Wave Scattering using Numerical Methods

01458 — Advanced Modelling - Partial Differential Equations

Supervisor:
Ove Skovgaard

Authors:
Ask H. Larsen, s021864
Martin F. Laursen, s021767
Kasper Reck, s021817

June 26, 2006

Technical University of Denmark
Lyngby

Abstract

This project is about combining a numerical method with an analytical method for solving partial differential equations. The specific problem used in this project is the modelling of shallow wave scattering on an object. The problem is related to the development of warning systems for tsunamis and damage estimation. This sadly became a very relevant problem on Dec 26, 2004 when a massive tsunami claimed the lives of 230,000 people in one of the worst natural disasters in written history.

In the report we first discuss the theory of wave propagation and formulate the wave problem and boundary conditions as a partial differential equation. This is done for a simple mirror symmetric geometry and solved analytically. The problem is then solved using the FEM-tool COMSOL™, which yields almost identical results. The next step is to *couple* the analytical solution to the numerical solution, using basis functions of the analytical solution in the far field as boundary conditions to a numerical problem in the vicinity of the object where analytical solution is in most cases impossible. This method is validated by comparison to the analytical solution and is in good agreement.

We proceed to use expressions based on the analytical solution as boundary conditions for the numerical problem, and solve similar problems for arbitrary and increasingly complex geometries, such as ellipses and later realistic coast lines.

The last step is modify the problem to account for variable water depth near the "coast" of the scattering object and solve the problem again. This step is the final major step towards a fully realistic model of the waves hitting actual countries, and is therefore important if the model is to be used in actual warning systems. However it must be underlined that this project investigates the solution method and should be regarded merely as a proof of concept regarding tsunami dynamics.

Contents

Preface

This report is a part of the 3-week course "01458 — Advanced modelling - Partial differential equations".

We would like to thank our brilliant teacher Ove Skovgaard. He has been a great motivator and inspiration to all of us. We would also like to thank Niels Aage for his help on the Comsol code.

Chapter 1

Introduction

1.1 Introduction

On August 26 1883 after several months of increasing volcanic activity and minor eruptions, the volcano on the island of Krakatoa near Indonesia, see Figure 1.1, erupted in four major explosions. The yield of the explosions is estimated to be equivalent to 200 megatons of TNT, making it the largest in recorded history. The explosions were heard as far away as 3500km - a twelfth of the planet's surface. As a result, ash was hurled 80km into the atmosphere. The death toll from the explosion was enormous. Pyroclastic flows killed 1000 people on the island of Ketimbang near Sumatra, after traveling 40km on a cushion of superheated steam across the ocean. There were no survivors of the 3000 people on the island Sebesi 13 km from Krakatoa. The biggest killer however, was the series of giant tsunami waves caused by the gigantic pyroclastic flows entering the sea. In some places 30m high waves were reported, and tsunami waves were reported as far away as the english channel. When the devastation had ended the death toll had reached 36,417 people world wide. [7]

121 years later on December 26, 2004 an earthquake of magnitude 9.15 on the Richter scale struck in the Indian Ocean, see Figure 1.3. The earthquake started a tsunami wave not even one meter tall but several hundreds of kilometers long. As the wave closed in on the coastline of Indonesia, Thailand, India and even the African continent, it rose to

Figure 1.1: *Krakatoa is placed in the Sunda Strait near Indonesia. [7].*

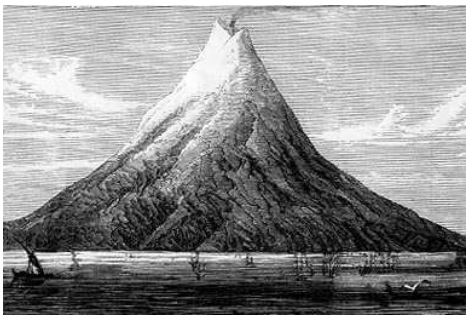


Figure 1.2: *Krakatoa as it looked before the explosion. [7].*

Figure 1.3: *The epicenter of the earthquake in 2004 was near Jakarta of Indonesia. [8]*

as much as 30m in height and penetrated several kilometers inland, killing 230,000. The high number of casualities made the 2004 tsunami the deadliest tsunami ever. The main reason of the many victims was the lack of a warning system in the Indian Ocean. This is partly because the region is one of the poorest in the world, and partly because a major tsunami event had not occured since Krakatoa. United Nations has since started working on an Indian Ocean Tsunami Warning System. [8]

Part of an effective warning system could be a prediction of where the tsunami would strike and how much damage it would cause in specific regions. A way to give a rough estimate of this is to run a numerical simulation of the tsunami wave, and estimate the impact on the different countries. This project will try to make such numerical simulations. We will start with analytical solutions to actual wave problems, and later combine analytical and numerical methods to expoit the strengths of both methods. The result should be fast simulations of waves hitting arbitrary geometries. To make the simulations as close to reality as possible we will also implement variable water depth in the numerical model. These results could be used as a warning system for tsunamis. However far from all earthquakes results in tsunamis. On March 28, 2005 an earthquake with a magnitude of 8.7 in roughly the same area as the one Dec 26, did not create any tsunamis. The numerical simulation is therefore not meant as a final warning system, but merely as part of one. By solving for many different earthquakes and wave scenarios it will be possible

Figure 1.4: *The tsunami wave killed 230,000 people world wide when it hit land.* David Rydevik*[8]*

to make fast predictions of tsunamis.

1.2 Report overview

We have three major sections in the report.

- Mathematical model
- Numerical solution and comparison
- Extension to complicated geometries

In the mathematical model section the wave problem is formulated and solved. In this section we only consider simple geometries i.e a plane wave hitting a cylinder. The details in the solution method is discussed and the results are presented.

In the next chapter we investigate how to transform the analytical problem into a numerical problem. To solve this we use the FEM solver COMSOL™.

In the section "Extension to complicated geometries" we couple the numerical and analytical solution together: we use the analytical solution in the far field and as the boundary condition for the numerical solution. In the vicinity of the diffracting object the problem is solved numerically. In this section we also introduce arbitrary geometries, such as ellipses and countries. At the end of the chapter we take variable water depth into acount in the model.

Chapter 2

Mathematical model

ARTHUR: Camelot!
GALAHAD: Camelot!
LAUNCELOT: Camelot!
PATSY: It's only a model.
ARTHUR: Shhh!

Monty Python and the Holy Grail

2.1 Analytic solution for a simple wave problem

We will in this section consider a simple wave problem which can be solved analytically. We consider an infinite ocean with a flat floor of some depth h in which there is a massive cylindrical construction of radius a, see Figure 2.1. The fish is assumed to be infinitely small.

Before we try to find an analytical solution we will have to choose an appropriate coordinate system. Since the construction is a cylinder it is natural to choose a cylindrical coordinate system, as shown on Figure 2.2, where we have put $z = 0$ at the ocean surface. We wish to describe the propagation of waves in the vicinity of this construction, specifically waves of very long wave length $L \gg h$ such as tsunami waves[1]. This *shallow water* assumption will greatly simplify the model. First we will consider the propagation of such waves in an entirely *empty* sea, and the results from this model will be applied to obtain the desired description of the dynamics near the construction.

In the open-sea case we consider a simple wave, η_i, propagating in some direction which we happen to define as the x direction. We require that the wave be harmonic, i.e. described by a time dependent factor $\exp(-i\omega t)$ and spatial variation $\exp(ikx)$ where ω is the angular frequency defined as $\omega = \frac{2\pi}{T}$, where T is the period and assumed constant, t is the time and k is the wavenumber defined as $k = \frac{2\pi}{L}$. To keep the mathematics as simple as possible we assume that the flow of the 3 dimensional fluid is irrotational, incompressible and that fluid motion mainly occurs at the surface. The incident harmonic wave will scatter on the cylinder and produce some resultant field, the precise form of which we shall derive in the following.

By integrating the fluid motion over the depth h, and changing from rectangular to cylindrical coordinates we can make the following transition into polar coordinates

$$(x, y, z) \mapsto (x, y) \mapsto r(\cos\theta, \sin\theta). \tag{2.1.1}$$

[1] Tsunamis can have wave lengths of several hundred kilometres according to [2].

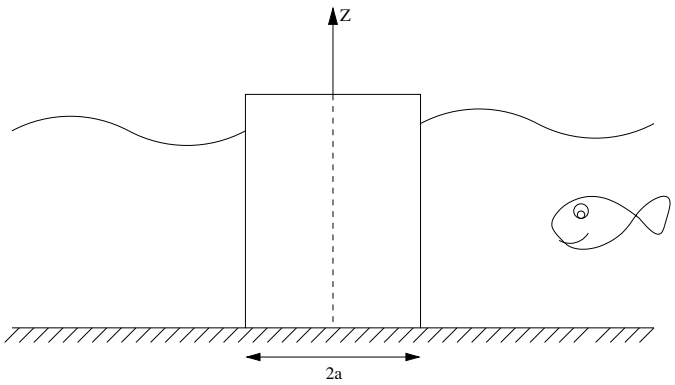


Figure 2.1: *Vertical cross section of the cylinder on the flat ocean floor. a is the radius of the cylinder and h is the depth. The cylindrical symmetry axis is denoted z.*

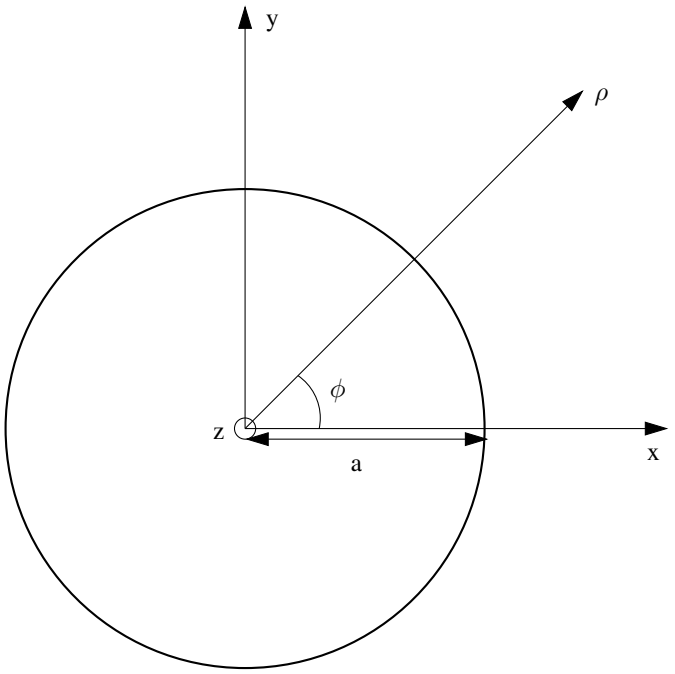


Figure 2.2: *The cylindrical coordinatesystem. The symmetry axis z is pointing out of the paper.*

The wave equation for a two dimensional fluid is in terms of the vertical displacement $\psi(x, y, t)$ in Cartesian coordinates given as

$$\Box^2\psi \equiv \left(\nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right)\psi = 0, \tag{2.1.2}$$

where ∇^2 is the Laplacian. It is easy to separate this partial differential equation into a time-dependent part which allows the canonical solution already introduced above, $\tau(t) = \exp(-i\omega t)$ (where the sign convention enforces that the wave will propagate in the positive x direction), and a spatially dependent part $\eta(x, y)$ governed by the *Helmholtz equation*

$$\left(\nabla^2 + k^2\right)\eta = 0, \tag{2.1.3}$$

such that a complete wave solution is $\psi = \eta\tau$. If we focus only on the spatially dependent motion of the fluid, we only have to solve Helmholtz equation. The time dependence can at any point be reinstated by multiplying by $\tau = \exp(-i\omega t)$.

2.2 Determining boundary conditions

We now return to the case where we have a cylinder of radius a placed at the origin. The incident wave, η_i, cannot penetrate the surface of the cylinder and will therefore be totally reflected, creating a scattered wavefield, η_s. Hence we regard the resulting wave field, η, as the sum of the incident wave field η_i and the scattered wave field η_s, or

$$\eta(x, y) = \eta_i(x, y) + \eta_s(x, y), \tag{2.2.1}$$

$$\eta_i(x, y) = \alpha\exp(ikx). \tag{2.2.2}$$

If we know the incident and the scattered wave field we can thereby find the resulting wave field using Equation (2.2.1). Unfortunately, it is not possible to formulate practical boundary conditions for the resulting wave field η at infinity. We therefore reformulate Equation (2.2.1) as

$$\eta_s(x, y) = \eta(x, y) - \eta_i(x, y) \tag{2.2.3}$$

with the intention of solving for η_s which will later prove to have more favourable boundary conditions.

The Helmholtz equation for the incident wave field (Equation (2.1.3)) is also valid for the resulting wave field η, if we assume that the amplitude is small. Note that $k = \frac{2\pi}{L}$ is indeed a constant, since $L = cT$ where both T and c are constants[2]. Hence we conclude that since η and η_i each satisfy the Helmholtz equation, so does the scattered wave field η_s, and we have

$$\left(\nabla^2 + k^2\right)\eta_s(x, y) = 0. \tag{2.2.4}$$

or in polar coordinates

$$\left(\nabla^2 + k^2\right)\eta_s(r, \theta) = 0. \tag{2.2.5}$$

As we will show in a moment, we are able to define boundary conditions at infinity for η_s, and we can then use the solution of the above equation, together with Equation (2.2.1),

[2] A result of constant depth.

to find the resulting wave field. Equation (2.2.5) is a second order partial differential equation in both r and θ, and thus we need two boundary conditions for each, or four in total. First we will demand that any wave that hits the cylinder will be totally reflected at its surface, since there it cannot flow into the cylinder, i.e. the wave field will only have a horizontal slope. Physically this means that the water surface only moves in the up and down direction. Mathematically this can be expressed as

$$\left[\frac{\partial \eta}{\partial r}\right]_{r=a} = 0. \tag{2.2.6}$$

Secondly, since there is no energy source at infinity (that is at $r \to \infty$), η_s must be going outwards, it cannot go inwards, or in other words; if an observer is looking towards the cylinder from infinity, he/she will only see the reflections from the cylinder, not from any other direction.

We now have two boundary conditions for r. The model must nessesarily be 2π periodic, we therefore define the two boundary conditions for the θ coordinate as

$$\eta(r, \theta) = \eta(r, \theta + 2\pi), \tag{2.2.7}$$

and

$$\frac{\partial \eta(r, \theta)}{\partial \theta} = \frac{\partial \eta(r, \theta + 2\pi)}{\partial \theta}. \tag{2.2.8}$$

We now have all four boundary conditions, and using Equation (2.2.1) we can reformulate these in terms of the scattered and incident wave field as

$$\left[\frac{\partial \eta_i}{\partial r} + \frac{\partial \eta_s}{\partial r}\right]_{r=a} = 0 \tag{2.2.9}$$

$$\text{For } r \to \infty \qquad \eta_s \text{ is going outward} \tag{2.2.10}$$

$$\eta_s(r, \theta) = \eta_s(r, \theta + 2\pi) \tag{2.2.11}$$

$$\frac{\partial \eta_s(r, \theta)}{\partial \theta} = \frac{\partial \eta_s(r, \theta + 2\pi)}{\partial \theta}. \tag{2.2.12}$$

2.3 Separation of variables

We shall now find a *product solution* for η_s in Equation (2.2.5). Specifically we assume the existence of a solution[3]

$$\eta_s(r, \theta) = R(r)\Theta(\theta). \tag{2.3.1}$$

where R and Θ are yet unknown functions. Note that the Laplacian in polar coordinates is [1, p. 196]

$$\nabla^2 = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2}, \tag{2.3.2}$$

such that η_s becomes

$$\nabla^2 \eta_s = \nabla^2(R\Theta) = \frac{1}{r}\frac{\partial}{\partial r}\left(rR'\Theta\right) + \frac{1}{r^2}R\Theta'' \tag{2.3.3}$$

[3]Note that the scattered wave must diminish towards 0 with increasing r, which makes linear combinations of such expressions eligible for solutions. This approach would be impossible if the wave did not diminish, because of inhomogeneous boundary conditions.

$$= R''\Theta + \frac{1}{r}R'\Theta + \frac{1}{r^2}R\Theta''. \tag{2.3.4}$$

Substituting into the differential Equation (2.2.5) results in the equation

$$R''\Theta + \frac{1}{r}R'\Theta + \frac{1}{r^2}R\Theta'' + k^2R\Theta = 0 \tag{2.3.5}$$

which can be rewritten, multiplying by $r^2(R\Theta)^{-1}$ and rearranging terms to appropriate sides of the equation, to

$$-\frac{\Theta''}{\Theta} = r\frac{R'}{R} + r^2\left(\frac{R''}{R} + k^2\right). \tag{2.3.6}$$

Notice that the left hand and right hand sides are dependent *only* on θ and r, respectively, and in order for the equation to be true for *all* values of these variables, the expressions of each side must be equal to the same *constant*, say λ. We have thus separated the partial differential equation, obtaining the two ordinary differential equations

$$\Theta'' + \lambda\Theta = 0 \tag{2.3.7}$$

$$r^2R'' + rR' + \left(r^2k^2 - \lambda\right)R = 0. \tag{2.3.8}$$

The former equation allows 2π-periodic solutions *only* if λ is a *non-negative real*. Further it will prove convenient to perform the substitution $\lambda = n^2$, since all $2\pi - periodic$ solutions Θ_n can then be written (cf. the well-known solution of second-order linear ordinary differential equations)

$$\Theta_n(\theta) = A_n\cos n\theta + B_n\sin n\theta, \quad n \in \mathbb{N}_0. \tag{2.3.9}$$

Note that the special case $n = 0$ is consistent with this notation since it eventually contributes with a constant term (physically corresponding to a wave with constant amplitude on circles concentric with the origin). The solution must necessarily be mirror symmetric about the x axis, since the incident wave, PDE and boundary conditions all possess this symmetry. This translates to a requirement that Θ_n be *even*, which can be achieved only for $B_n = 0$ except for $n = 0$ in which case the sine-dependent term is zero anyway. The solution can now be written

$$\Theta_n(\theta) = A_n\cos n\theta, \quad n \in \mathbb{N}_0. \tag{2.3.10}$$

It is now time to turn to the differential equation for R, Equation (2.3.8), which by introduction of the index n reads

$$r^2R_n'' + rR_n' + \left(r^2k^2 - n^2\right)R_n = 0, \quad n \in \mathbb{N}_0. \tag{2.3.11}$$

This is the *parametric form* of the *Bessel* equation of order n, which we will first rewrite to the ordinary Bessel equation. By making the substitution $u = kr \Rightarrow \mathrm{d}u = k\mathrm{d}r$ and introducing the function χ, $R(r) = R\left(\frac{u}{k}\right) = \chi(u)$ the *chain rule* can be used to rewrite the ODE, since

$$R_n'(r) = \chi_n'(u)\frac{\mathrm{d}u}{\mathrm{d}r} = k\chi_n'(u) \tag{2.3.12}$$

$$R_n''(r) = k^2\chi_n''(u), \tag{2.3.13}$$

which inserted into Equation (2.3.11) yields

$$\begin{aligned} r^2k^2\chi_n'' + rk\chi_n' + (r^2k^2 - n^2)\chi_n &= \\ u^2\chi_n'' + u\chi_n' + (u^2 - n^2)\chi_n &= 0, \end{aligned} \tag{2.3.14}$$

which has the general solution [1, p. 242]

$$\chi_n(u) = P_nJ_n(u) + Q_nY_n(u), \tag{2.3.15}$$

where J_n and Y_n are the Bessel functions of first and second kind, respectively, of order n. The boundary condition at infinity states that the waves are outgoing at infinity. It is therefore convenient[4] to use the so-called Hankel funtions, defined as

$$\begin{aligned} H_n^{(1)}(x) &\equiv J_n(x) + iY_n(x) \\ H_n^{(2)}(x) &\equiv J_n(x) - iY_n(x), \end{aligned} \tag{2.3.16}$$

where i is the imaginary unit. Using these functions the general solution to the Bessel ODE can be rewritten as

$$R_n(r) = P_n^*H_n^{(1)}(kr) + F_n^*H_n^{(2)}(kr). \tag{2.3.17}$$

For $r \to +\infty$ it is known [3, table 16-1] that $H_n^{(2)}(kr)\exp(-i\omega t)$ will create a wavefront propagating inward. The boundary condition (2.2.10) will therefore exclude this term of the equation yielding

$$R_n(r) = P_n^*H_n^{(1)}(kr). \tag{2.3.18}$$

The scattered wavefield η_s is then found from a superposition of product solutions

$$\begin{aligned} \eta_s(r,\theta) &= \sum_{n=0}^{+\infty} R_n(r)\Theta_n(\theta) \\ &= \sum_{n=0}^{+\infty} P_n^*H_n^{(1)}(kr)A_n\cos n\theta \\ &= \sum_{n=0}^{+\infty} D_nH_n^{(1)}(kr)\cos n\theta \quad \text{where } D_n = A_nP_n^*. \end{aligned} \tag{2.3.19}$$

The incident wave field, Equation (2.2.2), can be written as [4, entry 94]

$$\eta_i(x,y) = \alpha\exp(ikx) = \alpha\exp(ikr\cos\theta) = \alpha\sum_{n=0}^{+\infty}\epsilon_ni^nJ_n(kr)\cos(n\theta), \tag{2.3.20}$$

where ϵ_n is

$$\epsilon_n = \begin{cases} 1, & n = 0 \\ 2, & n = 1, 2\ldots \end{cases} \tag{2.3.21}$$

The total solution is given by Equation (2.2.1) in polar coordinates:

$$\eta(r,\theta) = \sum_{n=0}^{+\infty}\left[\alpha\epsilon_ni^nJ_n(kr) + D_nH_n^{(1)}(kr)\right]\cos n\theta \tag{2.3.22}$$

[4]This will become clear in a moment

Using the first boundary condition then yields

$$\frac{\partial \eta}{\partial r} = \sum_{n=0}^{+\infty} \left[\alpha \epsilon_n i^n k J_n'(ka) + D_n k H_n'^{(1)}(ka) \right] \cos n\theta = 0 \Leftrightarrow$$

$$\sum_{n=0}^{+\infty} D_n k H_n'^{(1)}(ka) \cos n\theta = -\sum_{n=0}^{+\infty} \alpha \epsilon_n i^n k J_n'(ka) \cos n\theta \Rightarrow$$

$$D_n = -\alpha \epsilon_n i^n \frac{J_n'(ka)}{H_n'^{(1)}(ka)}, \tag{2.3.23}$$

where we have used that two Fourier series are equal if and only if they are coefficient-wise equal. Inserting this in Equation (2.3.22) we have

$$\begin{aligned} \eta(r,\theta) &= \sum_{n=0}^{+\infty} \left[\alpha \epsilon_n i^n J_n(kr) - \alpha \epsilon_n i^n \frac{J_n'(ka)}{H_n'^{(1)}(ka)} H_n^{(1)}(kr) \right] \cos n\theta \\ \frac{\eta(r,\theta)}{\alpha} &= \sum_{n=0}^{+\infty} \epsilon_n i^n \left[J_n(kr) - \frac{J_n'(ka)}{H_n'^{(1)}(ka)} H_n^{(1)}(kr) \right] \cos n\theta, \end{aligned} \tag{2.3.24}$$

where the final expression has been normalized. This is the analytical solution of the proposed problem. Finally we note that the differential quotients of the Bessel- and Hankel functions can be evaluated using the formulae

$$\begin{aligned} xJ_n'(x) - nJ_n(x) &= -xJ_{n+1}(x) \\ xH_n^{(1)\prime}(x) - nH_n^{(1)}(x) &= -xH_{n+1}^{(1)}(x), \end{aligned} \tag{2.3.25}$$

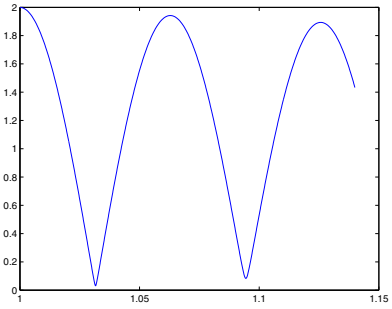
which make function evaluations practical.

2.4 Discussion of results

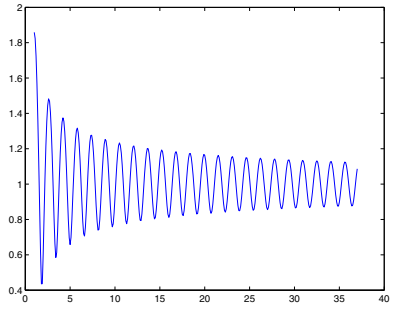
Let us finally consider some of the primary geometrical features of the solution to verify its credibility and physical significance. Figure 2.3 shows radial plots of the *amplitude* of η for angles $\theta = \pi$ and $\theta = 0$. For the $\theta = \pi$ direction which corresponds to the negative x-axis, it is observed that the solution for *small* r (Figure 2.3a) resembles a *standing wave*. This is because for $r \ll a$, the cylinder locally resembles a *plane* which would indeed result in a standing wave with amplitude twice that of the incident wave [9, p. 192]. For larger r the reflected wave disperses, and the incident wave expression, Equation (2.2.2), will be dominant; this wave has an amplitude of 1, and indeed Figure 2.3b shows part of the decay of amplitude oscillations towards 1.

On the far side of the cylinder, i.e. for $\theta = 0$, there is a "valley" in amplitude, see Figure 2.3c. This plot is made for $k \gg a$, and the amplitude at the cylinder surface is quite low (around 0.15). The amplitude increases steadily towards 1 for larger r. Last, Figure 2.3d where $\theta = 0$ and $k \ll a$ hardly shows any change in amplitude at all. Thus if the wavelength is much longer than the cylinder then the cylinder will not have any significant influence on the overall flow. Compare this to the well-known result that you cannot measure particles by means of waves with wavelength larger than the particle size.

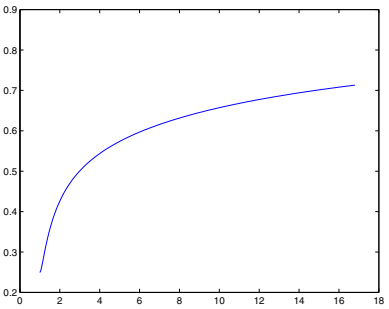
Figures 2.4 and 2.5 show a three-dimensional plot of the absolute value of the total wave field, and a contour plot of the the region around the cylinder, respectively. It appears that curves of nodes and anti-nodes "bend" around the cylinder.



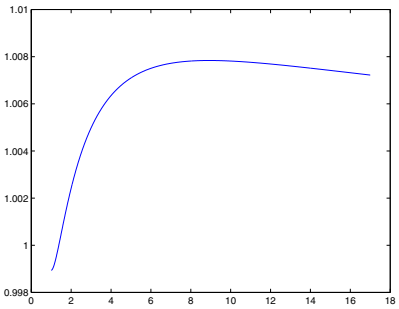
(a) *Near the cylinder on the angle of incidence* $\theta = \pi$ *the amplitude oscillations resemble a standing wave. Here* $ka = 50$.



(b) *Far from the cylinder for* $\theta = \pi$ *the interference due to the cylinder decays and the amplitude approaches that of the incident wave, which is* 1. *Here* $ka = 2$.



(c) *Immediately on the far side of the cylinderm* $\theta = 0$, *the amplitude is nearly* 0, *i.e. the cylinder screens the surface from the indident wave. Further away the amplitude increases toward that of the incident wave.* $ka = 13$.



(d) *For high wavelengths* $ka = 1/5$ *and* $\theta = 0$, *almost no scattering occurs. This is consistent with well-known results of wave mechanics. It is interesting that the amplitude is slightly* larger *than* 1 *in some places.*

Figure 2.3: *Radial plots of the amplitude of the solution* η. *On all plots the radius of the cylinder is* $a = 1$ *(i.e.* r *and* k *take* relative *values).*

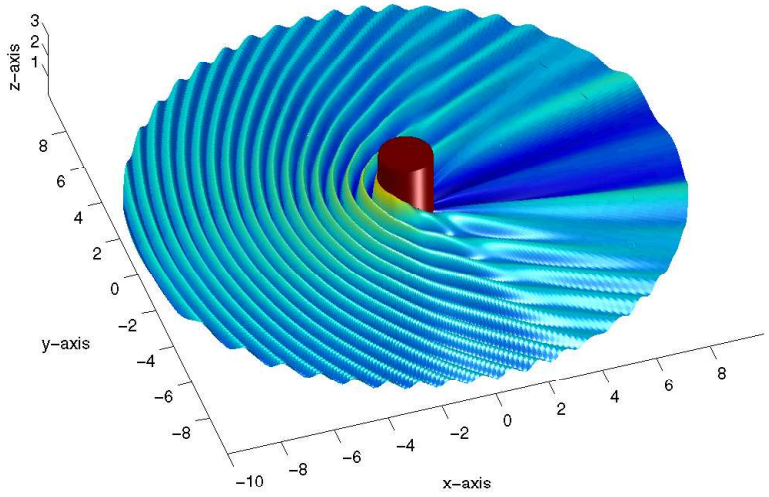


Figure 2.4: *Three-dimensional plot of the exact amplitude of the solution η from Equation* (2.3.24). *The red cylinder represents the construction.*

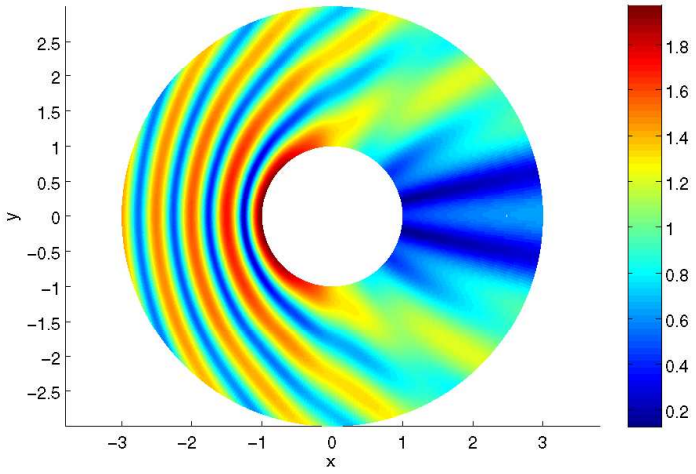


Figure 2.5: *Contour plot of the exact solution η of Equation* (2.3.24).

Chapter 3

Numerical solution and comparison

This chapter deals with the transformation of the differential problem of the previous chapter into one which can be solved numerically, specifically using the *Finite Element Method* (FEM). This method will then be applied and the solution compared to the analytical one.

3.1 PDE to FEM transformation

We will in this section show how to transform a PDE problem, defined in some closed region R, into a set of linear equations. In this example we will consider the *Poisson equation* in two dimensions, defined as

$$\nabla^2 u(x,y) \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y). \tag{3.1.1}$$

If we want to solve this equation using the finite element method, we will have to restrict it to a closed and bounded region R. Rather arbitrarily we choose the region shown in Figure 3.1. To simplify the example as much as possible we demand that $u = 0$ on the boundaries. The transformation could be done with non-zero boundary conditions, but it would complicate the derivation, and thereby the understanding of the fundemental transformation procedure, more than necessary. The partial differential equation, (3.1.1), together with the boundary conditions is called the *strong form* of the PDE problem. We will later derive another form called the *weak form*. The usual approach to this problem would be to apply the method of separation of variables, yet this is only possible if all but one of the parameters are constant along every boundary segment which is clearly not the case on the diagonals in Figure 3.1. Instead we will divide R into smaller, triangularly shaped subdomains. We do this by selecting n interior points $\boldsymbol{p}_j$ and connecting them to their nearest neighbor points and their nearest boundary corner points. This is illustrated on Figure 3.2 where $n = 4$. It can be shown that when using this method, the total number of triangles, $\triangle_{\text{Total}}$, is given by

$$\triangle_{\text{Total}} = \triangle_{\text{Boundary}} - 2 + 2n, \tag{3.1.2}$$

where $\triangle_{\text{Boundary}}$ is the number of boundary triangles. Hence on Figure 3.2, where $n = 4$, we have 16 triangles.

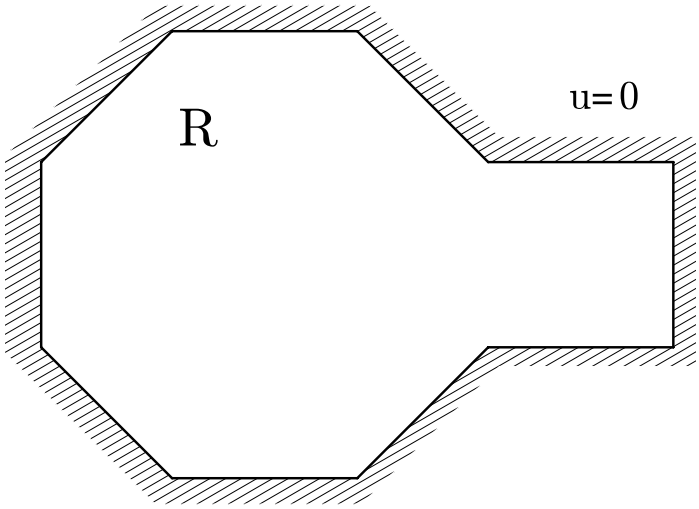


Figure 3.1: *The geometry defining the closed region R is a 10 sided polygon. For all boundaries $u = 0$.*

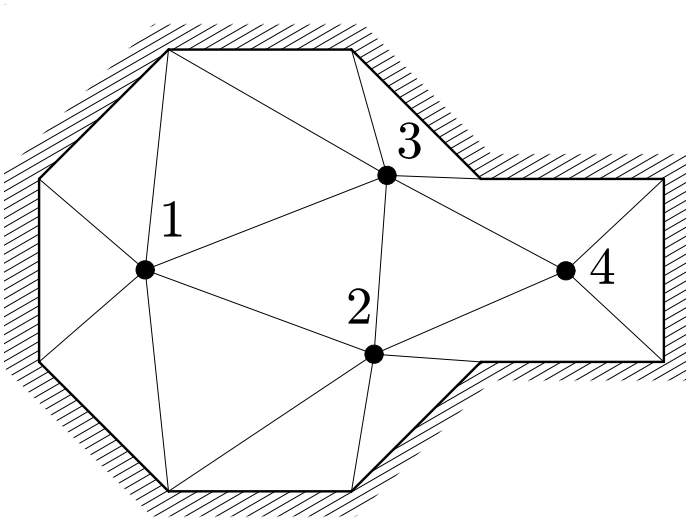


Figure 3.2: *The region R is divided into 16 smaller subdomains by inserting 4 interior points and connecting them to their nearest neighbor and their nearest boundary corner points.*

We now introduce a set of functions called the *trial* functions $T_j : R \mapsto \mathbb{R}$, $j = 1 \ldots n$, corresponding to each of the n chosen interior points. Further we require that each of these functions be non-zero *only* in the subdomains adjacent to the corresponding interior point, green area in Figure 3.3, and that for each interior point $\boldsymbol{p}_j$, $T_j(\boldsymbol{p}_j) = 1$ (we say that the trial functions have *local support*). For the sake of simplicity one can use *affine* trial functions, though in practice they could be any continuously decaying functions with the above constraints. Note that COMSOL™ uses quadratic trial functions.

Furthermore define the *test* functions $E_i : R \mapsto \mathbb{R}$ for $i = 1 \ldots n$. In reality we have some freedom in choosing these functions, but it greatly simplifies matters if the test functions are simply equal to the trial functions. Thus we set $E_j = T_j$. Note that *if* the test functions are indeed kept different from the trial functions, this will carry through to the results of this section and result in the loss of desirable symmetry.

We now multiply the differential equation by a test function, i.e.

$$T_j \nabla^2 u = T_j f. \tag{3.1.3}$$

Integrating this expression over the entire domain of definition R, yields

$$\iint_R T_j \nabla^2 u \mathrm{d}A = \iint_R T_j f \mathrm{d}A. \tag{3.1.4}$$

We wish to eliminate the Laplacian ∇^2 and achieve an expression in which only one differentiation is carried out. To do this we shall apply the well-known formula

$$\boldsymbol{\nabla} \cdot (f \boldsymbol{\nabla} g) = f \nabla^2 g + \boldsymbol{\nabla} f \cdot \boldsymbol{\nabla} g, \tag{3.1.5}$$

which, using $f = T_j$ and $g = u$ results in the expression

$$\iint_R \{\boldsymbol{\nabla} \cdot (T_j \boldsymbol{\nabla} u) - \boldsymbol{\nabla} T_j \cdot \boldsymbol{\nabla} u\} \mathrm{d}A = \iint_R T_j f \mathrm{d}A. \tag{3.1.6}$$

The first term in the left hand side of this equation can, by using Gauss' theorem, be written as

$$\iint_R \boldsymbol{\nabla} \cdot (T_j \boldsymbol{\nabla} u) \mathrm{d}A = \oint T_j \boldsymbol{\nabla} u \cdot \boldsymbol{n} \mathrm{d}s, \tag{3.1.7}$$

where the vector $\boldsymbol{n}$ is an outward pointing unit normal vector as seen on Figure 3.4. This yields

$$\oint T_j \boldsymbol{\nabla} u \cdot \boldsymbol{n} \mathrm{d}s - \iint_R \boldsymbol{\nabla} T_j \cdot \boldsymbol{\nabla} u \mathrm{d}A = \iint_R f T_j \mathrm{d}A, \tag{3.1.8}$$

when inserted into the previous equation. Since the trial function T_j is zero outside the adjacent triangular subdomains, and thus along the boundary of R (see Figure 3.3), the line integral is equal to zero. We now have:

$$\iint_R \boldsymbol{\nabla} T_j \cdot \boldsymbol{\nabla} u \mathrm{d}A = -\iint_R f T_j \mathrm{d}A, \quad j = 1, 2 \ldots n. \tag{3.1.9}$$

This is the previously mentioned *weak form* of the PDE, Equation (3.1.1). We now introduce an approximation function U which approximates the exact solution u

$$\underbrace{U(x,y)}_{\approx u} \equiv \sum_{i=1}^{n} B_i T_i(x,y), \tag{3.1.10}$$

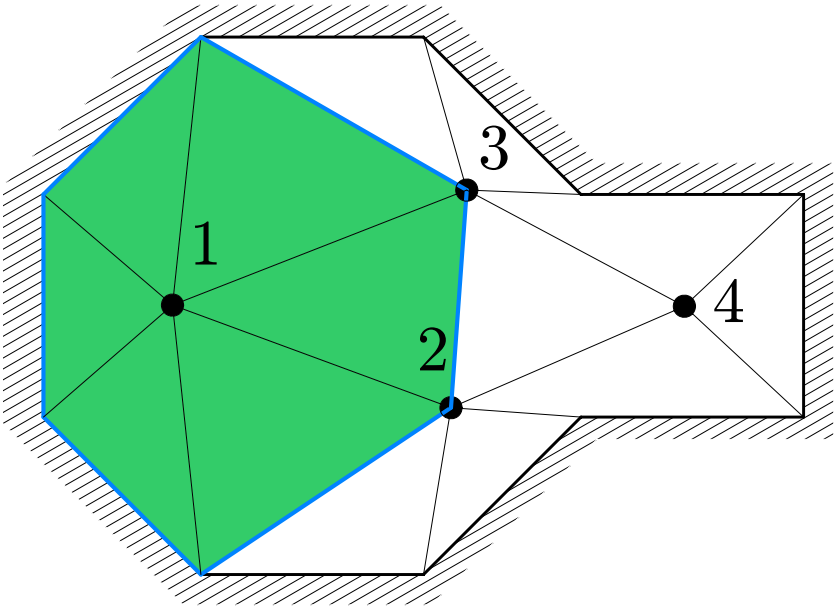


Figure 3.3: *The trial function is zero outside the local region surrounding its associated internal point (green). At point 1 it takes the value one and is decreasing towards the perimeter (blue) of the local region. Along the perimeter the trial function is zero and the line integral is therefore also zero.*

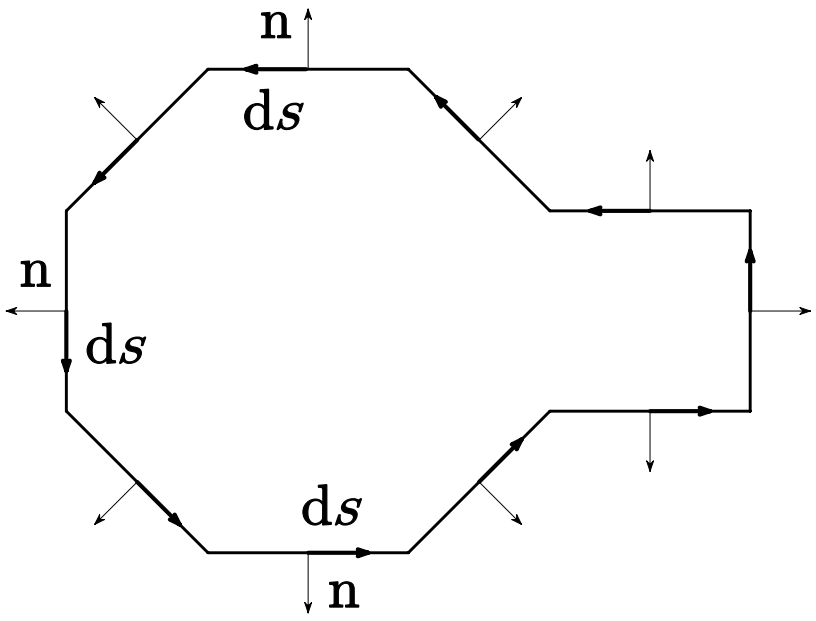


Figure 3.4: *The orientation of the boundaries equipped with outward pointing normal vectors.*

where B_i are constants. When this approximation is inserted into Equation (3.1.9), the equation yields

$$\iint_R \boldsymbol{\nabla} T_j \cdot \boldsymbol{\nabla} \left(\sum_{i=1}^{n} B_i T_i \right) \mathrm{d}A = - \iint_R f T_j \mathrm{d}A, \quad j = 1, 2 \ldots n \tag{3.1.11}$$

$$\sum_{i=1}^{n} B_i \iint_R \boldsymbol{\nabla} T_i \cdot \boldsymbol{\nabla} T_j \mathrm{d}A = - \iint_R f T_j \mathrm{d}A, \quad j = 1, 2 \ldots n. \tag{3.1.12}$$

We now have n linear equations with n unknowns B_i. We now define the $n \times n$-dimensional symmetrical *stiffness matrix* $\mathbf{K} = [K_{ij}]$ and the n-dimensional load vector $\boldsymbol{F} = [F_j]$ by

$$K_{ij} = \iint_R \boldsymbol{\nabla} T_i \cdot \boldsymbol{\nabla} T_j \mathrm{d}A \quad i, j = 1, 2 \ldots n \tag{3.1.13}$$

$$F_j = \iint_R f T_j \mathrm{d}A \quad j = 1, 2 \ldots n. \tag{3.1.14}$$

The problem can now be rewritten in matrix notation as

$$\mathbf{K}\boldsymbol{B} = \boldsymbol{F}. \tag{3.1.15}$$

Thus the previous partial differential equation (i.e. the strong form) has been transformed to a set of linear equations, the solution of which (in terms of the constants B_j) will yield an approximation to the solution u of the original problem.

In conclusion we note that the previous *differential* problem which was *explicitely dependent* on the coordinates (x, y) has been transformed into a *purely algebraic* problem in which *no* differentiation takes place. The loss of coordinate dependency stems from the integration over the entire region R in the expressions (3.1.13) and (3.1.14). Also note – as stated previously – that the jth test function has been defined such that it is 1 on the interior point $\boldsymbol{p}_j$ at which point all other test functions are 0. This means that B_j will be equal to the value of the approximated solution U at the point $\boldsymbol{p}_j$, i.e.

$$U(\boldsymbol{p}_j) = B_j, \tag{3.1.16}$$

which shows the mathematical significance of the constants B_j. Another quite important consequence of the local support of the test functions is that if two interior points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ are not adjacent (i.e. they are not vertices of the same triangular subdomain of R) the integral in Equation (3.1.13) is zero. For a large problem with many thus separated interior points, most of these integrals will therefore cancel and only those corresponding to geometrically neighbouring points will remain. This will have important consequences for the application of the results of this section numerically since most of a computer's calculatory power will be expended on evaluating the stiffness matrix.

3.2 COMSOL™ implementation

In order to find solutions for more complicated geometries than for the cylinder discussed in Chapter 2, one needs to make use of numerical approximation, such as the finite element method (FEM) discussed in Chapter 3.1. We will still be using the same Helmholtz differential equation, and as in the analytical solution we will therefore have to define

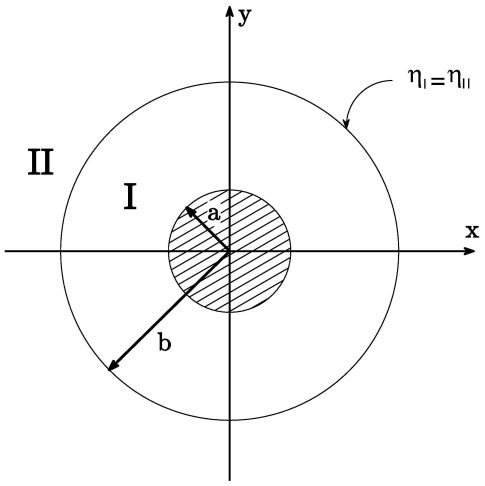


Figure 3.5: *Since the FEM requires finite boundaries, the FEM model is limited to the area between the cylinder and a circle with radius b. The model is divided into two subdomains. The analytical solution is computed in II and the numerical in I. To make the solution in subdomain I unique, continuity at the border between the subdomains is demanded.*

two boundary conditions for each coordinate. At the cylinder surface we assume total reflection, mathematically expressed as

$$\left[\frac{\partial \eta}{\partial r}\right]_{r=a} = 0. \tag{3.2.1}$$

In the analytical solution we could define boundary conditions at infinity, but for a finite element solution this is not possible. We therefore limit the FEM solution to a circle of arbitrary radius b, see Figure 3.5. The FEM solution will thus only be calculated for $a < r < b$ and $-\pi < \theta < \pi$. At the outer boundary, $r = b$, we can use the analytic solution as a Dirichlet boundary condition. If we divide the model into two subdomains, I and II (see Figure 3.5), we may use the numerical solution in subdomain I and the analytical solution in subdomain II and demand continuity on the boundary, that is

$$[\eta(II)]_{r=b} = [\eta(I)]_{r=b}. \tag{3.2.2}$$

Physically it seems reasonable to demand continuity between the two subdomains since we expect the wave to be continous. We now have all the boundary conditions for the FEM model, and it is now possible to implement the model in the commercial FEM application COMSOL™ The code for setting up and solving the problem is included in Appendix A. The analytically determined values on the boundary, used for the Dirichlet condition in the FEM solution, are calculated numerically in MATLAB™ by evaluating Equation (2.3.24). Since we cannot include an infinite number of terms in the summation we have to make a stop criterion. We do this by comparing the last computed term to the total sum. If the last term is numerically a factor γ smaller than the total sum we assume it is reasonably close to its limit and stop the computation. The condition can be

written as

$$\left| \frac{\epsilon_m i^m \left[J_m(kr) - \frac{J'_m(ka)}{H'^{(1)}_m(ka)} H^{(1)}_m(kr) \right] \cos m\theta}{\sum_{n=0}^{m} \epsilon_n i^n \left[J_n(kr) - \frac{J'_n(ka)}{H'^{(1)}_n(ka)} H^{(1)}_n(kr) \right] \cos n\theta} \right| < \gamma. \tag{3.2.3}$$

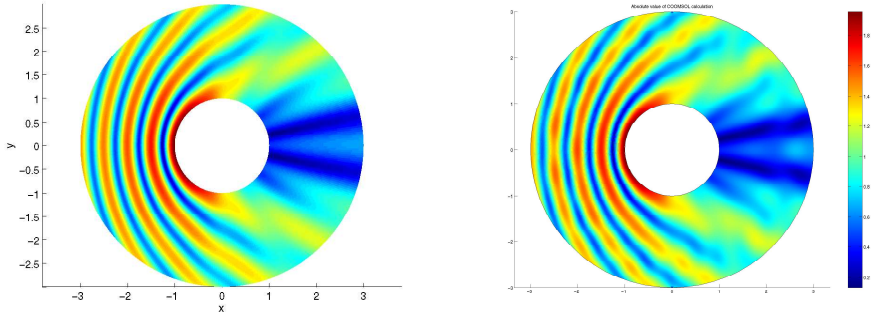
We notice that the cosine term may be equal to zero for certain combinations of θ and m. This may result in the stop criterion being fullfilled prematurely. To avoid this situation we further require that the penultimate term obey a similar condition

$$\left| \frac{\epsilon_{m-1} i^{m-1} \left[J_{m-1}(kr) - \frac{J'_{m-1}(ka)}{H'^{(1)}_{m-1}(ka)} H^{(1)}_{m-1}(kr) \right] \cos\left((m-1)\theta\right)}{\sum_{n=0}^{m} \epsilon_n i^n \left[J_n(kr) - \frac{J'_n(ka)}{H'^{(1)}_n(ka)} H^{(1)}_n(kr) \right] \cos n\theta} \right| < \delta. \tag{3.2.4}$$

The two expressions will never be zero at the same time, ensuring that the calculation does not stop before we want it to. To simplify the implementation we choose $\gamma = \delta$. Since we only calculate a finite summation, we can save all the Bessel and Hankel functions, once they have been calculated and, if possible, reuse them later in the computation. Whether this is possible depends on the actual coordinates, but it will always be possible for the functions with ka as argument, since ka is constant.

The FEM solution is by nature an approximation. It is therefore of great interest to verify the results obtained from COMSOL™. This is done by comparing it with the analytic solution for subdomain I. To compare the analytic and the FEM solution a plot of each of the solutions is made. The analytic solution is shown in Figure 3.6a and the FEM solution in Figure 3.6b. The calculation is done from the center of the cylinder to a radius of $3a$ (in this case 3 since $a = 1$). Although there are small differences, they are most likely a consequence of a relatively low number of elements in the grid in the FEM model. The calculation time is highly dependent on the number of elements in the mesh, so in order to avoid too long calculation times, the number of elements in the grid is set to about 30.000. Apart from the small differences due to the grid in the FEM model the two solutions look identical. If one were to increase the number of elements in the grid to solution would be closer to the analytical solution.

To further investigate the difference in the solutions a line plot is made in the distance 1.5 from the center of the cylinder for both the solutions. These line plots are then subtracted from each other and the difference is plotted. The result is shown in Figure 3.7. If the solutions were identical we would expect a 0 result. The maximum absolute error is about 0.02 resulting in a relative error of about 4% compared to the analytic solution. The difference between the FEM solution and the analytical solution can be reduced by using a larger number of internal points in the FEM model, and precision can thus be increased at the cost of computation time.



(a) *The analytic solution of the wave problem.*

(b) *The finite element solution of the wave problem*

Figure 3.6: *The FEM and the analytic solution look very similar. The small differences are caused by the number of elements in the grid. When the grid is refined the FEM solution will converge to the analytical solution.*

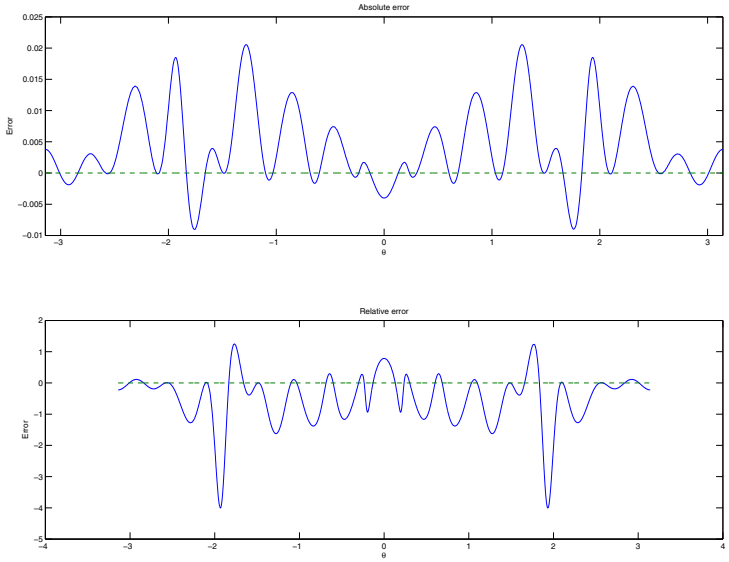


Figure 3.7: *The FEM solution subtracted from the analytic solution at constant $r = 1.5$. The upper graph shows the absolute error, while the lower graphs shows the relative error.*

Chapter 4

Extension to complicated geometries

4.1 The analytical/numerical coupling method

We have now numerically and analytically solved a wave scattering problem, where the scattered wave component is outgoing and the scattering cylinder is assumed to be fully reflecting[1]. To generalize the problem further the latter boundary condition is now removed. This allows an arbitrary object with arbitrary properties regarding for example reflectivity. We shall, however, limit our interest to objects that are mirror symmetric in the x-axis. Further it is assumed that Equation (2.3.22) remains valid far away from the object such that the scattered wave component can be expressed as a linear combination of Hankel functions (we shall discuss this statement later), where we shall allow new values of the coefficients D_n. The problem is depicted on Figure 4.1: We now have the outer subdomain II and the inner circular subdomain I with radius b, and an arbitrary symmetrical object resides within I.

The problem remains of deciding the boundary conditions on the boundary between subdomains I and II in Figure 4.1. We require that the solutions $\eta(I)$ and $\eta(II)$ in subdomains I and II be joined on their common boundary continuously and differentiably. Thus we define the two boundary conditions (still with the ambiguity of selecting the constants D_n), namely (omitting the constant of proportionality α of the previous expressions)

$$\eta(I)(b,\theta) = \eta(II)(b,\theta) = \sum_{n=0}^{+\infty} \left[\epsilon_n i^n J_n(kb) + D_n H_n^{(1)}(kb)\right] \cos n\theta \qquad (4.1.1)$$

$$\left[\frac{\partial \eta(I)}{\partial r}\right]_{r=b} = \left[\frac{\partial \eta(II)}{\partial r}\right]_{r=b} = k\sum_{n=0}^{+\infty} \left[\epsilon_n i^n J_n'(kb) + D_n H_n'^{(1)}(kb)\right] \cos n\theta. \qquad (4.1.2)$$

It is clear that every choice of D_n will result in a perfectly valid problem which COMSOL™ can solve in subdomain I. However there can obviously only be one correct such choice, and COMSOL™ will be able to find these values of D_n by applying certain requirements[2].

[1] See the boundary conditions used to achieve the solution of the wave problem in Chapter 2.

[2] The details of these requirements were kindly supplied by Nils Malm of COMSOL™ support [6]. The requirement is integrated into the COMSOL™ model as a minimization of certain functionals in a way similar to the implementation of Dirichlet boundary conditions. This eventually results in an expansion of the system of linear equations which COMSOL™ maintains internally.

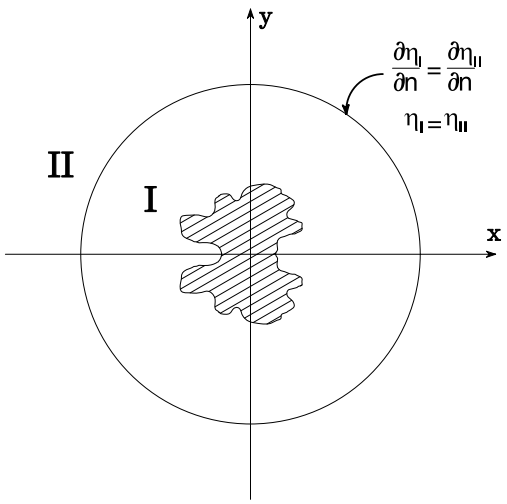


Figure 4.1: *The model is divided into two Subdomains, I and II. The two subdomains are joind continously and differentiably.*

We shall refer to the use of Hankel functions (analytically determined) as a boundary condition to the numerical problem as the *coupling* method since the numerical solution is coupled to the analytical one by means of the Hankel coefficients.

Recall that the analytical solution (2.3.22) was derived with the requirement that the inner boundary was a circle. This is so far still true for region II; but it is uncertain whether a change to non-cylindrical boundary conditions *inside* subdomain I might affect the solution outside, i.e. whether there does at all *exist* a set of constants D_n such that the Equation (2.3.22) can express the exact solution on the boundary. It is clear that if the function space spanned by $H_n^{(1)}(kr)\cos(n\theta)$ is *complete* within the space of outward-propagating (in r) and even (in *theta*) functions, then *any* function can be expressed by such an (infinite) linear combination. Therefore Equation (2.3.22) is unquestionable if the Hankel functions are together complete. If this is *not* the case, a solution using Equation (2.3.22) could include any number of terms and use arbitrary precision, and yet the error would not approach 0. Even though this function space is hardly complete (since all constants are already necessary to express an arbitrary θ-dependence) it will still be plausible that linear combinations of the Hankel functions can provide a good solution numerically, since the problem *still* resembles the cylindrical one in many respects (same differential equation, assumptions, etc.).

Before continuing with the solution we shall consider some efficiency issues. Until now the analytical solution has been evaluated by means of a finite sum (see Section 3.2), and the summation was stopped when terms became sufficiently small compared to the accumulated sum. We would like to minimize the necessary number of terms. Consider Figure 4.2. The figure shows the values of n where the function evaluation terminates as a function of ka. If, for instance, the value ka is kept at around 1, it is necessary to include approximately 8 terms in order for the relative sizes of the remaining terms (compared to the accumulated sum) each to be smaller than 10^{-4}.

Setting $ka = 1$ and $b = 2a$ the coefficients D_n can be calculated using Equation (2.3.23), assuming that the values of the previously considered problem will be at least

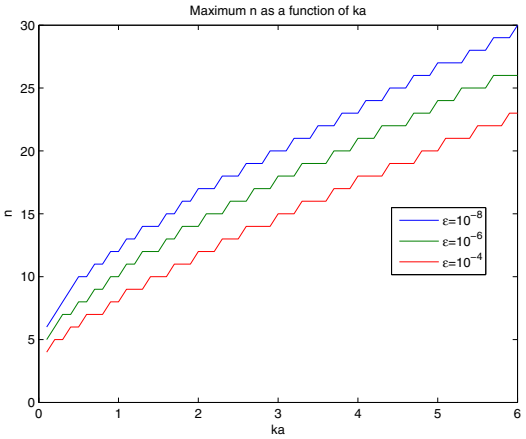


Figure 4.2: *The number of terms, n, as a function of ka for different tolerances.*

n	$\Re(D_n)$	$\Im(D_n)$	$\lvert D_n \rvert$
0	$-2.4087e-01$	$-4.2761e-01$	$4.9079e-01$
1	$-6.5616e-01$	$-2.4538e-01$	$7.0054e-01$
2	$1.3823e-02$	$-1.6570e-01$	$1.6627e-01$
3	$7.1093e-03$	$2.5271e-05$	$7.1093e-03$
4	$-1.1510e-08$	$1.5173e-04$	$1.5173e-04$
5	$-1.9355e-06$	$-1.8731e-12$	$1.9355e-06$
6	$1.3401e-16$	$-1.6371e-08$	$1.6371e-08$
7	$9.8530e-11$	$4.8540e-21$	$9.8530e-11$
8	$-9.8308e-26$	$4.4360e-13$	$4.4360e-13$
9	$-1.5506e-15$	$-1.1940e-30$	$1.5506e-15$

Table 4.1: *D_n for $ka = 1$. Values decay rapidly with n.*

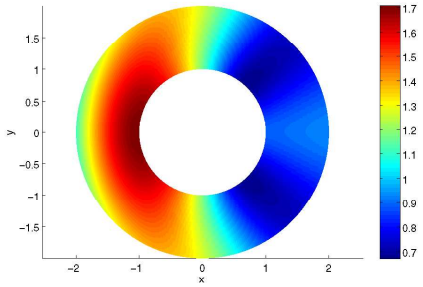
somewhat similar in magnitude to those of the new problem. The first six terms are listed in Table 4.1. Their numerical sizes decrease rapidly with n, and coupled with the fact that the Bessel functions $J_n(1)$ approach zero for increasing n as well ($J_7(1) \approx 10^{-6}$), this shows that we will most likely need only a few terms to evaluate the function η with reasonable precision.

4.2 COMSOL™ solution and validation

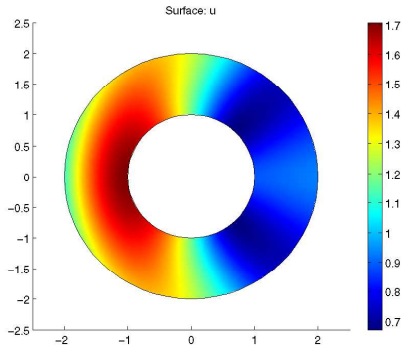
In order to evaluate the credibility of a COMSOL™ solution, we shall now compare values of D_n found by COMSOL™ with those of the analytical solution above. The entire geometry from above is therefore implemented in a COMSOL™ script, and the thus calculated values of the constants D_n are listed in Table 4.2. The last column shows the relative errors of these numerical solutions compared to the values of Table 4.1. The relative error is consistently smaller than 10^{-3} as long as the absolute values of D_n are larger than 10^{-10}, which means that large relative errors only occur when the coefficients are numerically insignificant (at which point this behaviour is expected).

n	$\Re(D_n)$	$\Im(D_n)$	$\lvert D_n\rvert$	$\frac{\lvert D_n\rvert^{\text{numerical}}}{\lvert D_n\rvert^{\text{analytical}}} - 1$
0	-0.24087	-0.42761	0.49079	$1.0080e-05$
1	-0.65616	-0.24538	0.70054	$1.5792e-06$
2	0.013823	-0.1657	0.16627	$-1.3960e-05$
3	0.0071093	$2.5272e-05$	0.0071093	$-5.2317e-06$
4	$-1.1536e-08$	0.00015173	0.00015173	$2.3064e-05$
5	$-1.9355e-06$	$-9.8576e-12$	$1.9355e-06$	$-1.3536e-05$
6	$6.0297e-13$	$-1.6373e-08$	$1.6373e-08$	$1.1794e-04$
7	$9.8075e-11$	$3.7601e-14$	$9.8075e-11$	$-4.6133e-03$
8	$-1.8112e-14$	$5.0556e-13$	$5.0588e-13$	$1.4039e-01$
9	$4.5787e-15$	$-1.2067e-14$	$1.2906e-14$	$7.3232e+00$

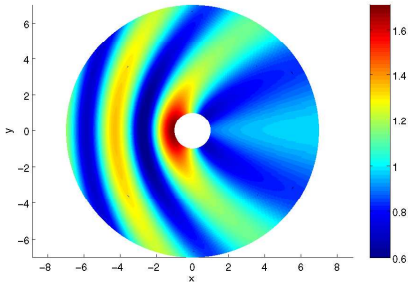
Table 4.2: *The numerically computed coefficients D_n for $ka = 1$. Values decay rapidly with n. The relative errors compared to the values of Table 4.1 are shown in the last column, and they are reasonably low as long as $|D_n|$ is appreciably large.*



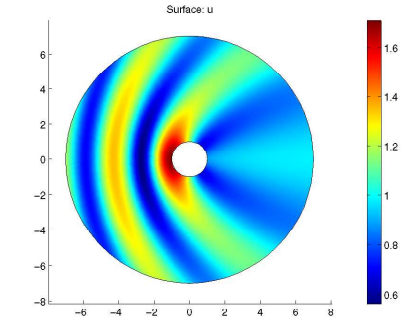
(a) *The analytic solution to the wave problem*



(b) *The numerically coupled solution to the problem.*



(c) *The analytic solution in a larger domain*



(d) *The numerical solution in a larger domain*

Figure 4.3: *The numerically coupled solution is here compared to the analytical solution to the wave problem. It is seen that it resembles the analytic solution very much.*

In Figure 4.3 the numerically coupled solution is compared to the analytical solution. The two plots look very much alike. The conclusion is that the COMSOL™ numerically coupled simulation produces realistic values and therefore correctly approximates the analytical solution.

4.3 Plots of arbitrary geometries

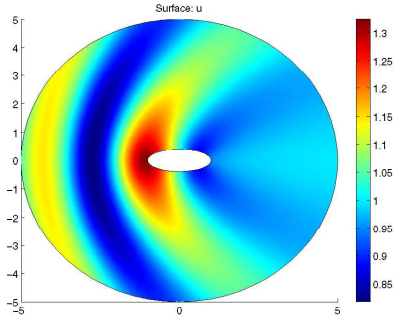
We have now shown how we can use the coupled solution method to solve the wave problem for different geometries as long as the subdomains specific to analytical and numerical solution methods could be joined along a circle. Having thus far considered only cylindrical obstructions we shall move one step further, extending our perspective to *elliptic* obstructions. In this section we shall use a and b to denote the minor and major axes of ellipses, respectively.

In Figure 4.4a and b, two ellipses are considered as obstacles and the wave problem is solved for $k = 1$. As seen the maximum amplitude is in both cases observed on the left side of the ellipse where the incoming wave strikes first. When the ellipse is oriented along the x-axis the amplitude of the wave is slightly above 1.3, see Figure 4.4c and d. Surprisingly when the ellipse is oriented along the y-axis the maximum amplitude is *above* 2 (about 2.1), which means the standing wave analogy mentioned in Section 2.4 breaks (at least partially) down. When the problem was solved with the circle geometry the maximum possible amplitude was 2. Although we cannot validate the result, the solution looks fairly reasonable: when the ellipse is oriented along the x-axis the max amplitude is smaller than when it is along the y-axis. This seems logical since the wave can more easily get around the obstruction in the latter case.

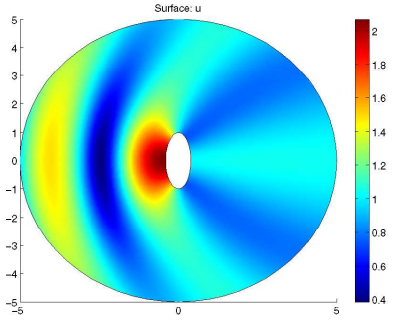
It seems that the maximum amplitude of the hitting wave is increasing when the ellipse is oriented along the y-axis. This also agrees with the previously established notion that long wavelengths (here compared to the size of the obstruction *perpendicular* to the angle of incidence) will not 'see' smaller obstacles. However the fact that the amplitude can exceed that of a standing wave prompts a closer consideration of this phenomenon. In Figure 4.5 ellipses with different eccentricities are used when solving the wave problem. When the major axis, oriented along the y-axis, is increased, the amplitude maximum curiously bifurcates into two maxima. The maximum amplitude is as much as 2.5 in Figure 4.5d. To investigate when this separation of maxima occurs, plots were made along different line segments parallel with the y-axis and tangential to the left side of the ellipse, see Figure 4.6. It seems that the bifurcation occurs when the major axis is about 3.2 and the minor 1.

Variation of wave number

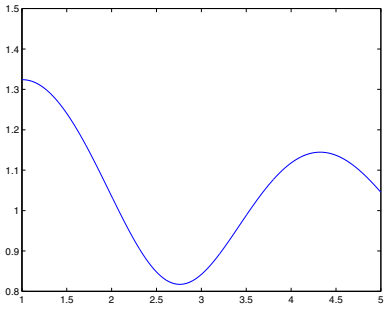
We shall now consider the effect of varying wave numbers using different eccentricities. Figure 4.7 shows two sets of ellipses with major axes oriented along x- and y-axes respectively. Figure 4.8 shows several plots of the amplitude at the front and back of each of these ellipses. The minor axis of the ellipse and the wave number k are then varied in turn. The result is shown in Figure 4.8. In Figure 4.8a, the amplitude was measured at the back of an ellipse with major axis parallel to the x-axis. The figure shows a dropping amplitude when the size of the minor axis is increased. Intuitively this seems correct because an ellipse with a minor axis of 0 would not disturb the waves at all and the amplitude would be one. When the wave number is decreased (wavelength increased) the amplitude increases toward 1. This also seems correct since a larger wavelength will make the ellipse



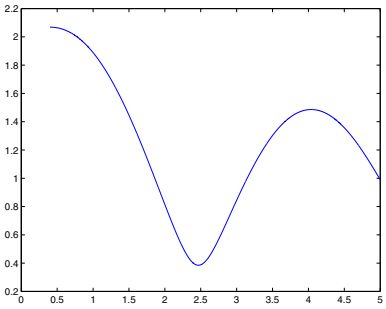
(a) *Solution with an ellipse oriented along the x-axis*



(b) *Solution with an ellipse oriented along the y-axis*



(c) *A 2d line plot of the incoming wave to the maximum of the amplitude for the x oriented ellipse.*



(d) *A 2d line plot of the incoming wave to the maximum of the amplitude for the y oriented ellipse.*

Figure 4.4: *Ellipses oriented along the x and y axes respectively. The wave number is $k = 1$.*

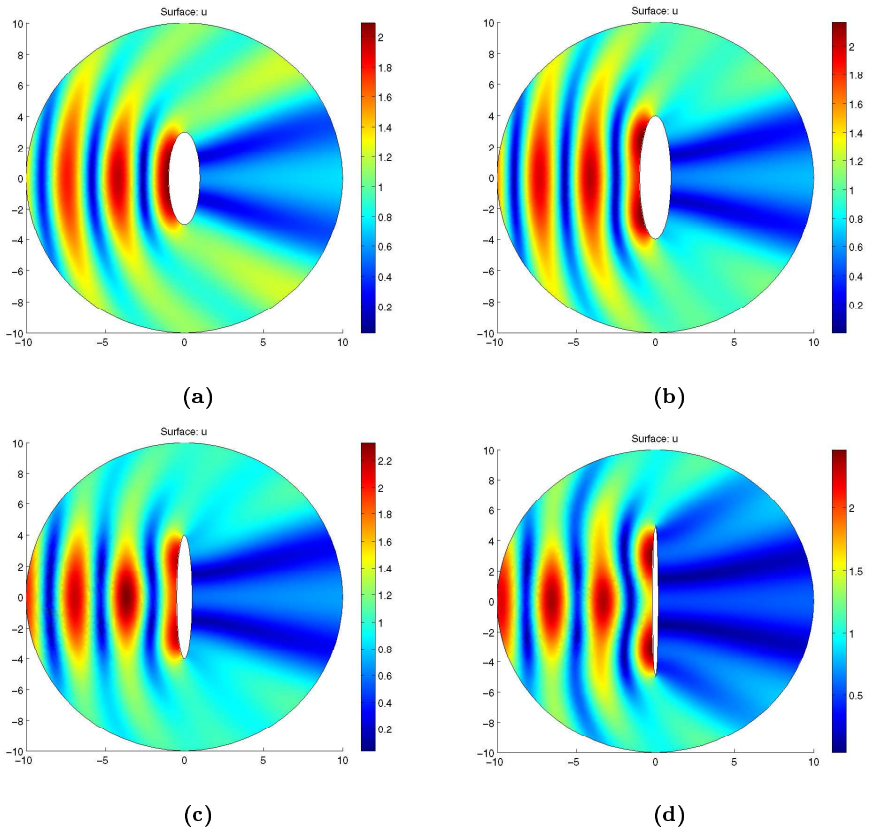


Figure 4.5: *When the ellipse's major axis is increased the maximum amplitude is greater and it splits up into two peaks.*

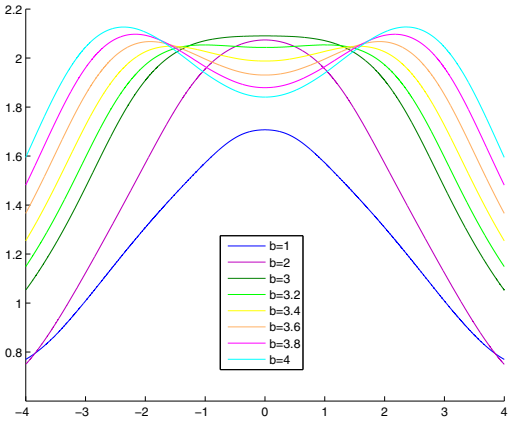
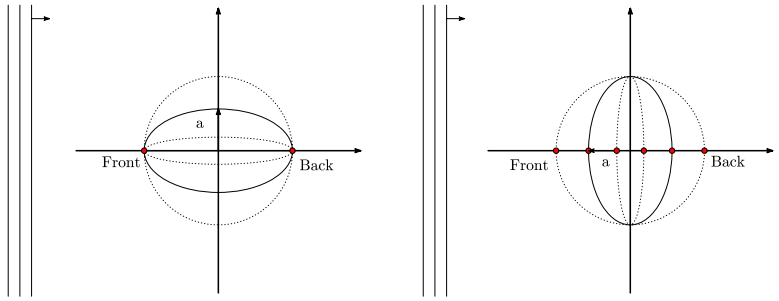


Figure 4.6: *A lineplot at $x = -1$ and $y = -4$ to 4. As seen in Figure 4.5d, the wave moves from one to two maxima when the major axis of the ellipse increases (b). The minor axis is always one. The maximum in front of the ellipse is separated around $b = 3.2$ with $a = 1$*



(a) *Major axis along x-axis* **(b)** *Major axis along y-axis*

Figure 4.7: *Variations of ellipse dimensions*

negligible to the waveform, and thus not create as large a void behind the ellipse (cf. Section 2.4). Note that for $k = 2$ the model is no longer as precise since we do not have enough terms in the numerical solution (the solution has 10 terms making it precise for $k = 1$, with an acuracy of 10^{-4}), see Figure 4.2[3].

In Figure 4.8b we consider the amplitude in front of the same ellipse as before. The figure now shows an increasing amplitude with an increasing minor axis of the ellipse. This also seems correct since an increasing a will make the area projected onto the wavefront larger, making it harder for the waves to move around the ellipse, make the water pile up in front of the ellipse. Again an increasing wavelength will make the ellipse negligible to the wave form, hence the amplitude drops. The plot for $k = 2$ seems to have a maximum when a is around 4, resulting in an amplitude of around 2.2. But again this may not be correct because of the high wave number.

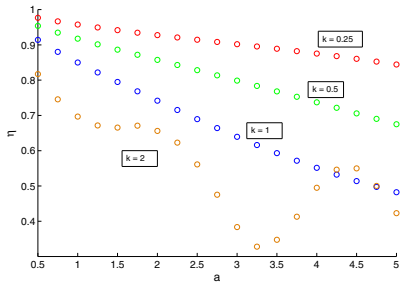
In Figure 4.8c the ellipse has been rotated 90 degrees and now has major axis located parallel to the y-axis. Again the minor axis is varied from 0.5 to 5 and the amplitude is measured at the back of the ellipse. For $k = 0.25$ the amplitude is seen to drop with a, and for $k = 0.5$ the amplitude is rising with a. It might seem surprising that the different wave numbers yield opposite results. Intuitively one would expect the that the amplitude would become larger when the ellipse becomes more circle-like (when a increases). This is clear if you think of the extreme cases where the ellipse becomes a line parallel to the y-axis or a circle. In this case the drag (and void) caused by the line would be greater than that of the circle. However this dependence seems to change with k. Qualitatively however, the amplitude increases with larger wavelength, which is expected.

Figure 4.8d is similar to the previous but now the amplitude is measured in front of the ellipse. For the two k values below unity the amplitude is seen to decrease when a is increased. But surprisingly when $k = 1$ and $k = 2$ the amplitude increases when a is increased until it is about 3, whereafter the amplitude drops slightly. It seem that the four plots all cross in the same point $a \approx 3$. This may be caused by the fact that the maximum seems to split up into two in this point, as seen in Figure 4.5.

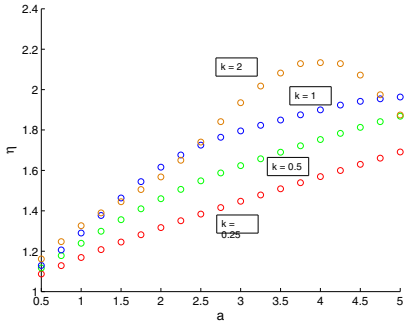
Excessively complicated geometries

So far we have only studied geometries possessing symmetry around the x-axis (i.e. circles and ellipses). In order to generalize the model to more complicated geometries, we have

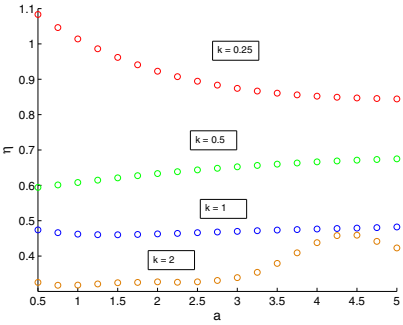
[3]It is easy to implement extra terms, and it is done in simulations later in the report, but we found it too time consuming to repeat the rather heavy computations for this particular case.



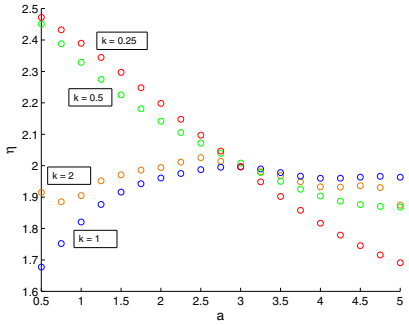
(a) *Amplitude measured at the back of an ellipse with major axis parallel to the x-axis.*



(b) *Amplitude measured at the front of an ellipse with major axis parallel to the x-axis.*



(c) *Amplitude measured at the back of an ellipse with major axis parallel to the y-axis.*



(d) *Amplitude measured at the front of an ellipse with major axis parallel to the y-axis.*

Figure 4.8: *Plots of wave amplitude for varying minor axis a and k, with different geometries. The major axis b of the ellipse is held at a constant value of 5.*

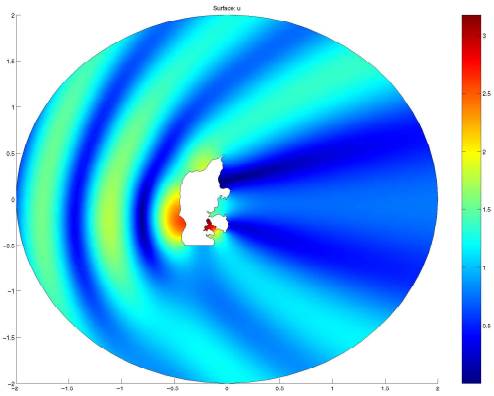


Figure 4.9: *The problem solved with geometry in the shape of Jutland and Funen of Denmark.*

to return to Equation (2.3.9). If the geometry is not mirror symmetric about the x-axis, we cannot neglect the sine term anymore and we have to evaluate the whole expression. We therefore add the sine term to the eqation in the COMSOL™ script[4], and the wave field equation now reads

$$\eta = \sum_{n=0}^{+\infty} \left[G_n H_n^{(1)}(kr)\cos(n\theta) + Z_n H_n^{(1)}(kr)\sin(n\theta) \right] + \exp^{ikx}, \tag{4.3.1}$$

where G_n and Z_n are unknowns to be calculated by COMSOL™. To test the model with more complex arbitrary geometry, maps of Jutland and Iceland are imported as `png` pictures, converted to a 2 dimensional COMSOL™ solid and the problem is solved. In order to use a wavelength comparable with those of real tsunamis[5], k is chosen to be 5, which in the scale of our model is equal to a wavelength comparable to the length of Jutland (400km)[6]. This is of course a rather crude approximation, since we chose the number of terms in the summation for $k = 1$, but as this is just a proof of concept and a low computation time is of greater interest than accuracy we will allow this for now. If needed, the number of terms can be increased until the desired accuracy is obtained.

Comparison and discussion

It is evident that the complicated geometries are not very different to the simple elliptical ones if we consider only differences on length scales comparable to the wavelength. It is therefore plausible that the behaviour in the vicinity of the considered complicated geometries resembles the behaviour in the elliptic cases. The plots on Figures 4.3, 4.9 and 4.10 show that such a simplified interpretation is *not* sufficient. Even though most of the coastal features of Iceland and Jutland are very small, the wave interference results in peaks of wave amplitude at unpredictable locations. However the amplitude behaviour at larger distances from the scattering objects do indeed behave roughly like in the simple

[4] See Appendix A.3

[5] Real tsunamis has a wavelength of several hundreds of kilometres.

[6] $L = \frac{2\pi}{k} \approx 1$, and Jutland, as seen on Figure 4.9, has a length of approximately unity.

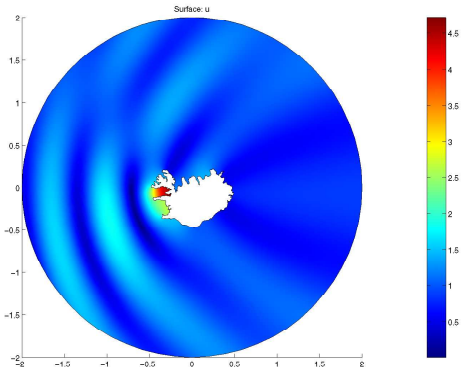


Figure 4.10: *The problem solved with geometry in the shape of Iceland.*

cylindrical or elliptic models (this is of course true on the boundaries because of the conditions used, but also in the intermediary area). We therefore conclude that simple models are sufficient to describe the scattering at large distances from the objects (i.e. at other coasts farther away, for example), whereas such models are, not surprisingly, bad at predicting the extent of damage on the particular coastal areas themselves.

Given that the wave patterns in the case of complicated geometries vary greatly from those seen in the simple models (which are known analytically to be precise), and that the real-world behaviour contains some more complicated variables than those we have implemented here (notably varying water depth and waves flowing onto land instead of being reflected immediately), we shall not state that the presented results are accurate without discussion, but we note that the behaviour at intermediate distances seems reminiscent of that observed previously, and is therefore quite realistic.

4.4 Ocean with appreciable variations in water depths

So far we have assumed that the water depth in our model has been constant. In this section we will remove this restriction and allow appreciable variations in the water depth. In the derivation of the analytical solution, which we use for the boundary conditions, we assume that k is a constant. Since $k = 2\pi(gh)^{-1/2}T^{-1}$ we have to make sure that the depth variations only occur within in the numerical subdomain (subdomain I). We therefore define the depth h as

$$h \equiv \begin{cases} h(x,y) & \text{for } r \leqslant b \\ h_0 & \text{for } r > b \end{cases}, \tag{4.4.1}$$

where h_0 is a constant. If the variations are comparable to the wavelength, an incoming wave will be scattered and can not be described accurately by the model used in this report. We shall thus consider only appreciable variations, i.e. variations should take place on a scale greater than or comparable to the wavelength. With a variable water depth, the spatially dependent equation is, as shown in [5, p. 110], given as

$$\boldsymbol{\nabla} \cdot (h\boldsymbol{\nabla}\eta) + \frac{\omega}{g}\eta = 0, \tag{4.4.2}$$

Figure 4.11: *The ocean floor of the northern part of the Atlantic ocean. For an island as Iceland the ocean floor is almost constant until a certain radius where it suddenly drops quickly to a lower, almost constant, level.*

where g is the constant of gravity[7]. This expression can be reformulated as

$$\nabla \cdot [h(x,y)\nabla\eta(x,y)] + hk^2\eta(x,y) = 0 \tag{4.4.3}$$

$$\nabla \cdot [h(x,y)\nabla\eta(x,y)] + \frac{4\pi^2}{gT^2}\eta(x,y) = 0, \tag{4.4.4}$$

where we, as in Chapter 2, assume that T is constant. If h is a constant, Equation (4.4.3) reduces to the Helmholtz equation (Equation (2.1.3)).

To simplify the implementation in COMSOL™, we will only look at radial variations, i.e. the water depth is independent of θ. This may sound like a rather crude approximation, but as seen on Figure 4.11, the ocean floor around Iceland is almost circular. Based on these observations, what we need to describe the depth variation is a function which has a fixed value close to the scattering object and then decreases slowly, compared to the wavelength, to a another fixed value, lower than the initial value. An example of such a function could be the continous, infinitely differentiable function

$$\rho_{a,b}(r) = \frac{\lambda(b^2 - \|r\|^2)}{\lambda(b^2 - \|r\|^2) + \lambda(\|r\|^2 - a^2)}, \tag{4.4.5}$$

where r is a coordinate vector and λ is defined as

$$\lambda(x) = \begin{cases} \exp\left(-\frac{1}{x^2}\right) & \text{for } x > 0 \\ 0 & \text{for } x \leqslant 0 \end{cases} \tag{4.4.6}$$

This function will take the value 0 for $r \leqslant a$ and the value 1 for $r \geqslant b$ and is infinitely differentiable in between. Hence we can fulfill the above mentioned requirements, that k is constant in subdomain II and that depth variation is "appreciable", by adjusting the parameters a and b. We define the water depth, which is always positive, at a point r, as

$$h(r) = h_{\text{end}} - h_{\text{diff}}\rho_{a,b}(r), \tag{4.4.7}$$

[7] 9.82m/s^2 in Denmark

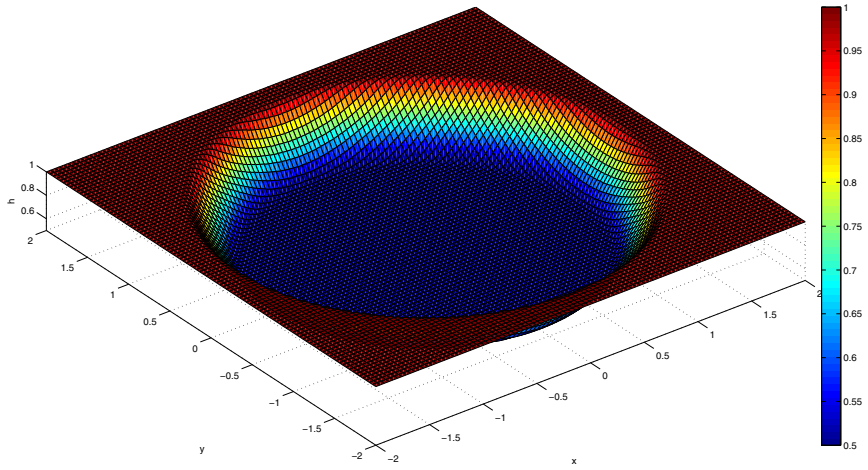


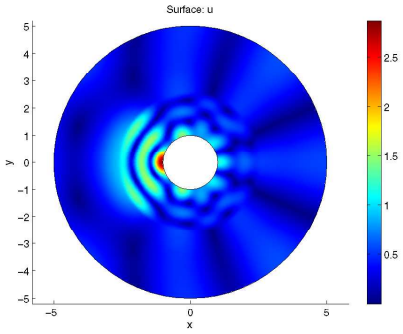
Figure 4.12: *The variation in h, as a function of r for $a = 1.2$, $b = 1.9$, $h_{diff} = 0.5$ and $h_{end} = 1$. Note that the function is constant 0.5 for small r and constant 1 for larger r.*

where $h_{\text{end}} - h_{\text{diff}}$ is the depth when $r \leqslant a$ and h_{end} is the depth when $r \geqslant b$. If we have the border between subdomain I and II at $r = 2$ we could then choose $a = 1.2$ and $b = 1.9$ and set $h_{\text{end}} = 1$ and $h_{\text{diff}} = 0.5$, or

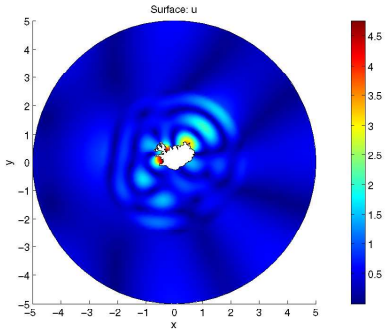
$$h(r) = 1 - 0.5\rho_{1.2,1.9}(r), \tag{4.4.8}$$

resulting in an ocean floor as shown on Figure 4.12. So far we have only had 10 terms in the sum in the analytical expression, restricting us to wavelengths less than 2π ($k = 1$). If subdomain I has the radius 2, we will then only see approximately one sixth of a full wave, and it will be difficult to observe the change in water depth, which at the same time has to be very small compared to the wavelength. We could alternatively just increase the radius of subdomain I, but that would not just require more terms in the summation[8], but also more elements in the mesh, thereby increasing computation time. We will therefore have to increase the number of terms in the summation. If we look at Figure 4.2 we see that atleast 23 terms are required if we want $ka = 6$ and still have δ and γ less than 10^{-4}. $ka = 6$ corresponds to a wavelength of $L = 1.05$, and considering that adding 13 extra terms, that is 26 extra unknowns associated with sine and cosine terms, significantly increases the computationtime, we will have to settle with this wavelength. On Figure 4.13 the result of simulations with depths ranging from $h = 0.2$ at $r = 2.5$ to $h = 5$ at $r = 3.5$ is shown. The wavelength is related to the ocean depth as $L = Tc = T\sqrt{gh}$, hence

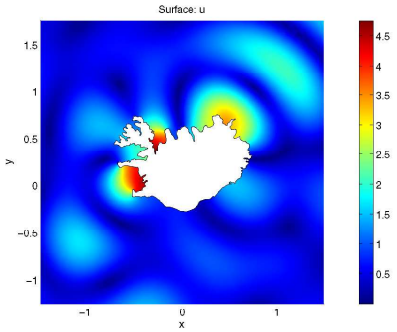
[8]Points far from structure requires more terms than points close to the structure.



(a) *Diffraction for a plane wave incident on a cylinder.*



(b) *Diffraction for a plane wave incident on Iceland.*



(c) *Closeup of the plane wave incident on Iceland.*

Figure 4.13: *The depth h increases from 0.2 to 5 in the interval $r = 2.5$ to $r = 3.5$ corresponding to a change in wavenumber from ≈ 4.5 to ≈ 0.9.*

at lower depths, the wavelength decreases according to the shallow water model. This is observed on Figure 4.13, where the wavelength is relatively large far from the structure, at large depths, compared to the wavelength close to the structure, at lower depths. The wave amplitude increases substantially near the coast, which is also in agreement with common knowledge.

Having thus taken into account arbitrary shapes, possibly lacking symmetry, and having introduced variable water depth, we shall regard the model in its present state as complete for the sake of this project.

Chapter 5

Conclusion

The purpose of this project is to combine numerical and analytical methods to a one step approach for solving a PDE problem. We develop the method by studying the wave interactions that are observed when ordinary water waves scatter on obstructions within the *shallow-water* domain.

These numerical calculations can be used to predict or simulate the propagation of tsunami waves and the contact with geometries such as coastlines. As mentioned in the introduction this became a relevant problem with the 2004 tsunami, where the high number of casualties partly was caused by a lack of effective warning systems.

This mathematical problem is approached by initially formulating a simple symmetric model using a harmonic wave incident on a cylinder, then deriving an analytical solution in the form of a Fourier series. Particularly the amplitude as a function of location has been of interest. It is shown that for wavelengths much larger than cylinder radius ($ka \ll 1$) the scattering is insignificant. For smaller wavelengths, i.e. for ka around unity or greater, the wave field will increasingly behave like a standing wave near the point of incidence of the cylinder, whereas a wave valley will appear on the opposite side.

Following this, the same problem is solved numerically in a COMSOL™/MATLAB™ environment using the analytical solution as a boundary condition. The proposed solution method involves expressing the solution on the boundary as a linear combination of the basis functions of the analytical solution, then using COMSOL™ to determine the optimal values of the coefficients. The results obtained are verified by comparison of the numerically calculated Fourier coefficients to the analytically determined ones, see Table 4.1.
This method of analytical/numerical *coupling* is now extended to do calculations on more complex geometries: the analytical solution is used in the far field and as a boundary condition for the numerical problem (still with modification of coefficients from the series expansion). In the vicinity of the complex scattering object, where the analytical solution would become unobtainable, the numerical solution can still be evaluated, thus exploiting the strengths of both solutions.
The problem is now extended to consider more complicated shapes, notably ellipses, and the scattering behaviour is studied and compared to the cylindrical case. It is observed that in some cases the amplitude at some locations increases *beyond* that of a standing wave, and for high eccentricity the interference causes amplitude extrema to *bifurcate* (i.e. split up). Otherwise the behaviour largely resembles that of the cylinder.

Next, the problem is further extended to consider non-symmetric models of high complexity, specifically using maps of Denmark and Iceland as scattering objects. In these cases the amplitude exhibits *very high* peaks in some areas, mostly in bays, which is surprising because the sizes of these features are much smaller than the wavelength. The simple elliptic models can therefore not be used to predict amounts of damage on the shore, but it is also observed that the wave amplitude behaves reasonably closely to that in the elliptic models at intermediate distance from the shore, meaning that the coastal features are insignificant far from the scattering objects (which is also inuitively plausible).

Finally, having above considered only cases of constant ocean depth, the model is modified to take into account variable depth. The Icelandic problem is modified by introducing a shallow circular *shelf* area around the boundary and a smooth transition to a deep *abyssal plain*. The wavenumber changes greatly as waves approach the coast, and this alters the amplitude and locations of peaks considerably.

The model is now capable of taking into account most of the important factors that determine the behaviour of tsunami waves. The accuracy can still be improved by including real ocean floor data. It might also be possible to alter the model, such that it can take into account more steep slopes of the ocean floor.

Bibliography

[1] Partial Differential Equations with Fourier Series and Boundary Value Problems (2nd edition), Nakhlé Asmar, Pearson Printice Hall, 2005. ISBN 0-13-148096-0.

[2] Wikipedia article on tsunamis, `http://en.wikipedia.org/wiki/Tsunami`

[3] An introduction to hydrodynamics and water waves, Bernard Le Méhauté, Springer, 1976. ISBN 0-387-07232-2.

[4] Bessel Functions for Engineers (2nd Edition), N.W. McLachlan, Oxford University Press, 1955.

[5] The Applied Dynamics of Ocean Surface Waves 2nd printing, Chiang C. Mei, World Scientific Publishing Co. Pte. Ltd., 1994. ISBN 9971-50-773-0.

[6] Nils Malm, COMSOL™ AB support, Sweden. `support@comsol.com`

[7] Wikipedia article on Krakatoa, `http://en.wikipedia.org/wiki/Krakatoa`

[8] Wikipedia article on the 2004 Indian Ocean earthquake (Sumatra-Andaman earthquake), `http://en.wikipedia.org/wiki/Indian_ocean_earthquake`

[9] Introduction to Optics, Frank L. Pedrotti and Leno S. Pedrotti, Prentice Hall, 2006. ISBN 0-131-97133-6.

Appendix A

MATLAB™ & COMSOL™ code

99 little bugs in the code
99 little bugs...
fix one bug, compile it again
101 little bugs in the code...

Taken from the *fortune* program

A.1 comsolmodel.m

```
%Comsol model of cylinder in infinite ocean. The geometry consists of a 2d "dougnout". The thickness is equal to the radius of the cylinder. The boundaryconditions are given by numerical evaluation of an analytic expressen in the function etasolution. This is a decoupled model.

clear all
flclear fem

% COMSOL version
clear vrsn
vrsn.name = 'COMSOL 3.2';
vrsn.ext = 'a';
vrsn.major = 0;
vrsn.build = 300;
vrsn.rcs = '$Name:  $';
vrsn.date = '$Date: 2005/12/20 19:02:30 $';
fem.version = vrsn;

% Constants. A is a scalingfactor, k is the wavenumber, tol decides how many terms to compute in the boundaryconditions and a is the radius of the cylinder.
fem.const = {'A','1', ...
  'k','2*pi', ...
  'tol','1e-8', ...
  'a','1'};

% Global expressions. Function for evaluation of the boundarycondition function (from the analytic expression).
fem.globalexpr = {'etadiag','etasolutiondiag(sqrt(x.^2+y^2),atan2(y,x),k,a)'};

% Geometry. Make two circles and subtract the second from the first to make a "doughnut".
clear draw
g1=circ2(3);
g2=circ2(1);
g4=g1-g2;
draw.s.objs = {g4};
draw.s.name = {'CO1'};
draw.s.tags = {'g4'};
```

```
fem.draw = draw;
fem.geom = geomcsg(fem);

% Initialize mesh
fem.mesh=meshinit(fem);
%fem.mesh=meshinit(fem, ...
%                  'hmaxfact',0.8, ...
%                  'hgrad',1.2, ...
%                  'hcurve',0.25, ...
%                  'hcutoff',0.0003, ...
%                  'hpnt',1, ...
%                  'xscale',1, ...
%                  'yscale',1, ...
%                  'hgradedg',[3,1,4,1,6,1,7,1], ...
%                  'hcurveedg',[3,1,4,1,6,1,7,1], ...
%                  'hmaxedg',[3,0.1,4,0.1,6,0.1,7,0.1], ...
%                  'hcutoffedg',[3,1,4,1,6,1,7,1]);

% Refine mesh
%fem.mesh=meshrefine(fem, ...
%                    'mcase',0, ...
%                    'rmethod','regular');
%fem.mesh=meshrefine(fem, ...
%                    'mcase',0, ...
%                    'rmethod','regular');
%fem.mesh=meshrefine(fem, ...
%                    'mcase',0, ...
%                    'rmethod','regular');
%figure
%meshplot(fem)
% (Default values are not included)

% Application mode 1
clear appl
appl.mode.class = 'Helmholtz';
appl.assignsuffix = '_hzeq';
clear equ
equ.f = 0;
equ.a = 'k^2';
equ.c = -1;
appl.equ = equ;

clear bnd
bnd.ind = [1,1,2,2,1,2,2,1];
%figure
%The coefficients for the boundaryconditions are given as: h*u=r for Dirichlet and n*(
    c*nabla u)+q*u=g for Neumann. We have dirichlet at the outer boundary and neumann
    on the inner.
bnd.type = {'dir','neu'};
bnd.r ={'etadiag','0'};
bnd.g = {0,0};
bnd.h = {1,0};
appl.bnd = bnd;
fem.appl{1} = appl;
fem.frame = {'ref'};
fem.border = 1;
fem.units = 'SI';

% Multiphysics
fem=multiphysics(fem);

% Extend mesh
fem.xmesh=meshextend(fem);

% Solve problem
fem.sol=femlin(fem, ...
               'solcomp',{'u'}, ...
               'outcomp',{'u'});

% Here we make a hack in order to get the complex part of the boundaryconditions,
    which is lost in comsol, by multiplying with i in the boundaryconditions function
    (here called "thehack")
femcomplex=fem;
femcomplex.bnd.r ={'thehack(x,y,k,a)','0'};
```

```
103
104 % Extend mesh and solve the hacked problem
105 femcomplex.xmesh=meshextend(femcomplex);
106 femcomplex.sol=femlin(femcomplex, ...
107                 'solcomp',{'u'}, ...
108                 'outcomp',{'u'});
109 %Add the original (real) problem to the hacked complex.
110 solu = femcomplex.sol.u(:)*i + fem.sol.u(:);
111 fem.sol = asseminit(fem,'init',solu);
112
113 %Plot solution.
114 figure,hold on
115 postplot(fem, ...
116          'tridata',{'abs(u)','cont','internal'}, ...
117          'trimap','jet(1024)', ...
118          'title','Absolute value of COOMSOL calculation')%, ...
119          %'axis',[-3,3,-2,2.2,-1,1]);
120 %postplot(fem,'contdata','abs(u)')
121 %postplot(fem,'tridata','u','contdata','u.*x','triz','u','contz','u');
122 %postint(fem,'u')
123
124 theta=-pi:0.001:pi;
125 r=1.5*ones(1,length(theta));
126 postvals=postinterp(fem,'u',[(r.*cos(theta));(r.*sin(theta))]);
127
128 r=1.5;
129 theta=-pi:0.001:pi;
130 sol=etasolutionabs(r,theta,2*pi,1,1e-9);
131 subplot(2,1,1)
132 plot(theta,sol-abs(postvals),theta,zeros(1,length(theta)),'--');
133 title('Absolute error')
134 xlabel('\theta')
135 ylabel('Error')
136 axis([-pi pi -0.01 0.025])
137 subplot(2,1,2)
138 plot(theta,100*(abs(postvals)-sol)./sol,theta,zeros(1,length(theta)),'--');
139 title('Relative error')
140 xlabel('\theta')
141 ylabel('Error')
142 %axis([-pi pi -5 0.025])
```

A.2 etasolutioncomplex.m

```
1  function wave=etasolutioncomplex(r,theta,k,a,tol)
2  %This function evaluates the analytical solution in polar coordinates given as 'r' and
       'theta' to a given tolerance, 'tol', and returns the complex result. The
      wavenumber and the radius of the structure is given by 'k' and 'a', respectively.
      The function has been highly optimized, by tabulating the besselfunctions, and
      bypassing standard Matlab checks.
3  if nargin<5
4  tol=1e-7;
5  end
6  A=1;
7  %Initialize.
8  A=A(1);
9  a=a(1);
10 k=k(1);
11 tol=tol(1);
12 ka=k*a;
13 count=1;
14 eta=0;
15 %wave=zeros(1,length(r)*length(theta));
16 jdiff0=(-ka.*besselmx(real('J'),1,ka,0))./ka;
17 hdiff0=(-ka.*besselmx(real('H'),1,ka,0))./ka;
18 pmax=0;
19 %Loop over r then theta.
20 for g=1:length(r)
21     if abs(r(g))>=1
22         kr=k.*r(g);
23         bjkr0=besselmx(real('J'),0,kr,0);
24         bhkr0=besselmx(real('H'),0,kr,0);
25         qmax=0;
```

```
        for d=1:length(theta)
            % Calculation of wave field in polar coordinates.
            %Calculate first step (where E=1).
            j=0;
            E=1;
            eta=A.*E.*i.^j.*(bjkr0-bhkr0.*(jdiff0./hdiff0)).*cos(j.*theta(d));
            lastterm(count)=eta;
            E=2;
            gamma=1;
            delta=1;
            %Calculate rest of the sum.
            while gamma>tol & delta>tol
                j=j+1;
                if j>pmax
                    jdiff(j,:)=(-ka.*besselmx(real('J'),j+1,ka,0)+j.*besselmx(real('J'
                        ),j,ka,0))./ka;
                    hdiff(j,:)=(-ka.*besselmx(real('H'),j+1,ka,0)+j.*besselmx(real('H'
                        ),j,ka,0))./ka;
                    pmax=pmax+1;
                end
                if j>qmax
                    bjkr(j,:)=besselmx(real('J'),j,kr,0);
                    bhkr(j,:)=besselmx(real('H'),j,kr,0);
                    qmax=qmax+1;
                end
                count=count+1;
                lastterm(count)=A.*E.*i.^j.*(bjkr(j)-bhkr(j).*(jdiff(j)./hdiff(j))).*
                    cos(j*theta(d));
                eta=eta+lastterm(count);
                gamma=abs(lastterm(count))./abs(eta);
                delta=abs(lastterm(count-1))./abs(eta);
            end
            wave(g,d)=eta;
        end
    else
        wave(g,1:length(theta))=3;
    end
    g
end
```

A.3 coupledmodel.m

```
function [fem0,c]=coupledmodel(kortdata,abvector,hvector)
%Written by Ask H. Larsen, Martin F. Laursen and Kasper Reck 2006
%
%Solves Helmholtz's equation outside a domain defined by the argument. If the argument
     'kortdata' is not given, a circle of radius 1 is used. 'abvector' defines the a
    and b in the rho function and 'hvector' the start and end depth as [a b] and [
    hstart hend] respectively. To reduce computationtime, terms in the eta and deta
    function may be removed, at the cost of accuracy. The picturfile should be black &
     white png format, but other formats may work too.
%
%Example: 'coupledmodel('iceland.png',[2.5 3.5],[0.2 5])'
%
%This example reads the file iceland.png and creates a geometry based on the
    information contained in this file. Furthermore it sets the ocean depth to vary
    between 0.2 and 5 in the interval 2.5 to 3.5. The geometry, mesh and solution is
    plotted.

flclear fem
%Start timer
tic
% COMSOL version
clear vrsn
vrsn.name = 'COMSOL␣3.2';
vrsn.ext = 'a';
vrsn.major = 0;
vrsn.build = 300;
vrsn.rcs = '$Name:␣␣$';
vrsn.date = '$Date:␣2005/12/20␣19:02:30␣$';
fem.version = vrsn;

```

```
% Geometry
disp('Converting visual data...')
clear draw
g1=circ2(10);
if nargin==3
kort=imread(kortdata);
kort=fliplr(rot90(rot90(kort(:,:,1))));
figure
image(kort);
colormap gray
[c,lr]=flim2curve(kort,{[],[0 255]}); %resolution is 255
%c=mirror(c,[0 1],[0 1]);
disp('Preparing geometry...')
c=solid2(c);
nvertices=geom1get(c,'nv')
c=scale(move(c,-0.5*length(kort),-0.5*length(kort)),1.5/length(kort),1.5/length(kort))
    ; %translate and scale
g4=g1-c;
else
g4=g1-circ2(1);
end
geomplot(g4,'Pointmode','off','edgelabels','on')
draw.s.objs = {g4};
draw.s.name = {'C01'};
draw.s.tags = {'g4'};
fem.draw = draw;
fem.geom = geomcsg(fem);

% Constants
fem.const = {'a',abvector(1),'b',abvector(2),'g',9.82,'T',1,'hdiff',hvector(2)-hvector
    (1),'hend',hvector(2)};

%Global expressions
fem.globalexpr={'cinfinity','cinfinity(b^2-(x^2+y^2))./(cinfinity(b^2-(x^2+y^2))+
    cinfinity((x^2+y^2)-a^2))'}; %
% Initialize mesh
fem.mesh=meshinit(fem);
%Refine mesh
fem.mesh=meshrefine(fem, ...
                    'mcase',0, ...
                    'rmethod','regular');
fem.mesh=meshrefine(fem, ...
                    'mcase',0, ...
                    'rmethod','regular');
%fem.mesh=meshrefine(fem, ...
%                     'mcase',0, ...
%                     'rmethod','regular');
hold on
meshplot(fem.mesh)
fem.mesh
% (Default values are not included)

% Application mode 1
clear appl
appl.mode.class = 'Helmholtz';
appl.assignsuffix = '_hzeq';
clear bnd
bnd.type = 'neu';

if nargin==3
bnd.ind = [ones(1,nvertices+4)];
else
bnd.ind = [1,1,1,1,1,1,1,1];
end
appl.bnd = bnd;
clear equ
equ.a = '(2*pi)^2/(g*T^2)';
equ.c = '-hxy';
equ.f = 0;
equ.ind = [1];
appl.equ = equ;
fem.appl{1} = appl;

% Application mode 2
```

```
clear appl
appl.mode.class = 'FlPDEWBoundary';
appl.dim = {'lm','lm_t'};
appl.assignsuffix = '_wb';
clear prop
clear weakconstr
weakconstr.value = 'off';
weakconstr.dim = {'lm2','lm3'};
prop.weakconstr = weakconstr;
appl.prop = prop;
clear bnd
bnd.weak = {'+(lm_test*(u-eta)+lm*(u_test-eta_test))-eta_test*deta',0};
bnd.usage = {1,0};
if nargin==3
bnd.ind = [1,1,1,1,2*ones(1,nvertices)];
else
bnd.ind = [1,1,2,2,1,2,2,1];
end
appl.bnd = bnd;
fem.appl{2} = appl;
fem.frame = {'ref'};
fem.border = 1;
fem.units = 'SI';
disp('Analyzing␣mathematical␣expressions...')
% Scalar expressions
fem.expr = {'r','sqrt(x^2+y^2)', ...
  'theta','atan2(y,x)', ...
  'hxy','hend-hdiff*cinfinity',...
  'k','2*pi/(sqrt(g*hxy)*T)', ...
   'eta',    'g0*etacos(0,k,theta,r)+g1*etacos(1,k,theta,r)+g2*etacos(2,k,theta,r)+g3*
        etacos(3,k,theta,r)+g4*etacos(4,k,theta,r)+g5*etacos(5,k,theta,r)+g6*etacos(6,k
        ,theta,r)+g7*etacos(7,k,theta,r)+g8*etacos(8,k,theta,r)+g9*etacos(9,k,theta,r)+
        g10*etacos(10,k,theta,r)+g11*etacos(11,k,theta,r)+g12*etacos(12,k,theta,r)+g13*
        etacos(13,k,theta,r)+g14*etacos(14,k,theta,r)+g15*etacos(15,k,theta,r)+g16*
        etacos(16,k,theta,r)+g17*etacos(17,k,theta,r)+g18*etacos(18,k,theta,r)+g19*
        etacos(19,k,theta,r)+g20*etacos(20,k,theta,r)+g21*etacos(21,k,theta,r)+g22*
        etacos(22,k,theta,r)+z1*etasin(1,k,theta,r)+z2*etasin(2,k,theta,r)+z3*etasin(3,
        k,theta,r)+z4*etasin(4,k,theta,r)+z5*etasin(5,k,theta,r)+z6*etasin(6,k,theta,r)
        +z7*etasin(7,k,theta,r)+z8*etasin(8,k,theta,r)+z9*etasin(9,k,theta,r)+z10*
        etasin(10,k,theta,r)+z11*etasin(11,k,theta,r)+z12*etasin(12,k,theta,r)+z13*
        etasin(13,k,theta,r)+z14*etasin(14,k,theta,r)+z15*etasin(15,k,theta,r)+z16*
        etasin(16,k,theta,r)+z17*etasin(17,k,theta,r)+z18*etasin(18,k,theta,r)+z19*
        etasin(19,k,theta,r)+z20*etasin(20,k,theta,r)+z21*etasin(21,k,theta,r)+z22*
        etasin(22,k,theta,r)+exp(i*k*x)', ...
'deta','g0*diff(etacos(0,k,theta,r),r)+g1*diff(etacos(1,k,theta,r),r)+g2*diff(etacos
      (2,k,theta,r),r)+g3*diff(etacos(3,k,theta,r),r)+g4*diff(etacos(4,k,theta,r),r)+g5*
      diff(etacos(5,k,theta,r),r)+g6*diff(etacos(6,k,theta,r),r)+g7*diff(etacos(7,k,
      theta,r),r)+g8*diff(etacos(8,k,theta,r),r)+g9*diff(etacos(9,k,theta,r),r)+g10*diff
      (etacos(10,k,theta,r),r)+g11*diff(etacos(11,k,theta,r),r)+g12*diff(etacos(12,k,
      theta,r),r)+g13*diff(etacos(13,k,theta,r),r)+g14*diff(etacos(14,k,theta,r),r)+g15*
      diff(etacos(15,k,theta,r),r)+g16*diff(etacos(16,k,theta,r),r)+g17*diff(etacos(17,k
      ,theta,r),r)+g18*diff(etacos(18,k,theta,r),r)+g19*diff(etacos(19,k,theta,r),r)+g20
      *diff(etacos(20,k,theta,r),r)+g21*diff(etacos(21,k,theta,r),r)+g22*diff(etacos(22,
      k,theta,r),r)+z1*diff(etasin(1,k,theta,r),r)+z2*diff(etasin(2,k,theta,r),r)+z3*
      diff(etasin(3,k,theta,r),r)+z4*diff(etasin(4,k,theta,r),r)+z5*diff(etasin(5,k,
      theta,r),r)+z6*diff(etasin(6,k,theta,r),r)+z7*diff(etasin(7,k,theta,r),r)+z8*diff(
      etasin(8,k,theta,r),r)+z9*diff(etasin(9,k,theta,r),r)+z10*diff(etasin(10,k,theta,r
      ),r)+z11*diff(etasin(11,k,theta,r),r)+z12*diff(etasin(12,k,theta,r),r)+z13*diff(
      etasin(13,k,theta,r),r)+z14*diff(etasin(14,k,theta,r),r)+z15*diff(etasin(15,k,
      theta,r),r)+z16*diff(etasin(16,k,theta,r),r)+z17*diff(etasin(17,k,theta,r),r)+z18*
      diff(etasin(18,k,theta,r),r)+z19*diff(etasin(19,k,theta,r),r)+z20*diff(etasin(20,k
      ,theta,r),r)+z21*diff(etasin(21,k,theta,r),r)+z22*diff(etasin(22,k,theta,r),r)+i*k
      *cos(theta)*exp(i*k*x)'};

% Functions
clear fcns
%cosine part
fcns{1}.type='inline';
fcns{1}.name='etacos(n,k,theta,r)';
fcns{1}.expr='besselh(n,1,k*r)*cos(n*theta)';
fcns{1}.dexpr={'0','0','0', ...
  '(-k*besselh(n+1,1,k*r)+n/r*besselh(n,1,k*r))*cos(n*theta)'};
fcns{1}.complex='true';

```

```
%sine part
fcns{2}.type='inline';
fcns{2}.name='etasin(n,k,theta,r)';
fcns{2}.expr='besselh(n,1,k*r)*sin(n*theta)';
fcns{2}.dexpr={'0','0','0', ...
  '(-k*besselh(n+1,1,k*r)+n/r*besselh(n,1,k*r))*sin(n*theta)'};
fcns{2}.complex='true';

fem.functions = fcns;

% ODE Settings
clear ode
ode.dim={'g0','g1','g2','g3','g4','g5','g6','g7','g8','g9','g10','g11','g12','g13','
    g14','g15','g16','g17','g18','g19','g20','g21','g22','z1','z2','z3','z4','z5','z6'
    ,'z7','z8','z9','z10','z11','z12','z13','z14','z15','z16','z17','z18','z19','z20',
    'z21','z22'};
ode.f={'0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0'
    ,'0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','
    0','0','0','0','0'};
ode.init={'0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0',
    '0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0
    ','0','0','0','0','0'};
ode.dinit={'0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0'
    ,'0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','
    0','0','0','0','0'};
fem.ode=ode;
% Multiphysics
fem=multiphysics(fem);

% Extend mesh
fem.xmesh=meshextend(fem);

% Solve problem
disp('Solving␣complex␣differential␣equations...Please␣wait')
fem.sol=femlin(fem, ...
               'complexfun','on', ...
               'solcomp',{'g0','g1','g2','g3','g4','g5','g6','g7','g8','g9','g10','g11
                   ','g12','g13','g14','g15','g16','g17','g18','g19','g20','g21','g22'
                   ,'z1','z2','z3','z4','z5','z6','z7','z8','z9','z10','z11','z12','
                   z13','z14','z15','z16','z17','z18','z19','z20','z21','z22','lm','u'
                   }, ...
               'outcomp',{'g0','g1','g2','g3','g4','g5','g6','g7','g8','g9','g10','g11
                   ','g12','g13','g14','g15','g16','g17','g18','g19','g20','g21','g22'
                   ,'z1','z2','z3','z4','z5','z6','z7','z8','z9','z10','z11','z12','
                   z13','z14','z15','z16','z17','z18','z19','z20','z21','z22','lm','u'
                   });

% Save current fem structure for restart purposes
fem0=fem;

% Plot solution
figure
postplot(fem, ...
         'tridata',{'abs(u)','cont','internal'}, ...
         'trimap','jet(1024)', ...
         'title','Surface:␣u', ...
         'axis',[-5.7956416464891,5.7956416464891,-2.2,2.2,-1,1]);
%Save solution
u0=fem.sol.u(:);

disp('Extracting␣meaningless␣coefficients...')
%Initialize the system with zeroes except the ODE variables
fem.equ.init={'0'};
fem.ode.init={'1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','
    17','18','19','20','21','22','23','24','25','26','27','28','29','30','31','32','33
    ','34','35','36','37','38','39','40','41','42','43','44','45'};

%Extend mesh
fem.xmesh=meshextend(fem);

%Initialize the solution vector
fem.sol=asseminit(fem,'out','u');

% Locate the position of the ODE variables and display their value.
```

```
disp('Numerical G_n from Comsol')
for i = 1:length(fem.ode.init)
eta_n_index(i) = find(fem.sol == i);
disp(['coefficient' num2str(i-1) ': ' num2str(u0(eta_n_index(i))) ...
', abs: ' num2str(abs(u0(eta_n_index(i))))])
end
% Insert the true solution back into the fem struct
fem.sol = asseminit(fem,'init',u0);

%disp('Computing ground zero...')
%j=1;
%for i=5:nvertices-4
%    j=j+1
%    I(j)=postint(fem,'u', ...
%           'dl',[i], ...
%           'edim',1);
%end
%[e,k]=max(I)
%disp(['Casualties: '  num2str(round(abs(e)*1000000))])
toc
```