : M,
    for k = 1 : NL
        R(j,k) = 1/(K(j)*A(j))*(sin(K(j)*(panelsL(k,2)+h))-sin(K(j) ...
            *(panelsL(k,4)+h)))/cos(K(j)*h);
    end
end
Q = S*R;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Calculation of the vector F :
F=zeros(NL,1);

for k=1:NL,
    F(k) = -2*exp(1*K(1))*K(1)*cos(K(1)*(mid_points_z(k)+h))/cos(K(1)*h);
end
```

```
%Note this program is based on existing code from wikiwaves.org
```

6.5 G1.m

```

function out = G1(outerboundary_panels,gamma_midpoints)
%creates matrix G1 as discussed in the dissertation

for i=1:length(gamma_midpoints)
    for j=1:length(outerboundary_panels)

            G(i,j)=type1(gamma_midpoints(i,:),outerboundary_panels(j,:));
    end
end
out=G;


%% subfunction here:%%

function out = type1(gamma_midpt,outer_panel)
%eg outer_panel = [-2 0 -1.9 0] and gamma_midpt = [-0.8889   -0.5000]
%line element is an important consideration

temp1= (gamma_midpt(1)-outer_panel(1)) / ...
    ((gamma_midpt(1)-outer_panel(1))^2+(gamma_midpt(2)-outer_panel(2))^2);
temp2= (gamma_midpt(1)-outer_panel(3)) / ...
    ((gamma_midpt(1)-outer_panel(3))^2+(gamma_midpt(2)-outer_panel(4))^2);

out=-1/(2*pi)*(temp1-temp2);

```

6.6 G2.m

```
function out = G2(outerboundary_panels,gamma_midpoints,count)
%creates matrix G2, as discussed in the dissertation.

```

```
for i=1:length(gamma_midpoints)
    for j=1:length(outerboundary_panels)

        if j<=count(1)
            G(i,j)=type1(gamma_midpoints(i,:),outerboundary_panels(j,:));
        elseif j<=count(2)
            G(i,j)=type2(gamma_midpoints(i,:),outerboundary_panels(j,:));
        elseif j<=count(3)
            G(i,j)=-type1(gamma_midpoints(i,:),outerboundary_panels(j,:));
        elseif j<=count(4)
            G(i,j)=-type2(gamma_midpoints(i,:),outerboundary_panels(j,:));
        end

    end
end
out=G;


%% %% first subfunction%%

function out = type1(gamma_midpt,outer_panel)

temp1 = log( (gamma_midpt(1) - outer_panel(1))^2+ ...
    (gamma_midpt(2) - outer_panel(2))^2 );
temp2 = log( (gamma_midpt(1) - outer_panel(3))^2+ ...
    (gamma_midpt(2) - outer_panel(4))^2 );
out = -1/(4*pi)* ( temp1-temp2 ) ;

%% %%second subfunction%%

function out = type2(gamma_midpt,outer_panel)

if gamma_midpt(2)==outer_panel(2)  %if z'=z
    temp1=pi/2
else
    temp1= atan( (gamma_midpt(1) - outer_panel(1) ) ...
        / (gamma_midpt(2) - outer_panel(2)  ) );
```

```
end

if gamma_midpt(2)==outer_panel(4)  %if z'=z
    temp2=pi/2
else
    temp2= atan( (gamma_midpt(1) - outer_panel(3) ) ...
        / (gamma_midpt(2) - outer_panel(4)  ) );
end

out= 1/(2*pi)*(temp1-temp2);
```

6.7 G3.m

```
function out = G3(outerboundary_midpoints,gamma_panels)
%creates matrix G3, as discussed in the dissertation.

for i=1:size(outerboundary_midpoints,1)
    for j=1:size(gamma_panels,1)

        G(i,j)=type1(outerboundary_midpoints(i,:),gamma_panels(j,:));
    end
end
out=G;


%% subfunction -- this is for -1/(2*pi)*atan( (x-xp) / (z-zp) )
function out = type1(outer_midpt,gamma_panel)

temp1= atan(  (outer_midpt(1) - gamma_panel(1) ) ...
    / (outer_midpt(2)-gamma_panel(2) )  );
temp2= atan(  (outer_midpt(1) - gamma_panel(3) ) ...
    / (outer_midpt(2)-gamma_panel(4) )  );

out=-1/(2*pi)*(temp2-temp1);

```

6.8 chebyshevII.m

```
function U = chebyshevII(M,x)
%Function file to compute chebyshev polynomials of the 2nd kind.

x=x(:);     %ensures it is always column vector
U=zeros(length(x),M);

U(:,1)=1;   %this is U_0(x)
U(:,2)=2*x; %this is U_1(x)

for k=3:(M)
    U(:,k)=2*(x).*U(:,k-1)-U(:,k-2);
end

```

6.9 outerpanels.m

```
function [outerboundary_panels,count] = outerpanels(stepsize,h,a1,a2)
%function [outerboundary_panels,count] = outerpanels(stepsize,h,a1,a2)
%This function file discretises a rectangular boundary [a1 a2] x [0 -h].
%Note this code assumes that a1=-a2.
%
%Parameters:    stepsize    -- distance between endpoints of panel
%               h           -- depth of fluid medium
%               a1          -- x-coord for LHS of boundary
%               a2          -- x-coord for RHS of boundary
%
%The variable count is a cumulative total of the number of points from
%each side of the boundary, starting from (a1,0), and orienting
%anticlockwise back to the beginning.

z1=0:-stepsize:-h;
x1=-2*ones(size(z1));
side1=cat(2,x1',z1');
final_side1=side1(end,:);

```

```
x2=final_side1(1):stepsize:a2;
z2=final_side1(2)*ones(size(x2));
side2=cat(2,x2',z2');
final_side2=side2(end,:);

x3=final_side2(1)*ones(size(z1));
z3=final_side2(2):stepsize:0;
side3=cat(2,x3',z3');
final_side3=side3(end,:);

x4=final_side3(1):-stepsize:a1;
z4=final_side3(2)*ones(size(z2));
side4=cat(2,x4',z4');
final_side4=side4(end,:);

side2=side2(2:end,:);
side3=side3(2:end,:);
side4=side4(2:end,:);

boundary=cat(1,side1,side2,side3,side4);
outerboundary_panels=cat(2,boundary(1:(end-1),:),boundary(2:end,:));

% %%%%%%
count1=size(x1,2)-1;     %how many midpoints on each side cumulatively
count2=size(x2,2)-1+count1;
count3=size(x3,2)-1+count2;
count4=size(x4,2)-1+count3;

count=[count1,count2,count3,count4];

```