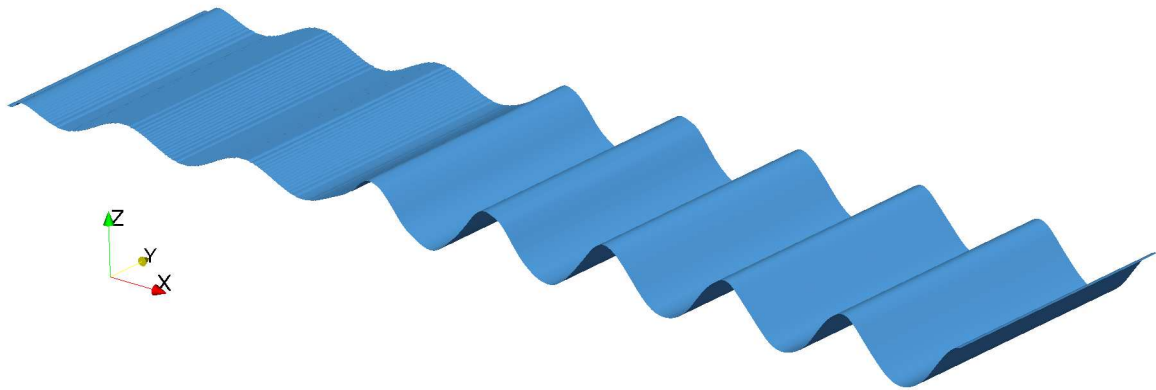


CHALMERS



Numerical Wave Generation In OpenFOAM[®]

Master of Science Thesis

MOSTAFA AMINI AFSHAR

Department of Shipping and Marine Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2010
Report No. X-10/252

A THESIS FOR THE DEGREE OF MASTER OF SCIENCE

Numerical Wave Generation In OpenFOAM® ¹

MOSTAFA AMINI AFSHAR



Department of Shipping and Marine Technology
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2010

¹ This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks.

Numerical Wave Generation In OpenFOAM®

MOSTAFA AMINI AFSHAR

© MOSTAFA AMINI AFSHAR, 2010

Report No. X-10/252

Department of Shipping and Marine Technology

Chalmers University of Technology

SE-412 96 Göteborg

Sweden

Telephone +46 (0)31-772 1000

Printed by Chalmers Reproservice

Göteborg, Sweden, 2010

Abstract

An open source CFD solver, OpenFOAM®, has been used to create a numerical wave tank. The study is based on the interFoam solver, i.e. a solver for incompressible multiphase flow problems. The solver uses the finite volume method for the spatial discretization of equations and applies the VOF approach for the free surface modeling.

Initially a convergence study was carried out. The study was based on the propagation of fifth order Stokes waves in deep water condition. To this end two separate applications, *waveWriter* and *errorCalculator*, were created. With the former the initial conditions for the velocities and pressure of fifth order Stokes wave can be specified directly for interFoam. The errorCalculator is a post-processing tool that estimate the computational errors at each time step. The study revealed that the model exhibits only first order convergence. The loss of one order is due to the waveWriter setting only first order initial conditions.

Wave generation and absorption in the wave tank are performed by the relaxation method. For this purpose the existing interFoam solver has been partially modified in order for replacing the computational solutions with desired analytical ones inside the relaxation zones. In this manner the modified solver is able to generate and dissipate different wave types in the numerical wave tank. It is shown that outgoing waves are absorbed efficiently by extending the damping relaxation zone to at least three wavelengths, while one wavelength extension is required for the wave-generating zone.

To validate the numerical wave tank, the Whalin shoaling test was considered. Unfortunately, inadequacies in the then existing version of interFoam, (version 1.6.x), in the handling of the pressure force balance on non-orthogonal and distorted meshes, hindered and finally stopped the validation test process. Subsequently it was found that the newer version OpenFOAM-1.7.x, can be used promisingly for the validation of wave tank by Whalin test and this has been defined as a recommended future work.

Keywords: OpenFOAM, interFoam, Numerical Wave Tank, Nonlinear waves, Free surface flow, Relaxation Method, Wave Generation and Absorption.

Preface

This thesis is submitted as the partial fulfilment of the master degree in Naval Architecture at Chalmers University of Technology, Göteborg, and has been carried out at Department of Shipping and Marine Technology, Chalmers University of Technology.

The project was done under the kind supervision of Dr. C. Eskilsson at Chalmers University. Without his knowledge and generous instructions and helps, realization of the thesis would have been totally impossible. I would like to express my sincere gratitude to my supervisor for his knowledge, time, patience and continuous support of the thesis work.

My special thanks also goes to the head of Hydrodynamics Department at Chalmers University, Prof. R. Bensow, who let me to work on this subject, and guided me through the earlier stages of the thesis.

Contents

1	Introduction	1
1.1	Outline and aims of the study	2
2	Water wave theory	3
2.1	General	3
2.2	Modeling	3
2.3	Governing equations	4
2.4	Linear theory	5
2.5	Nonlinear theory	6
2.6	5th Order Stokes wave	7
2.7	Wave breaking	9
3	OpenFOAM®	10
3.1	OpenFOAM	10
3.2	Transport equations	11
3.3	Discretization method	12
3.4	Free surface modeling	12
4	waveWriter and errorCalculator	14
4.1	General	14
4.2	waveWriter	14
4.3	errorCalculator	15
5	Convergence study	23
5.1	Model specifications	23
5.2	Computational errors	24
5.2.1	Effects of cAlpha	24
5.2.2	Error Plots	24
5.2.3	Convergence order	32
5.2.4	Unwanted air velocity and premature breaker	33
5.3	Discussion	35
5.3.1	Relaxation of air velocity	36
5.3.2	Removing convection in the air phase	37
5.3.3	Non-Smoothness of initial wave profile	38
6	Wave Generation and Absorption	39
6.1	General	39
6.2	Relaxation Method	40
6.2.1	Relaxation Functions	42

6.3	waveFoam	43
6.4	Wave generation test cases	43
6.4.1	Linear progressive wave	43
6.4.2	Linear standing wave	44
6.4.3	Nonlinear progressive wave	46
7	Verification and validation	47
7.1	Test description	47
7.2	Mesh generation	48
7.3	Problems with OpenFOAM 1.6	48
8	Conclusions and recommendations	51

Chapter 1

Introduction

Design of marine structures is totally dependent on their behavior under the influence of environmental conditions such as ocean waves. Wave induced loads and motions must be studied and calculated in order for safe operation of the structures in deep oceans. This highlights the importance and significance of the study of wave hydrodynamics at least from the structure design point of view. Generally interesting variables in the calculation of wave induced load and motions are the wave velocity and pressure fields.

The study of wave hydrodynamics dates about a century back when the first theories for linear and nonlinear wave motions developed more in the interest of the science than design and applications in the oceans. Among them the Stokes theory for nonlinear water waves can be mentioned [16]. But during the twentieth century, the results of wave motion studies were more oriented to the purpose of design, and considerable researches have been carried out to investigate the interaction of waves with fixed and floating structures. For example Morison [17] carried out a classic study on the wave forces applied to the piles and proposed a way for their calculation.

Before the advent of high speed computers, experimental and analytical studies were the only tools at the hands of researchers in this field. Development of the high performance computers in the past twenty years added extensively to the power of researchers for the investigation of complicated problems of wave hydrodynamics using numerical methods called *computational hydraulics*. Consequently there was a quick growth in the number of numerical approaches which developed and used for the study of wave-induced forces and motions.

During the past ten years the open source codes for computational fluid dynamics problems have found attention among the researchers. It is specially desirable because the developed source code is free and open to all for the purpose of research and education. This enables researchers to have the source material and modify it for their own CFD problem. Among these open source CFD solvers is OpenFOAM [21] which has extensive use for fluid flow problem solving.

After the development of OpenFOAM and specially during the recent years, researchers in the field of ocean engineering have shown great interest to use OpenFOAM for study of wave induced forces on marine structures. This resulted in to development and creation of various methods for wave generation and dissipation in a numerical wave tank using OpenFOAM. Of the most notable research in this field is that of X. Liu [8] in which he conducted a very detailed study on the interaction of sea bed pore pressure and piling with current and waves in a numerical wave tank. Liu used a time varying velocity profile for wave generation in a manner of piston type wave maker. Also he applied relaxation

method for wave damping at the end of the tank. The attempts of some other researchers in this field, has emerged as OpenFOAM applications that can be used for wave generation. For example *GroovyBC* [22] is the name of an application that is for applying wave boundary, and has been widely used in studies regarding wave generation in OpenFOAM. However, a brief review of the literature on this subject, makes it known that a dedicated study on the potentials and drawbacks of OpenFOAM in use for numerical wave tank has not been done yet.

1.1 Outline and aims of the study

The Department of shipping and marine technology at Chalmers University found it interesting to define a project within the capacity of master thesis in Naval Architecture program for investigation of feasibility of OpenFOAM solvers for wave generation. The purpose of the thesis is to create a numerical wave tank by modification of some existing solvers and source codes in OpenFOAM. The project is motivated mainly by the idea that OpenFOAM as an open source package of CFD solvers can be applied for study of the response of floating structures due to ocean waves.

Obviously in the study of wave-induced forces and motions by numerical wave tanks, wave generation and dissipation are of the fundamental problems. And the thesis here tries to deal with these problems using OpenFOAM capabilities as far as possible.

The report starts with a very brief description of linear and nonlinear water waves and their breaking phenomenon in chapter 2. And then it follows in the next chapter with an explanation of numerical methods that is used by OpenFOAM.

For the purpose of the convergence study in this project, two separate application have been created that are used for initialization of wave field and calculation of computational errors. A description of theses application is given in chapter 4. A detailed study of the grid resolution requirements based on the accuracy for wave modeling in numerical wave tank has been performed by the prepared applications, and chapter 5 is for presentation of its results and relevant issues. In this chapter we also discuss some problems that may hinder the proper modeling of wave motion in OpenFOAM. The method for wave generation and dissipation in the tank is given in chapter 5. In this chapter it is shown how we modify the existing code to create a new solver that can be applied for wave generation and dissipation. We use the prepared solver for some well-known example of wave generation in order to verify the performance of the numerical wave tank.

We tried to validate the method by modeling of a three dimensional wave shoaling over a submerged half cylinder (Whalin experiment) [12]. But unexpectedly due to some shortcomings in the then version of source codes of OpenFOAM, we could not run the validation test. After a while it was revealed that newer version of OpenFOAM may be used successfully for the mentioned three dimensional test. So in chapter 6 we show the hindrances we faced while modeling Whalin experimental test in the numerical wave tank.

Finally a summary of the results of the project and some recommendation for future research in this subject is given in chapter 7.

Chapter 2

Water wave theory

2.1 General

To understand the behavior of ocean and coastal structures under the action of waves, it is necessary to know the forces applied on the structures. This can be achieved by estimation of pressure field of the wave that is itself coming from the study of kinematics of water particles.

An undulation on the surface of a still basin disturbs the hydrostatic pattern of the pressure field in such a way that calculation of pressure values can no longer be done simply by multiplying density, height and gravity. In fact vertical motion of water, induces additional vertical acceleration to the field of the particles in the basin. So we have to resort to the fundamental equations of fluid mechanics, and try to find the correct velocity and pressure field of the wave motion analytically. Due to reasons such as different boundary conditions and wave generating agents, it is not possible to describe the behavior of water wave motions just by a single universal theory. All type of motions from tsunami to small ripples induced by a droplet on the still water surface can have a generic name as *wave*, but the influencing forces and wave pattern do not have a similar identity.

2.2 Modeling

Although water wave theories are of a great use in understanding the behavior of ocean and coastal structures, some propositions are made before they can be applied to engineering works. For example in estimation of wave forces to objects in the ocean using the well-known *Morisson* equation, normally it is assumed that the wave kinematics are not influenced by the existence of the object itself. It is clear that acceptability of this assumption depends to a large extent on the degree of accuracy we care for, and on the relative dimensions of the wave and object.

Generally to deal with the problem we can resort to the following approaches :

- **Analytical**, Simply speaking, using analytical solution we set up a new boundary value problem for wave-structure system including the equations and the the relevant boundary conditions. The solution to the differential equations in this case gives the kinematics and dynamics of the water wave around the object. Approximation in this approach concerned with the neglecting of less dominant terms in the governing equations. Although we can bring some simplification to the equations,

analytical solution to wave problems like this requires a high knowledge and skill of advanced mathematics; not to mention the formidability of finding analytical solutions to many wave-structure problems.

- **Experimental**, The problem in this approach dealt with as it is. A scaled down model of prototype or sometimes the prototype itself is created to investigate the real situation in an experimental manner. Now everything concerned with proper measurements of the phenomenon and then correlation of the results. Outcomes of experimental studies of wave-structures problems are so valuable as they are coming form the study of real conditions. The benefits of experimental research can be compromised by the cost of model making incurred to the project or by some problems inherent to the experimental works like *scale effects*.
- **Numerical**, In a numerical study of the problem the fundamental equations and the relevant boundary conditions treated in a different manner than analytical approach. Now all equations of the wave-structure problem are discretized to create a matrix of linear systems of equations. In this case the equations remain unchanged, but the approximation lies in the numerical schemes that are used for discretization of the system of equations. At the expense of time and cost for computations, numerical models are powerful and flexible tools.

2.3 Governing equations

Normally, in the numerical and analytical study of wave hydrodynamics like all other fluid flow problems, the fundamental governing equations are the set of Navier Stokes equations for an incompressible, constant viscosity fluid as follows:

$$\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = -\frac{\partial p}{\partial x} + \rho g_x + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (2.1)$$

$$\rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = -\frac{\partial p}{\partial y} + \rho g_y + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) \quad (2.2)$$

$$\rho \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = -\frac{\partial p}{\partial z} + \rho g_z + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \quad (2.3)$$

In which μ is the viscosity, and u , v , and w are velocities in three coordinate systems x , y and z respectively. Moreover the conservation of mass in fluid flow is described by the differential continuity equation for an incompressible fluid as:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (2.4)$$

Equations (2.1)-(2.3) together with the above continuity equation and the relevant boundary conditions describe the motion of a viscous incompressible fluid flow. Navier Stokes equations are of the well known nonlinear and coupled system of differential equations, and finding the its analytical solution for many fluid problems is impossible.

In study of wave motion and for many engineering applications due to the large dimensions of the structures and intensity of the motion of the water particles, viscous effects of the fluid are negligible in the most part of the medium. This can be used, together with the assumption of irrotational flow, to simplify the problem of defining a wave theory to some extent as it allows the use of velocity potentials in the fundamental differential equations which reduces the number of fluid unknowns at the expense of increasing the order of the equations (potential flow problem) [18]. Generally this turns the continuity equation to a Laplacian (a linear equation) of the velocity potential that should be solved under the imposition of linear and nonlinear boundary conditions. So for a three dimensional three directional wave kinematics that can be described by a velocity potential ϕ , continuity equation is defined as :

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} \quad (2.5)$$

Where x , y and z are two coordinates in horizontal and one in vertical direction respectively. Boundary conditions can be divided in to two groups, i.e:

Linear BC

1. Bed boundary condition that requires specification of horizontal and vertical components of velocity at the bed. For instance for an impermeable sea bed the vertical velocity is set to zero, and slip condition is applied to the horizontal velocity.
2. Periodic boundary condition that requires periodicity of the wave characteristics both in time and space.

Nonlinear BC

1. Dynamic free surface boundary condition that is generally a force balance at the interface between the air and water
2. Kinematic free surface boundary condition that relates the vertical velocity of the particles on the interface to the total time rate change of height of the surface elevation with respect to a datum.

Depending on the way of treatment of the above mentioned nonlinear boundary conditions, wave theories can be divided generally into two wide categories of *Linear* and *Nonlinear*. Within each category there are a variety of wave theories that are capable of predicting the wave motion for different conditions of the sea environment.

According to the use that made of the wave theories in this project, in what follows a brief description of linear and nonlinear wave theories is presented.

2.4 Linear theory

Airy wave theory [1], is the well-known linear theory that has a considerable use in engineering applications. In linear wave theory all nonlinear boundary conditions are linearized and the problem turns to solve a linear second order differential equation under the application of linear boundary conditions. The nonlinear dynamic and kinematic conditions described as:

$$-\frac{\partial\phi}{\partial t} + \frac{u^2 + w^2}{2} + g\eta = C(t) \quad \text{on} \quad z = \eta(x, t) \quad (2.6)$$

that is just the unsteady form of Bernoulli equation for an incompressible fluid applied on the free surface with $p = 0$, and

$$-\frac{\partial\phi}{\partial z} = \frac{\partial\eta}{\partial t} - \frac{\partial\phi}{\partial x} \frac{\partial\eta}{\partial x} \quad \text{on } z = \eta(x, t) \quad (2.7)$$

A simple way of linearization of nonlinear free surface boundary conditions is described in [1] that applies the Taylor series to approximate the value of the boundary conditions on the free surface by expanding their values around the mean water surface and neglecting higher order terms. This is why linear theory is also called small amplitude wave theory as the approximation remains acceptable by applying the theory to small amplitude undulations of sea water or small $\frac{kH}{2}$. citeDean

Now the resulting boundary value problem can be solved analytically to find the velocity potential. This has been done using the method of separation of variables in [1]. The resulting value for velocities and pressure field are as follows:

$$u = \frac{H}{2}\sigma \frac{\cosh k(h+z)}{\sinh kh} \cos(kx - \sigma t) \quad (2.8)$$

$$w = \frac{H}{2}\sigma \frac{\sinh k(h+z)}{\sinh kh} \sin(kx - \sigma t) \quad (2.9)$$

$$p = -\rho gz + \rho g \frac{H}{2} \frac{\cosh k(h+z)}{\cosh kh} \cos(kx - \sigma t) \quad (2.10)$$

It is evident from the expressions that the dependency of the water wave to the horizontal space and time is governed by the periodic functions of \sin and \cos . The dependency to the vertical space, depth, is defined by the exponential functions of \sinh and \cosh . According to linear theory the shape of the wave is described by just a simple sine or cosine function. The crest and the trough have the same pattern and the path-lines of water particles consist of closed orbits. Moreover it is now clear that pressure is no longer a static field as we can see that in the pressure equation an extra term has been added to the product of ρ , g and z . These are in fact an indication of the vertical acceleration of water particles.

Sea water as a medium for propagation of water waves is a dispersive medium and waves with different frequencies and lengths have different celerities in it. The dispersion relationship for linear waves are described as :

$$\sigma^2 = gk \tanh kh \quad (2.11)$$

Although linear theory gives us a great deal of insight into the characteristics of the wave motion and has considerable engineering application, its range of applicability is limited to small $\frac{kH}{2}$. [1]. So nonlinear wave theories defined in order to be able to predict a larger range of wave motions that observed in the ocean and coastal regions.

2.5 Nonlinear theory

Generally in nonlinear theories, dynamic and kinematic free surface boundary conditions treated in a different manner than linear one. The well-known Stokes wave theory describes a group of nonlinear water waves that are named according to the degree of non-linearity described by them. This category of nonlinear waves can be defined by

perturbation method for solving differential equations [1]. The method is suitable to describe the behavior of systems that are so sensitive to variation of a particular parameter (perturbation parameter) in their governing equations. The solution is presented by an expansion series of the perturbation parameter as :

$$x = X_0 + \epsilon X_1 + \epsilon^2 X_2 + \epsilon^3 X_3 + O(\epsilon^4) \quad (2.12)$$

Now in the simplest case when the perturbation parameter ϵ is zero, the solution is described by just the first term in the series X_0 . Depending on the value of ϵ we can add more higher order term to close to the exact solution. After substitution of the expanded solution to the equation and boundary conditions the solution for X_1, X_2, \dots can be found successively.

In Stokes wave theory steepness of the wave ka designated as the perturbation parameter for the expansion. Successive inclusion of the parameter ka in the system of equations and boundary conditions results in the solution for Stokes wave of increasing order. Finding the solution for higher order theories than three becomes so difficult in such a way that resorting to some other alternative methods is more desirable. In order to illustrate the influence of non-linearity ka in wave kinematics and shape, the results for velocity components and wave profile for a second order Stokes wave are given as follow [1] :

$$u = \frac{H}{2} \frac{gk}{\sigma} \frac{\cosh k(h+z)}{\cosh kh} \cos(kx - \sigma t) + \frac{3}{16} \frac{H^2 \sigma k \cosh 2kh(h+z)}{\sinh^4 kh} \cos 2(kx - \sigma t) \quad (2.13)$$

$$w = \frac{H}{2} \frac{gk}{\sigma} \frac{\sinh k(h+z)}{\cosh kh} \sin(kx - \sigma t) + \frac{3}{16} \frac{H^2 \sigma k \sinh 2kh(h+z)}{\sinh^4 kh} \sin 2(kx - \sigma t) \quad (2.14)$$

$$\eta = \frac{H}{2} \cos(kx - \sigma t) + \frac{H^2 k}{16} \frac{\cosh kh}{\sinh^3 kh} (2 + \cosh 2kh) \cos(kx - \sigma t) \quad (2.15)$$

First terms are the contribution of the first order Stokes wave, which is in fact the linear solution, and the second terms gives the influence of steepness on the solution. In contrary with linear theory, solutions of Stokes waves are described as a function of ka as we assumed earlier.

It is also evident that the shape of the wave is no longer a simple cosine function as an extra term has been added now to the wave profile. This causes the Stokes waves to be steeper on the crest and flatter on the trough. In nonlinear waves the path-line of the water particles is no more a closed orbit and particles experience a net excursion after one period of the wave. Generally, Stokes waves are suitable to predict the wave motion for deep water conditions ($\frac{h}{L} > 0.5$)

2.6 5th Order Stokes wave

The focus in this project shall be mainly on the behavior of deep water waves. This led us to look for a wave theory that is basically applicable for deep water wave modeling. Consequently, 5th order Stokes wave field that has been proposed by [2] is used for this purpose. This analytical solution is for an inviscid incompressible fluid and defined using the potential flow theory. The solution for Stokes wave that is presented here will be used for resolution study in the project. Skjelberia and Hendrickson in [2] expanded the

velocity potential and wave profile by trigonometric series as follows:

$$\begin{aligned} \frac{\beta\phi}{\bar{C}} = \frac{2\pi\phi}{L\bar{C}} &= (\lambda A_{11} + \lambda^3 A_{13} + \lambda^5 A_{15}) \cosh \beta S \sin \theta \\ &+ (\lambda^2 A_{22} + \lambda^4 A_{24}) \cosh 2\beta S \sin 2\theta \\ &+ (\lambda^3 A_{33} + \lambda^5 A_{35}) \cosh 3\beta S \sin 3\theta \\ &+ (\lambda^4 A_{44}) \cosh 4\beta S \sin 4\theta \\ &+ (\lambda^5 A_{55}) \cosh 5\beta S \sin 5\theta \end{aligned} \quad (2.16)$$

and

$$\begin{aligned} \beta y &= \lambda \cos \theta + (\lambda^2 B_{22} + \lambda^4 B_{24}) \cos 2\theta \\ &+ (\lambda^3 B_{33} + \lambda^5 B_{35}) \cos 3\theta + (\lambda^4 B_{44}) \cos 4\theta \\ &+ (\lambda^5 B_{55}) \cos 5\theta \end{aligned} \quad (2.17)$$

Where θ is the phase angle and equal to $\frac{2\pi}{L}(x - \bar{C}t)$. λ is a constant to be determined for each wave.

The expansions show how the final solution is divided to different parts according to the perturbation parameters that are a combinations of A_{ij} and B_{ij} . The first contributing term in (2.16) and (2.17) can be easily recognized as the linear wave solution. Substitution of the assumed profile and velocity potential in the continuity equation and the relevant boundary conditions resulted in twenty equations involving the twenty constants A_{ij} and B_{ij} . These constants have been calculated as the functions of h/L .

To find the value of λ , Skjelberia and Hendrickson made use of the fact that the wave height is actually the difference between wave profile at two phase angles of π and 0. It gives the relationship for λ as:

$$\frac{\pi H}{h} = \frac{1}{\frac{h}{L}} [\lambda + \lambda^3 B_{33} + \lambda^5 (B_{35} + B_{55})] \quad (2.18)$$

The relationship between velocity components and pressure is expressed by Bernoulli equation as follows:

$$\frac{2p}{\rho} + (u^2 + v^2) + 2\frac{\partial\phi}{\partial t} = -2g(C(t) + S - h) \quad (2.19)$$

Where $C(t)$ is the Bernoulli constant.

In the process of finding analytical solution, K and \bar{C} are expanded as equations below:

$$\beta K = \lambda^2 C_3 + \lambda^4 C_4 \quad (2.20)$$

$$\beta \bar{C}^2 = C_0^2 (1 + \lambda^2 C_1 + \lambda^4 C_2) \quad (2.21)$$

Using the expansion for \bar{C} they derived the dispersion equation for fifth order Stokes wave as follows:

$$\frac{h}{L_0} = \frac{h}{L} \tanh \beta d [1 + \lambda^2 C_1 + \lambda^4 C_2] \quad (2.22)$$

Where $L_0 = \frac{gT^2}{2\pi}$. According to their solution a complete wave can be described by knowing water depth, wave height and period. Then the ratio $\frac{h}{L}$ can be calculated by iterative solution of (2.18) and (2.22).

In this project we follow a different track, and at the beginning assume a value for wave height, water depth and wave length. This gives a definite value for deep water ratio $\frac{h}{L}$. Consequently values of all coefficients and wave period can be calculated.

2.7 Wave breaking

Generally the mechanism of wave breaking differs from shallow water to deep water conditions. In shallow water the main reason for wave breaking is the excessive *shoaling* of the waves while propagating over the sea bed. The height of shallow water waves is strongly influenced by the changes in water depth, and increase as the wave train travels over the shallow regions or towards the shore. This can be easily verified by the conservation of wave energy. Normally the criteria for wave breaking in shallow waters are expressed based on the sea bed steepness and water depth. According to one of the simplest and earliest criterion, shallow water wave would break when its height reaches to 0.78 times the water depth [1].

In deep water conditions wave breaking occurs due to the hydrodynamic instabilities. There are some theoretical explanations for the mechanism of these instabilities that give us the extreme limit for deep water wave heights. A comprehensive review on the studies of mechanisms for deep water wave breaking is presented in [6]. In this project wave breaking in deep water is considered to be based on the Fenton expression [5] that proposes a limit for the steepest stable water waves in deep water as follows:

$$\left(\frac{H}{L}\right)_{max} \approx \frac{0.141063 + 0.0095721\left(\frac{2\pi}{kh}\right) + 0.0077829\left(\frac{2\pi}{kh}\right)^2}{1 + 0.0788340\left(\frac{2\pi}{kh}\right) + 0.0317567\left(\frac{2\pi}{kh}\right)^2 + 0.0093407\left(\frac{2\pi}{kh}\right)^3} \quad (2.23)$$

Chapter 3

OpenFOAM[®]

3.1 OpenFOAM

Open Field Operation And Manipulation, is an open source package of C++ libraries and codes that are created to conduct numerical modeling of solid and fluid mechanics problems. The whole tasks of mesh generation, discretization of equations and matrix manipulations can be performed using applications that are intact an "assembly" of source codes and libraries in OpenFOAM. Mainly they are written using *object oriented programming* method, and this greatly simplifies manipulation with field values like velocities and pressure. The package enables us to prepare a vast variety of user-defined applications that suits our desired problem. OpenFOAM can be downloaded as pre-compiled executable applications, or it is available as the original source codes and needs compilation to create executables. It also comes with a third party package that provides extra and valuable supports for the whole OpenFOAM. Among them *paraview* can be mentioned which is used for post-processing of the field data.

OpenFOAM has its own specially designed data types and classes that enable user to manipulate field operation in a very effective way. For instance to define a velocity field over a specified grid points, one needs just to create an instance of a class called *volVectorField* for velocity. And this alleviates the need for accessing all data points while doing some operation on them.

We make use of OpenFOAM in this project first of all for writing an application that enables us to write the initial conditions for a 5th order Stokes wave field including its profile, velocities and pressure values. In the same manner another application is prepared that is able to compare computational velocity field with the analytical one, in order for calculation of error values at each time step. With the mentioned applications and *interFoam* solver we conduct a grid resolution study in chapter 4.

Next use of OpenFOAM is for implementation of a numerical wave tank which is done by direct modification of *interFoam* solver. Basically it is written for multiphase problems in incompressible fluid mechanics problems. It uses control volume numerical method for discretization of transport equations, and is able to calculate free surfaces according to *Volume of Fluid* approach.

3.2 Transport equations

Since discretization of fundamental equations in OpenFOAM is performed using control volume scheme, we prefer here to present governing equations defined in the previous chapter, as one *transport equation* that can represent all relevant fluid mechanics equations like continuity, momentum and energy. Conservative form of the equation describing transportation of a general property φ by diffusion and convection in a compressible Newtonian fluid flow can be defined as follows:

$$\frac{\partial(\rho\varphi)}{\partial t} + \text{div}(\rho\varphi\mathbf{u}) = \text{div}(\Gamma \text{grad } \varphi) + S_\varphi \quad (3.1)$$

That is obtained by applying the *Reynolds Transport Theorem* to a differential control volume in order to describe the time rate of change of a particular amount of the property φ . At the left the first term is to represent the time rate of change of φ within the control volume. Convection and diffusion flux of φ from the control surfaces are defined by the second term in the left and first term in the right of the equation respectively. The last term in the equation signifies the existence of sources or sinks of the property φ in the control volumes. In above equation φ can represent any scalar quantity like concentration, temperature, a pollutant or even the value of the velocity.

Now based on the type of the problem being discussed in this project we represent continuity and momentum equations by inserting the relevant property in transport equation. To obtain continuity equation we substitute φ by 1 and get the continuity equation for an incompressible and steady fluid flow as :

$$\text{div}(\mathbf{u}) = 0 \quad (3.2)$$

By replacing φ by the value of the velocity vector and selecting appropriate values for diffusion coefficient Γ and source terms, we reach to the momentum equations for incompressible fluid as given below :

x-momentum

$$\frac{\partial u}{\partial t} + \text{div}(u\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \text{div}(\mu \text{grad } u) \quad (3.3)$$

y-momentum

$$\frac{\partial w}{\partial t} + \text{div}(w\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \text{div}(\mu \text{grad } w) \quad (3.4)$$

z-momentum

$$\frac{\partial v}{\partial t} + \text{div}(v\mathbf{u}) = -\frac{1}{\rho} \frac{\partial p}{\partial z} + \text{div}(\mu \text{grad } v) \quad (3.5)$$

In fact they are transport equations for velocity values in a fluid field. The material derivative of velocity (acceleration) is given by time rate and convective terms on the left side. On the other side of the equations pressure and shear forces are represented by source and diffusion terms respectively.

3.3 Discretization method

In finite volume method integration of continuity and momentum equation over control volume and time gives :

$$\int_{CV} \left[\int_t^{t+\Delta t} \frac{\partial}{\partial t} (\rho\varphi) dt \right] dV + \int_t^{t+\Delta t} \left[\int_A \mathbf{n} \cdot (\rho\mathbf{u}\varphi) dA \right] dt = \int_t^{t+\Delta t} \left[\int_A \mathbf{n} \cdot (\Gamma \text{grad } \varphi) dA \right] dt + \int_t^{t+\Delta t} \int_{CV} S_\varphi dV dt. \quad (3.6)$$

Where in the equation, Gauss theorem is applied to convert integral of divergence terms over control volume to the flux of the vectors over the control surfaces. In this way for specified control volume, flux terms can be easily determined by surface integration. For convection terms in the equation we need to calculate net flux of $(\rho\varphi u)$ and for diffusion terms we need to find net amount of flux of $(\Gamma \text{grad}\varphi)$ from the surface of the control volume. For spacial discretization of the transport equation one should decide on the schemes that can be used to represent the gradient of the property and the value of the property itself on the control surfaces. The same also applies for diffusion coefficient, density and velocity. This leads us to different discretization schemes within the control volume approach, among which central differencing for diffusion terms and upwind scheme for convection terms are commonly used.

Conventionally the result of integration of source terms over control volume in the equation are kept as a linear function of the φ . Unsteady term in equation 3.6 can in the simplest case be discretized by differencing the next and current time values of general property φ .

For the time integrations we need to decide at which time the values of integrand should be evaluated. The result is different temporal discretization schemes with their own advantages and disadvantages. To march the solution forward in time by explicit schemes all current values should be applied for the integration, and in implicit schemes next step values are used in the integrand of the diffusion and convection terms in equation 3.6. Additionally there are cases which a combination of next and current time values are evaluated to march the solution in the time. A detailed discussion on finite volume method and various discretization schemes can be found in [4].

Finally discretized equations are assembled in to a systems of linear equations as matrices and a matrix solution method is applied to reach the numerical solution of the problem.

3.4 Free surface modeling

Normally, in the modeling of free surface flow the dynamic and kinematic boundary conditions should be applied at the interface which is unknown before the flow problem is solved. Still to find the solution the free surface boundary conditions should be applied properly. As it has been mentioned earlier the problem in the linear waves theory is solved using linearization of the free surface boundary conditions.

Free surface tracking in OpenFOAM is done by *Volume of Fluid* approach [19] in the *interFoam* solver. Generally and form the viewpoint of transport equations, free surface motion of the fluid is a convective type transport of a property called *phase fraction* which is denoted by α in OpenFOAM. The property has values of zero and 1 to represent phase 1 (air) and phase 2(fluid) respectively. At each time step phase fractions are convected

by the existing velocity field, and in this way distribution and development of free surface can be estimated.

This is the basis of the free surface tracking by *volume of fluid method*, and still corrections and considerations need to be applied on the scheme. Of the most notable one is the remedy proposed to avoid adverse effects of diffusion of phase fraction that results in smearing of the sharpness of the interface. Although equations are written for convection of phase fraction, there can be always the possibility of getting *false diffusion* [4].

A possible remedy used in OpenFOAM is to introduce an extra term called *Artificial Compression* to the equation of phase fraction convection. Physically its role is to exert a pressure on the interface to keep it from dispersing. In this manner the transport equation becomes [7]:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) + \nabla \cdot (\alpha(1 - \alpha) \mathbf{u}_r) = 0 \quad (3.7)$$

Where \mathbf{u}_r is a velocity field normal to the interface and applies the artificial compression on the surface. Based on the the term $\alpha(1 - \alpha) \mathbf{u}_r$, the region under the influence of compression velocity has phase fraction values other than 0 and 1. In following chapters we present some results of the influence of the compression velocity on the wave profile. The artificial compression velocity in OpenFOAM is shown by *cAlpha* in the model setting. The zero value for *cAlpha* applies no compression velocity, and higher values than zero introduce the corresponding artificial velocities at the interface.

Chapter 4

waveWriter and errorCalculator

4.1 General

In order to get a better quantitative understanding of the correctness of the computational method used in this work for water wave modeling, a grid resolution study has been conducted. This enables us to find a proper computational requirements for a desired accuracy in the wave kinematics and dynamics. We shall investigate how far the computational solution is close to the analytical wave field for different steepnesses in deep water condition. As it has been mentioned earlier we use 5th order Stokes wave, and in order to facilitate its calculations a header file called *StokesCoeff.H* is prepared that gives Stokes wave constants and can be included easily in the source codes.

To perform the convergence study, first we initialize Stokes waves by setting their initial values for pressure and velocities in *interFoam*. This means that at time zero we are able to prescribe analytical Stokes wave profile and its corresponding kinematics and dynamics. After defining the initial conditions for the wave field, *interFoam* run gives the computational values of velocities, pressures and profile for the next time steps. Finally by comparison of the computational values and the corresponding analytical solution, we are able to find a measure of the error of the Stokes wave modeling by *interFoam*.

In this project two distinct applications have been prepared that are able to do initialization of the water waves and calculation of computational errors respectively. In following sections the workings of the applications are explained to some extent.

Before using the analytical solution for initialization of Stokes waves, we make sure that the wave steepness does not exceed the breaking limit in deep water region according to the relationship prescribed by [5], in equation (2.23). In order for simplifying the process of initialization, a separate application called *waveData* is created that gets input values for steepness, wave height, water depth and wave length. Then it gives such wave data as λ and wave period after checking the steepness against equation (2.23).

4.2 waveWriter

An application called *waveWriter* has been written that enables us to write initial conditions for velocities, pressure and profile of the Stokes wave in OpenFOAM. Using this application a Stokes wave is written over the entire grids of computational domain at time 0 and is located at 0 folder of the case directory. It is also able to write the initial fields for a progressive and standing linear wave.

The application uses center of control volumes as the points on which the wave field values are calculated, so in order to better resolve the wave information we need to provide a fine grid at least over a band of the width of a wave height. A complete set of initial conditions for the linear and fifth order Stokes waves, as an example run of the application is given in figure 4.1 to 4.4. In this example application writes a wave field of 250m length and 15m height in a water depth of 30m.

After setting the initial values for the fields, *interFoam* is run to solve for the wave motion over the run time of the case. This gives us a computational Stokes wave field that basically should look like the analytical one we defined at time 0. But as we shall see in following sections, there is always some source of errors that results in the deviation from the exact analytical solution we assumed at the beginning. We first introduce a way of estimation of the computational errors and then describe some remarkable sources of errors in the computation. Finally we present some remedy and recommendations in order to have limited and controlled computational errors in the numerical wave tank.

4.3 errorCalculator

errorCalculator is an application that has been prepared for calculation of computational errors during run time of the case. It works based on this idea that at each time step of the *interFoam* computation, the resulting field is supposed to look like the exact wave solution and all other differences are calculated as the errors. This means that application gives no estimation of errors for those parts of the fluid that may travel above the analytical wave profile for the same time step. This can occur for instance in breaking case that causes the wave to lose its normal shape. It is also important to note that in the case of dispersion of wave shape over the run time, the profile gradually turns to a smooth line. In this case, *errorCalculator* still calculates errors based on the computational values obtained for the cells below the exact profile at that time step, even if now they turned to air phase due to diffusion of numerical solution. The scope of the application is well illustrated in figure 4.5.

errorCalculator does no judgement for regions **III** and **V**, simply because these are above the analytical profile. In fact a value of zero is chosen for them. On the other hand, for regions **II** and **IV** the application calculates the amount of error with respect to the analytical solution, even if they are now part of the air phase. All computational cells in region **I** are in the fluid phase and are compared with the corresponding analytical solutions at that time step. In this manner application reports highest error values always around the wave interface as there would be some deviation between analytical and computational wave profile.

Error value for velocities and pressure is computed by the following relationship:

$$\chi_{Error} = \left| \frac{\text{Analytical } \chi - \text{Computational } \chi}{\text{Maximum Analytical } \chi} \right| \quad (4.1)$$

Where parameter χ represents two velocity components and the wave pressure. At each time step, computational values are compared with the analytical solution and the results is divided by the maximum value of the analytical solution that is a constant value for each particular Stokes wave. As it is shown in equation (4.1) error values are always positive.

After *interFoam* solves the computational fields *errorCalculator* writes three fields, *UxError*,

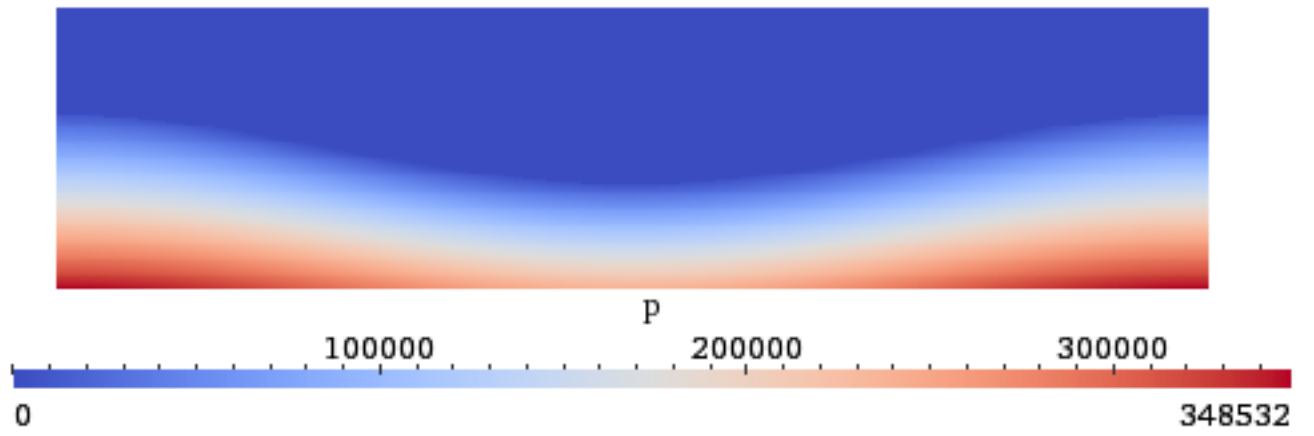


(a) alpha (Linear)

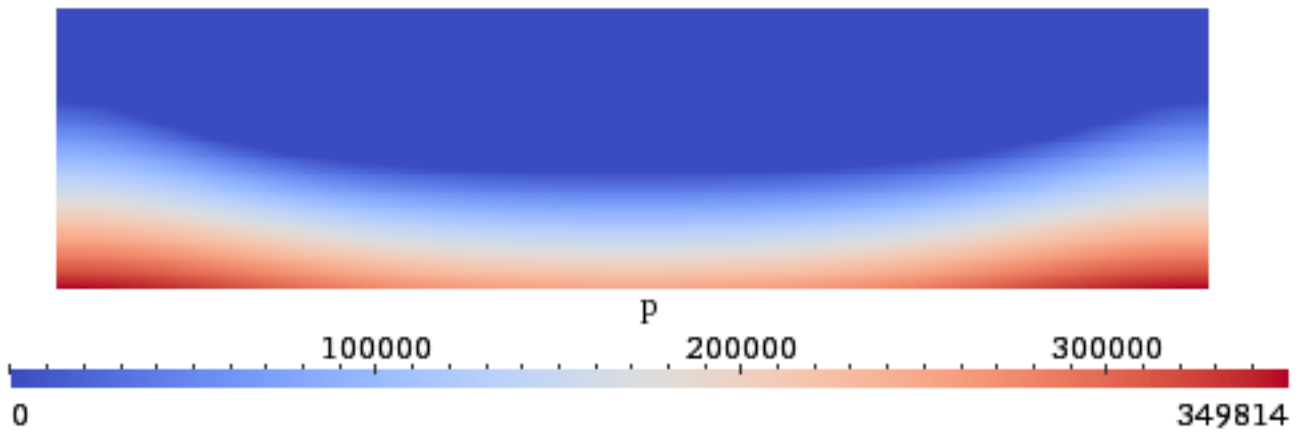


(b) alpha (Stokes)

Figure 4.1: Wave profile

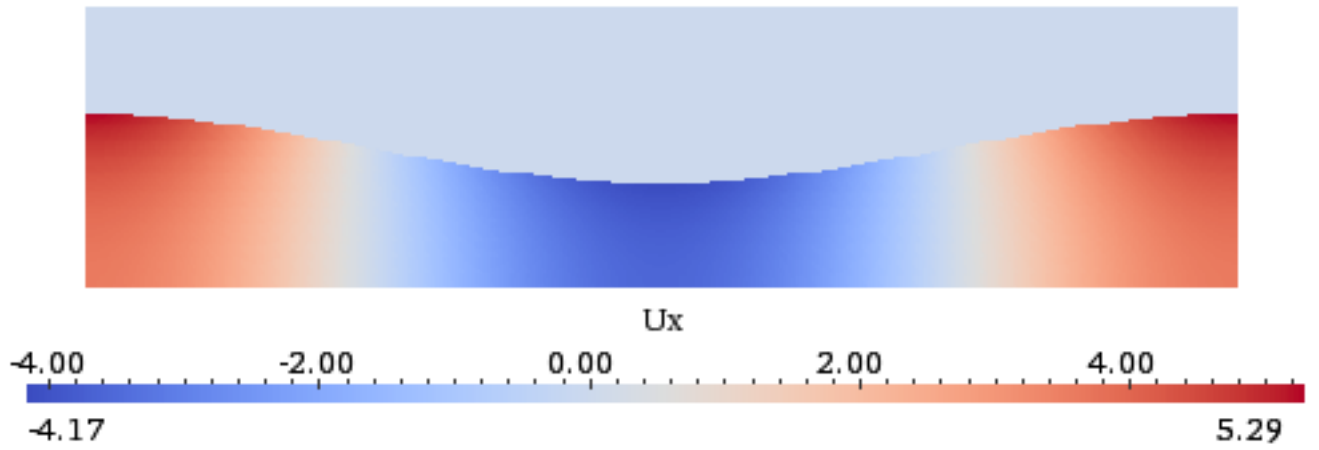


(a) Pressure (Linear)

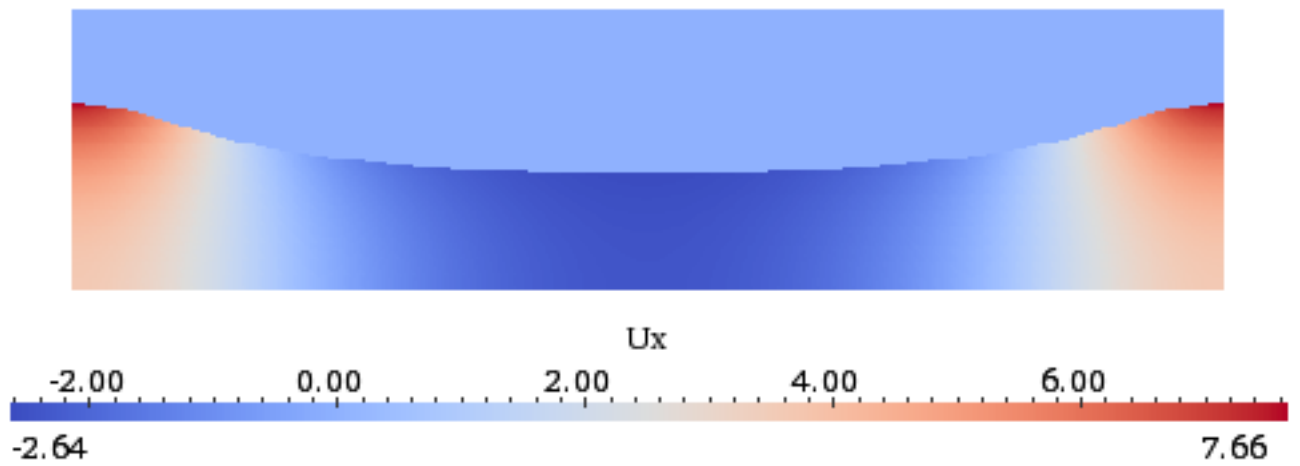


(b) Pressure (Stokes)

Figure 4.2: Wave pressure

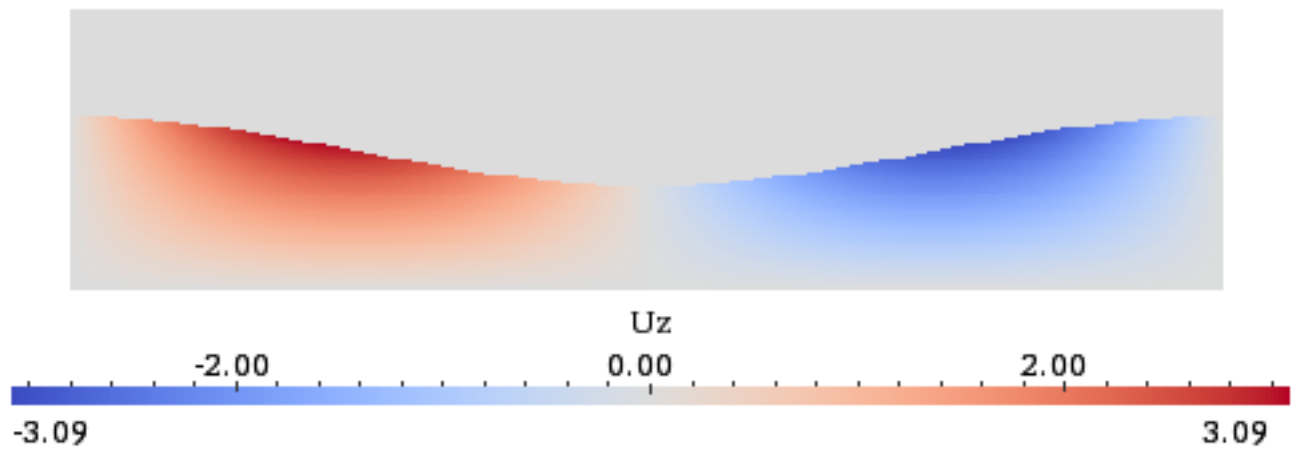


(a) U_x (Linear)

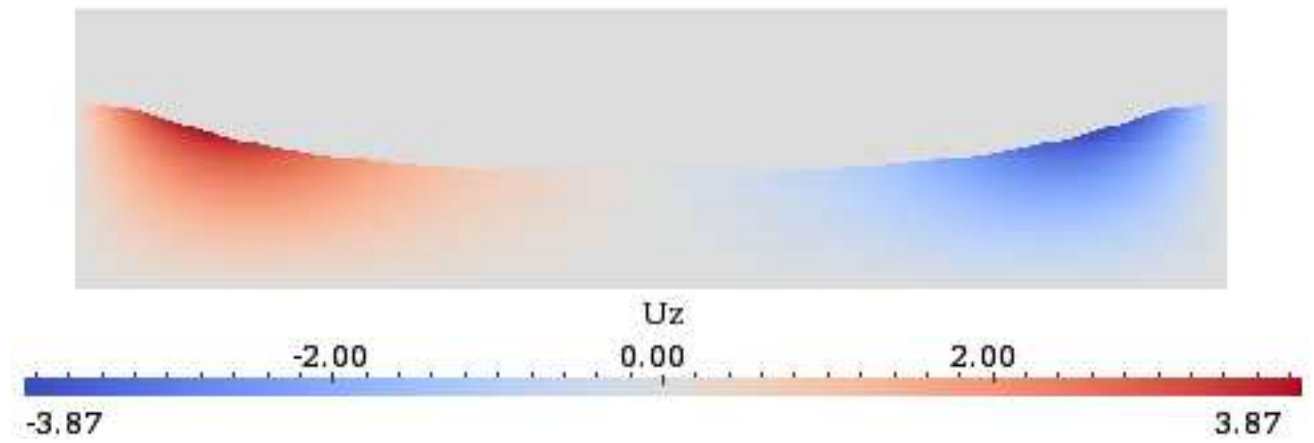


(b) U_x (Stokes)

Figure 4.3: Wave horizontal velocity



(a) U_z (Linear)



(b) U_z (Stokes)

Figure 4.4: Wave vertical velocity

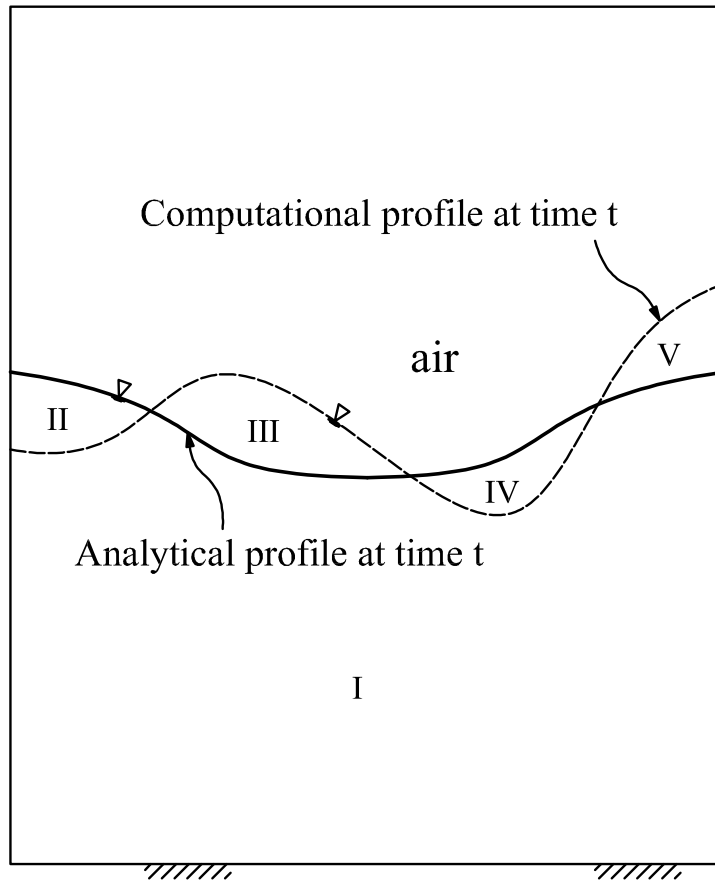
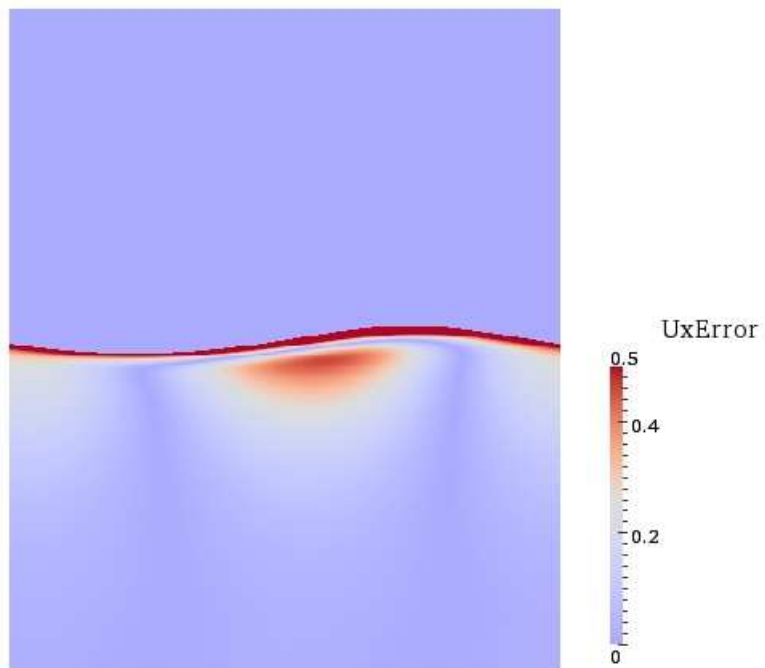
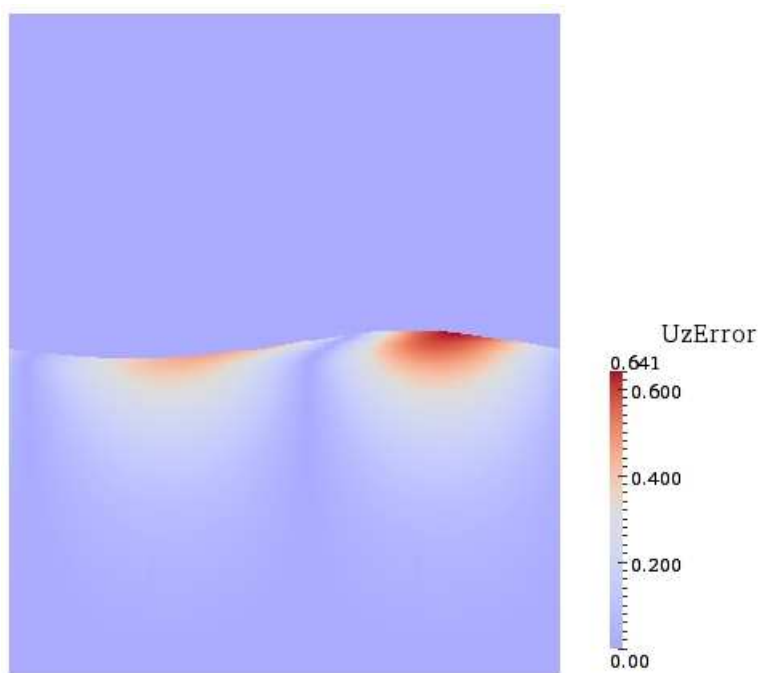


Figure 4.5: Regions for error calculation

$UzError$ and $pError$, in each time directory accordingly. Moreover, the maximum value of the errors of three fields at each time step are written in a text file, $LogErrors$, placed in the case directory. Figures 4.6 to 4.7 show how the results of a run of $errorCalculator$ look like for an example case. As it is seen, the color codes for error values can help us to better understand the regions with high and low errors throughout the wave field. An excerpt of $LogErrors$ is also presented in figure 4.8. Using these two applications a grid resolution study has been carried out and the results are presented in the next chapter.

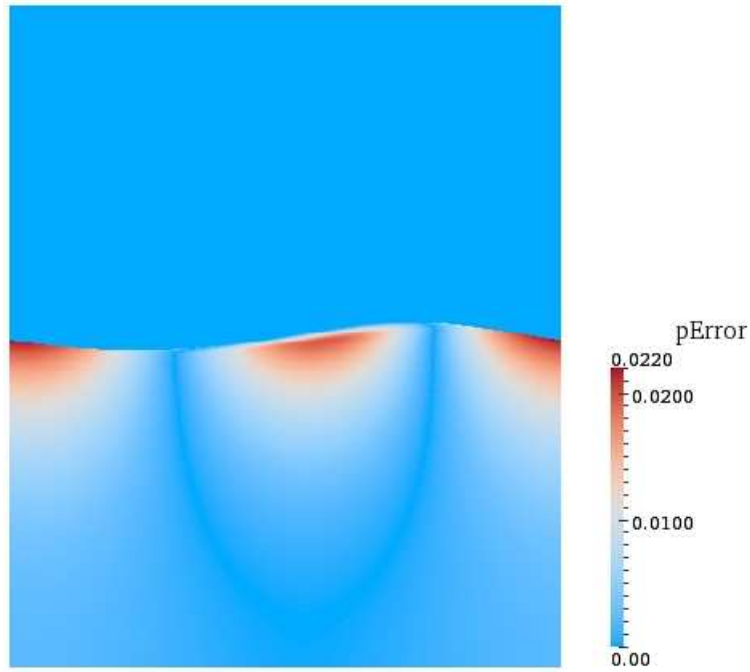


(a) U_xError



(b) U_zError

Figure 4.6: Velocity error color plots



(a) pError

Figure 4.7: Pressure error color plots

time	UxMaxErr	UzMaxErr	pMaxErr
0	5.45876e-07	6.03488e-07	1.69873e-06
6	0.487823	0.131444	0.00279495
12	0.612097	0.110304	0.00263902
18	0.719097	0.097934	0.00265084
24	0.810133	0.119482	0.00259735
30	0.886529	0.121407	0.00270078
36	0.950121	0.138883	0.00281928
42	1.01719	0.130432	0.00299639
48	1.08669	0.149317	0.00320489
54	1.16337	0.16269	0.00383453
60	1.23453	0.176204	0.00425498
66	1.27906	0.211275	0.00499861
72	1.32794	0.218875	0.00581366
78	1.36869	0.247678	0.00655683
84	1.40198	0.313009	0.00746574
90	1.42342	0.335801	0.0080595
96	1.47143	0.397705	0.00911198
102	1.51304	0.41979	0.00986064
108	1.57824	0.462805	0.0111766
114	1.63718	0.531131	0.0124753
120	1.76346	0.553170	0.0137505

Figure 4.8: Maximum Error values at each time step

Chapter 5

Convergence study

5.1 Model specifications

A two-dimensional wave tank with one wavelength extension is used for the resolution study. Periodic boundary conditions are set for both ends of the tank, and the bed boundary condition is defined in such a way that it permits horizontal components of velocity at the bottom of the tank (i.e slip condition). The wave profile is prescribed using *phase 1*, and the air at the top is represented by *phase 2* for the solver. See figure 5.1.

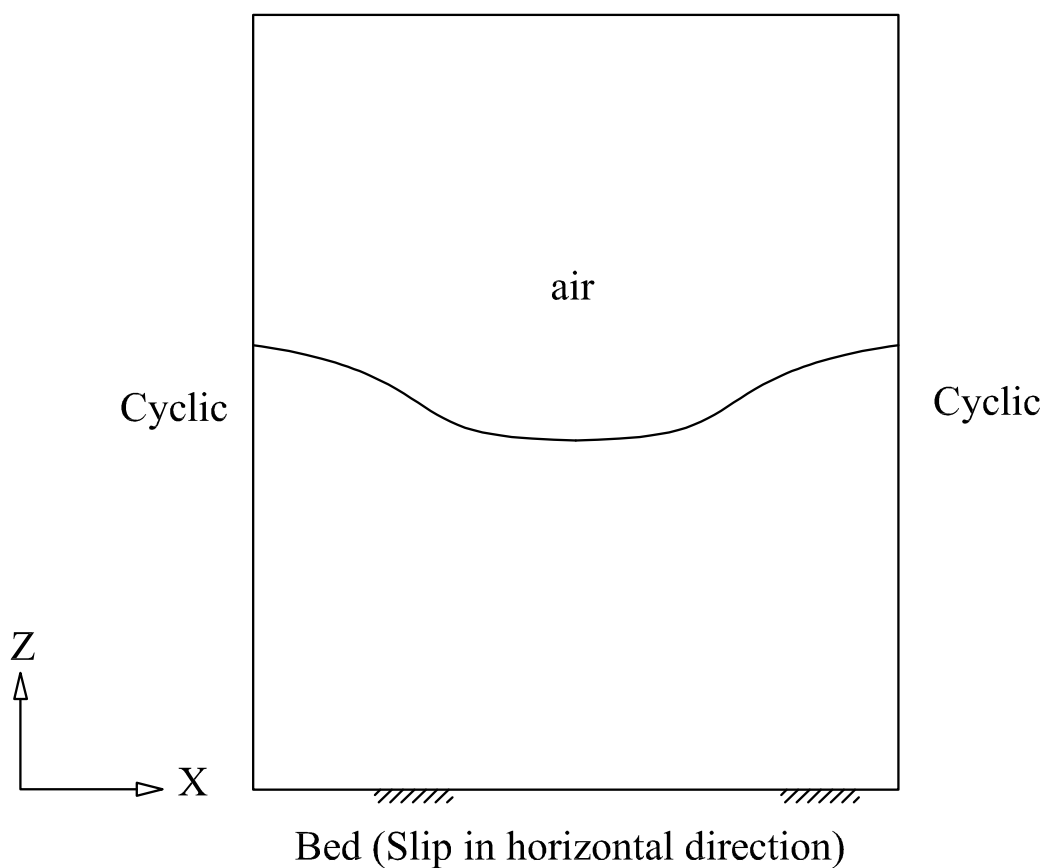


Figure 5.1: Boundary conditions

In order to be as close as possible to the Stokes theory, viscous and surface tension effects are not included in the study. All cases will be run in a deep water limit of $\frac{h}{\lambda} = 0.6$. After each time step, computational solution is compared with the corresponding analytical solution and the resulting error is normalized by the maximum analytical value of horizontal velocity, vertical velocity or pressure (whichever is relevant). For each case the maximum error of the whole computational wave field is singled out and kept for the purpose of resolution study. It is also important to note that in order to investigate the numerical diffusion and dispersion of wave profile, the total run time of the cases in the study is extended intentionally to about 100 wave period time. For the resolution study in this work, first of all we write the initial conditions by *waveWriter* and then solve the field over the run time by *interFoam*. Finally we use *errorCalculator* to estimate values of errors. In the next section a summary of results is presented.

As it has been mentioned earlier in this chapter, the study is done for a deep water ratio h/L of 0.6. Refinement of the grids is done by dividing each cell's dimension by 2 at each time. We start with 100 grids per wavelength as the initial grid size for the resolution study. So grids with numbers 1, 2, 4, 8,.. are grids with the initial size, $\frac{1}{2}$ of the initial size, $\frac{1}{4}$ of initial size and so on. It is also important to note that grids are uniformly distributed in the entire computational domain.

5.2 Computational errors

5.2.1 Effects of *cAlpha*

First we look in to the influence that artificial compression can have on the wave profile. For cases with $cAlpha = 0$, as it is expected, the interface profile will be smeared and diffused after some periods of the wave propagation, see figure 5.4. Although profile dispersion can be retarded by increasing the grid resolution, waves are finally doomed to lose their shape. This leaves always a still basin with no undulation of the surface.

By changing *cAlpha* to 1 the diffusion of interface will be reduced remarkably. This can be shown easily by comparison of two similar cases with different compression factor. Special care should be taken while applying values other than 0, since in long run of the cases, say 100 wave periods, compression velocity gives non physical interfaces. This is well shown in figures 5.2-5.5.

Wave with no compression factor disappeared totally, but artificial compression velocity for the other case produced a wave profile that is much higher than the expected analytical solution. The weird solution for the interface in the case of $cAlpha = 1$ is more pronounced for a very small amplitude wave ($H/L = 0.01$). See figures 5.3b and 5.5b. Generally, compression velocity performs its role very well during short runs of the *interFoam* for Stokes waves profile, but its use for longer run times should be accompanied by a more detailed study of its influences on the interface.

5.2.2 Error Plots

At this deep water limit we run three cases with different steepnesses and resolutions. We did not rely on artificial compression velocity and set $cAlpha = 0$. The results are presented as the graphs that show the amount of errors for two velocity components and pressure over 10 wave periods, say 60 sec. See figures 5.6 - 5.11. In the figures are also



(a) After 5 periods

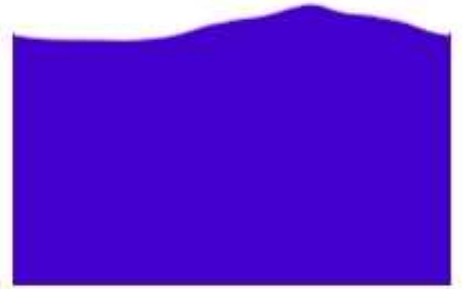


(b) After 50 periods

Figure 5.2: $c\text{Alpha}=0$ and steepness 0.01



(a) After 5 periods



(b) After 100 periods

Figure 5.3: $c\text{Alpha}=1$ and steepness 0.01



(a) After 5 periods

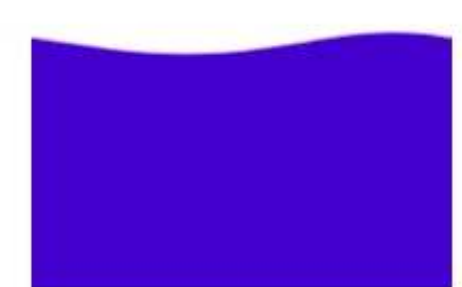


(b) After 50 periods

Figure 5.4: $c\text{Alpha}=0$ and steepness 0.02

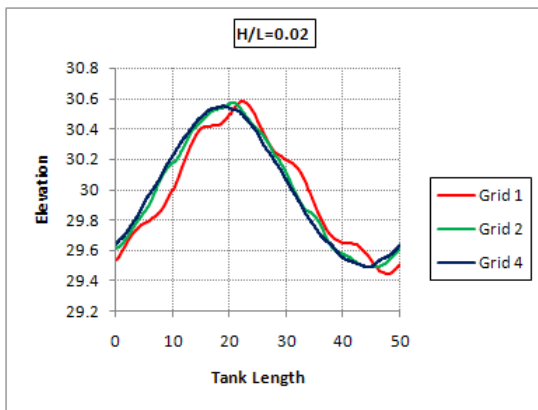


(a) After 5 periods

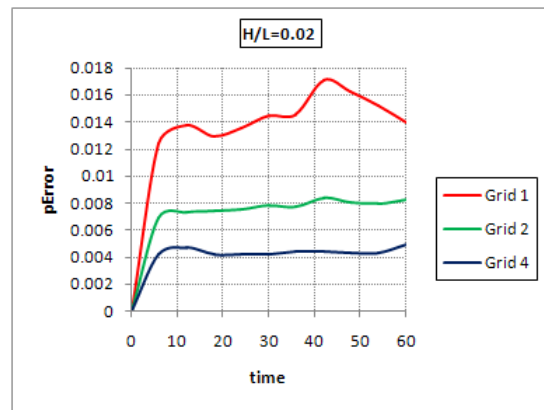


(b) After 50 periods

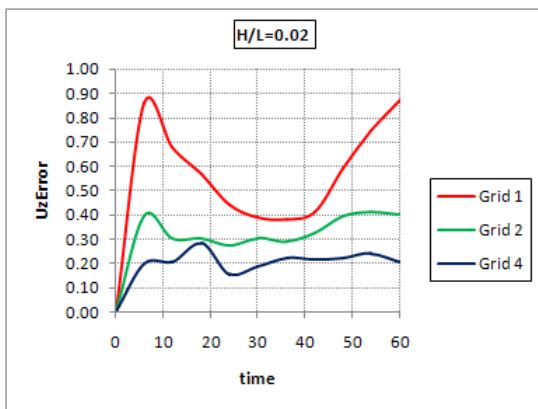
Figure 5.5: $c\text{Alpha}=1$ and steepness 0.02



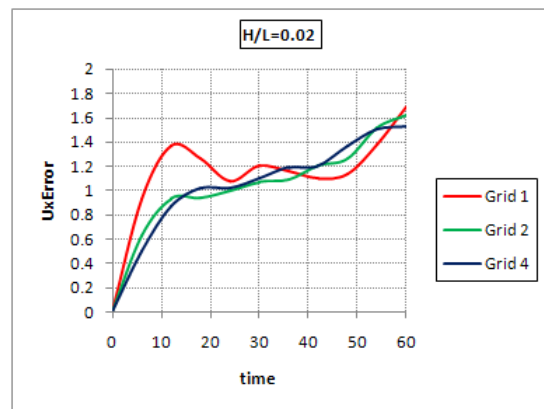
(a) Profile



(b) p

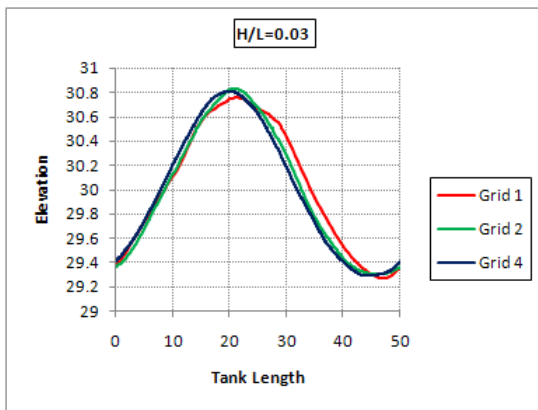


(c) U_z

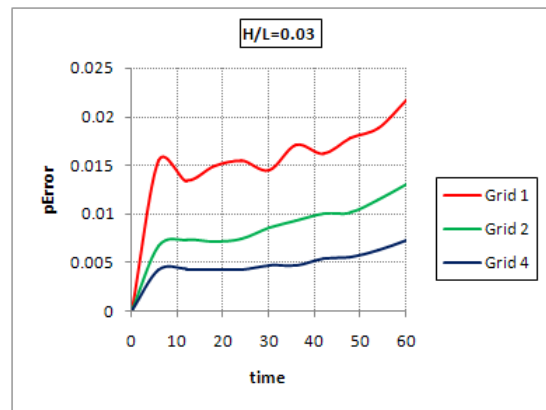


(d) U_x

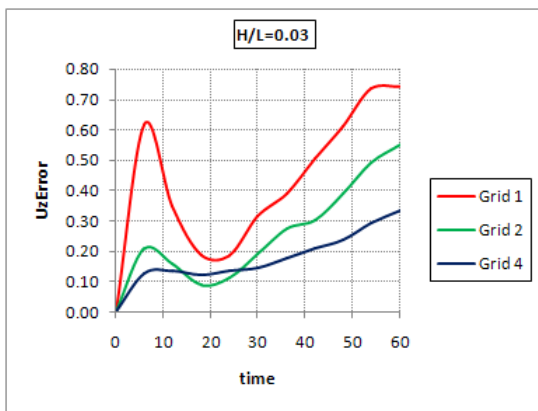
Figure 5.6: $H/L=0.02$



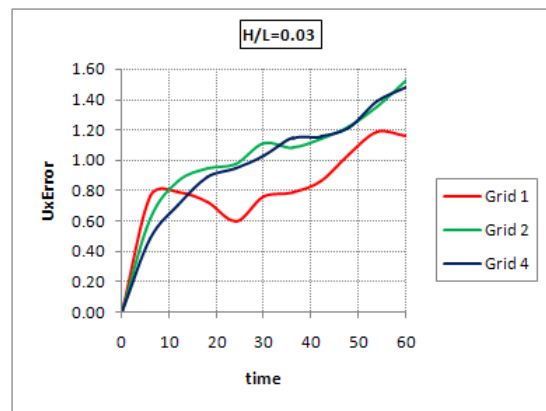
(a) Profile



(b) p

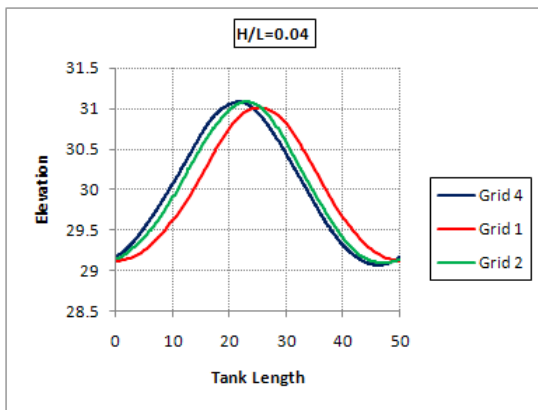


(c) U_z

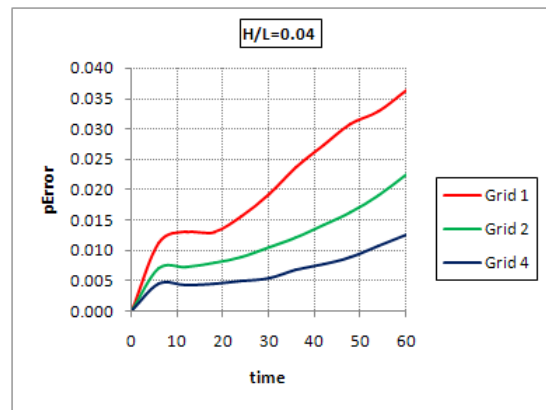


(d) U_x

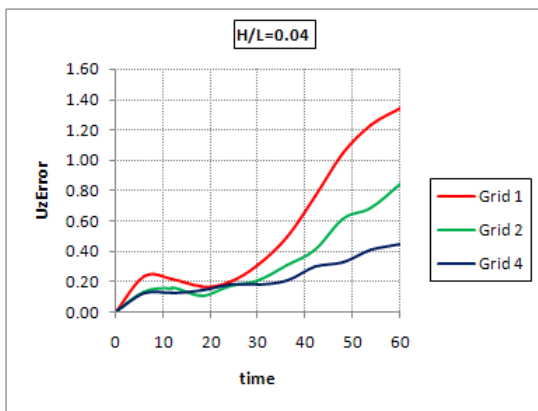
Figure 5.7: $H/L=0.03$



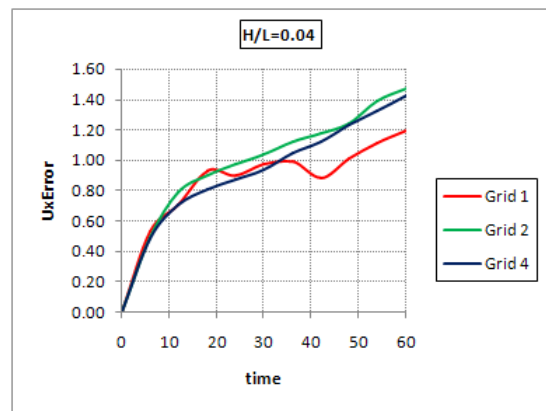
(a) Profile



(b) p

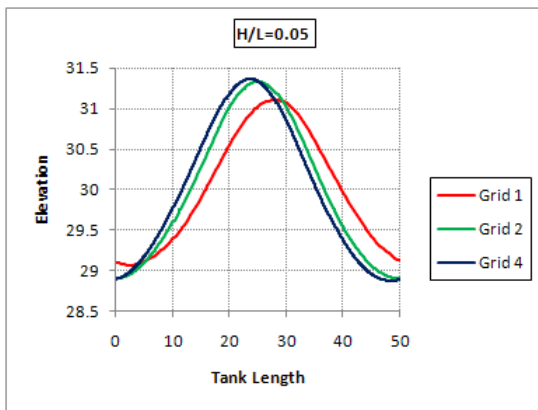


(c) U_z

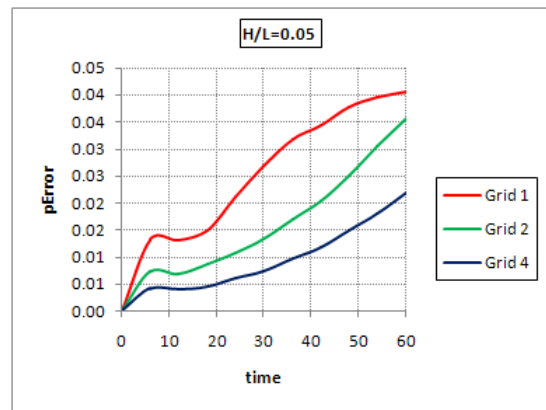


(d) U_x

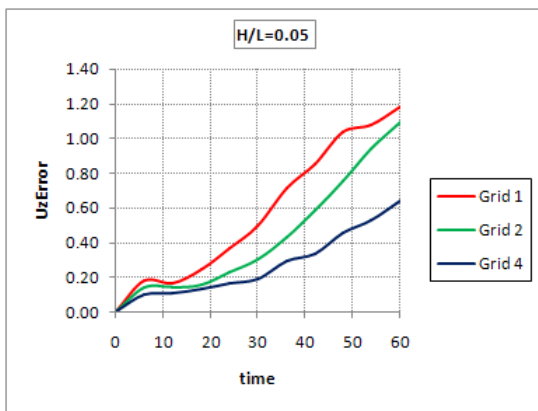
Figure 5.8: $H/L=0.04$



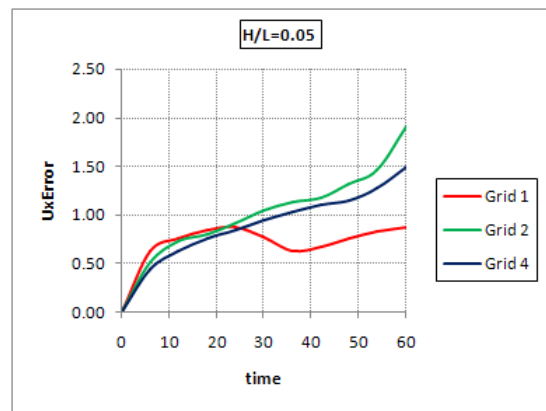
(a) Profile



(b) p

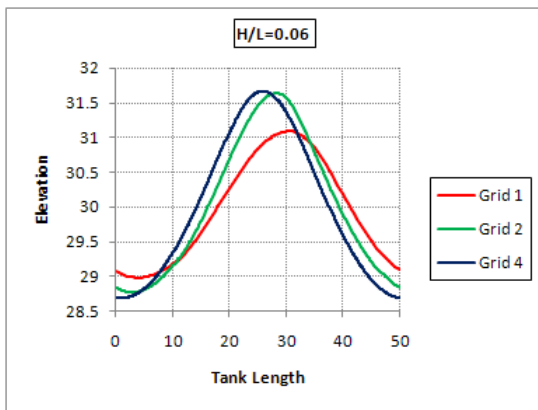


(c) U_z

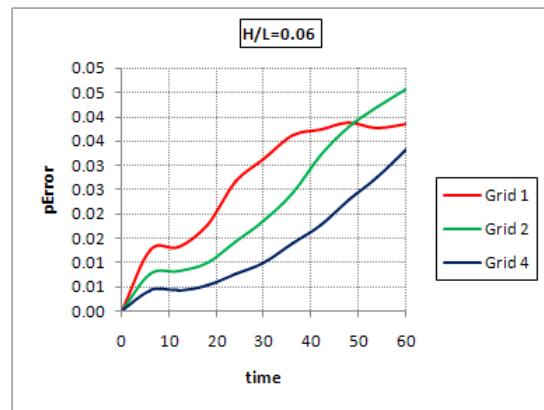


(d) U_x

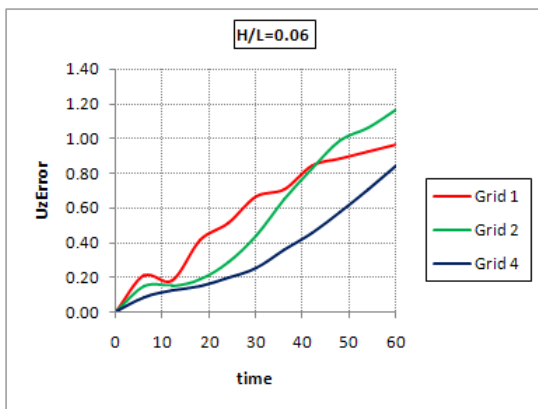
Figure 5.9: $H/L=0.05$



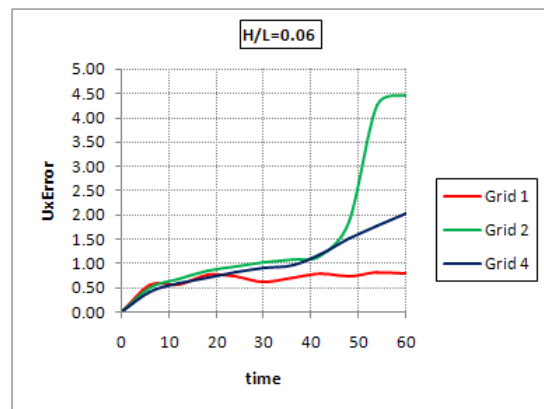
(a) Profile



(b) p

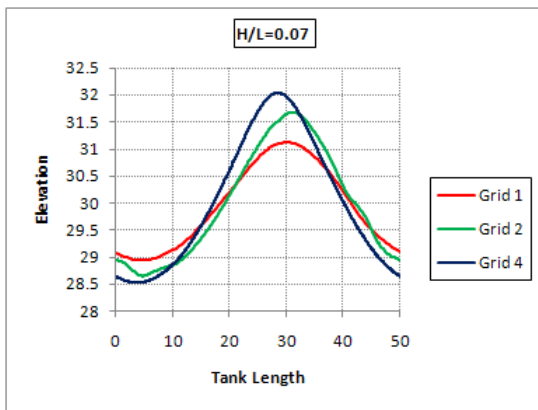


(c) U_z

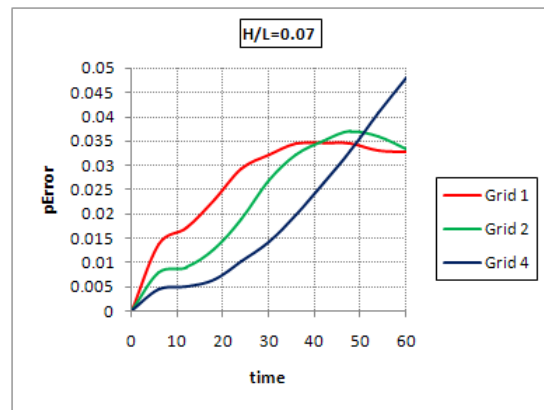


(d) U_x

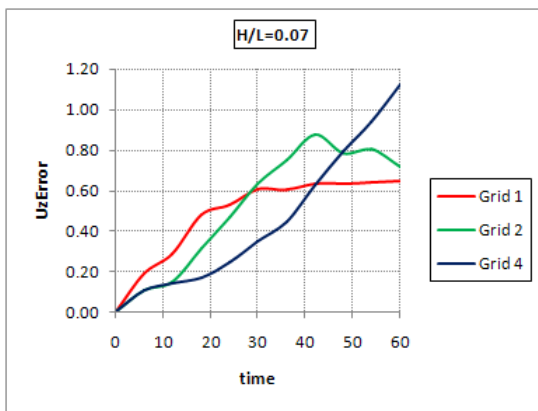
Figure 5.10: $H/L=0.06$



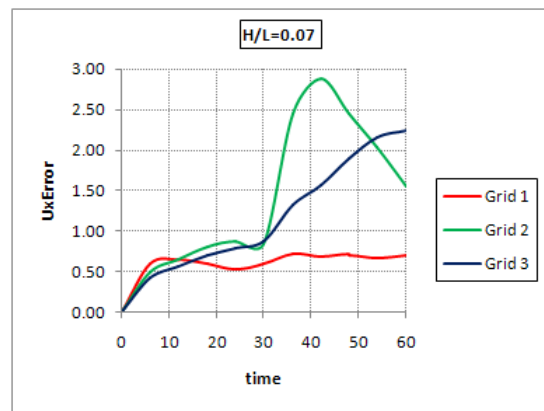
(a) Profile



(b) p



(c) U_z



(d) U_x

Figure 5.11: $H/L=0.07$

Table 5.1: Maximum UzError

Grids per wavelength	Wave steepness	
	0.02	0.04
100	0.8595	0.2389
200	0.3948	0.1371
400	0.1961	0.1197
800	0.1474	0.0811

Table 5.2: Maximum pError

Grids per wavelength	Wave steepness	
	0.02	0.04
100	0.0124	0.0112
200	0.0069	0.0070
400	0.0042	0.0045
800	0.0027	0.0027

included graphs that show the influence of grid resolution on wave profile after around 6 wave period.

5.2.3 Convergence order

In order to get a more quantitative understanding of the behavior of errors in mesh refinement process, we calculate also the order of convergence of the method used for wave modeling in this project. The relationship for defining the error trend is defined simply as follows [15]:

$$E = f(h) - f_{exact} = Ch^p + H.O.T \quad (5.1)$$

In which E denotes the error and f represents the quantity under the study like velocity or pressure in this study and is dependent on a measure of grid spacing, h . The order of convergence is shown by p and C is a constant. Neglecting the higher order terms and taking the logarithm of both sides of the above equation results in:

$$\log(E) = \log(C) + p \log(h) \quad (5.2)$$

In this manner the order of convergence p can be estimated by the reading the slope of the curve of $\log(E)$ versus $\log(h)$. To calculate the order of convergence, the maximum error values for the vertical velocity and the pressure of the four cases with different grid resolution have been selected and are shown tables 5.1 and 5.2. The values are extracted after around one period of wave propagation.

The errors versus the size of the computational grid have been plotted in the logarithmic scale in figure 5.12 and 5.13. For the purpose of the comparison, lines showing a theoretical convergence order of 1 are also added to the plots. Results prove that a considerable improvement in the computational model is achieved while refining the mesh from 100 to 200 grids per wave length. The rate of convergence is around 1 and differs for pressure and vertical component of velocity. According to the order of the numerical schemes which is used in *interFoam*, an order of convergence of 2 should be expected

theoretically in this project. One reason to experience the convergence order of 1 here may be the special way of initialization of the wave field. In fact we used the center coordinate of the computational cells to distinguish between air and water phase. The value of velocity and pressure initialized in this manner are constant throughout each computational cell. This implies that we initialize the wave field in a first order manner. This issue is further explained in the next sections

5.2.4 Unwanted air velocity and premature breaker

Figures 4.6 to 4.7, show example run of the *errorCalculator*. It can be easily understood that always maximum values of errors occur along the interface of the wave. This is generally the same for all three fields of study here, say velocities and pressure. In this manner, all occurrence of maximum errors plotted in the figures locate in more or less the same region as those presented in 4.6 to 4.7. According to the plots shown in figures 5.6 - 5.11, the error application written for this projects gives more or less sensible outputs for pressure and vertical velocity. But due to a large *unwanted* air velocity at the interface, horizontal velocity at that location are disturbed dramatically in such a way that even increasing the number of grids can not improve the maximum computational error and sometimes exacerbates its values. See figure 5.11d for an example.

A good explanation of the reason of this behavior is done by referring to the VOF scheme that is used by *interFoam* for free surface modeling. In this scheme the difference between air and water phase is recognized by the change in the density of the phases, and transport equations are solved for one fluid and all cells. Corresponding density then dictates the value that each phase takes for the velocity and pressure fields. Obviously the density of air is chosen around 1000 times smaller than that of the water and the velocity magnitude in the air phase is far greater than in the water phase. This leaves always a considerable region of air with very high velocity along and around the interface. The high velocity can amount up to 5-6 times of the maximum horizontal velocity of the waves field. This region of *unwanted* velocity is well shown in figure 5.14. It is strongly believed that this issue makes it difficult to evaluate maximum error values (at least for horizontal component of velocity) using *errorCalculator*. Error values in other parts of the field than the interface region, are not influenced greatly by the air velocity, and can be used to get an overall error on computational solution. This requires to prepare a new application that includes the error of all cells in the field in one relationship.

Still another adverse effect of the high air velocity is to enhance breaking of the waves even at the half of their maximum theoretical stable steepness. This means that *interFoam* predicts wave breaking sooner than its real occurrence in deep water. This premature instability predicted by the solver for deep water stokes waves, manifests itself as a roll of the crest along the interface when artificial compression velocity is applied. For the case with $cAlpha=0$ this happens as a sooner than expected diffusion of wave profile. For example for a case with the maximum stable steepness of 0.141, an unstable crest after around two wave period is predicted by *interFoam* at steepness of 0.10, as can be seen in figure 5.15 and 5.16.

In view of the above discussion, the air velocity hinders a proper evaluation of errors on

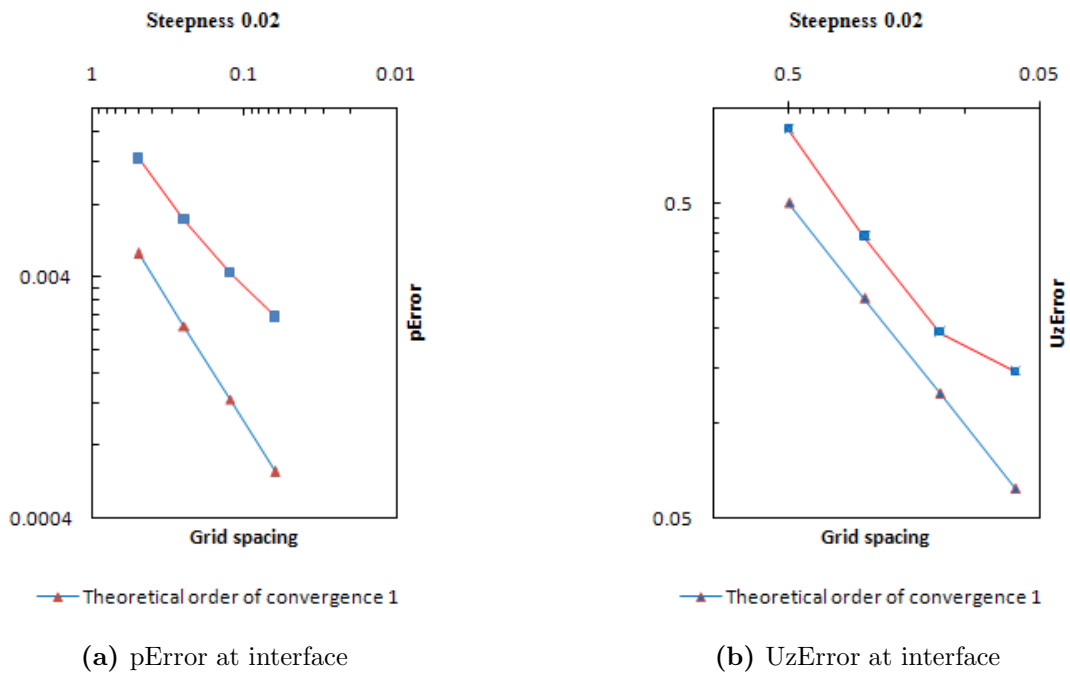


Figure 5.12: Convergence rate, Steepness 0.02

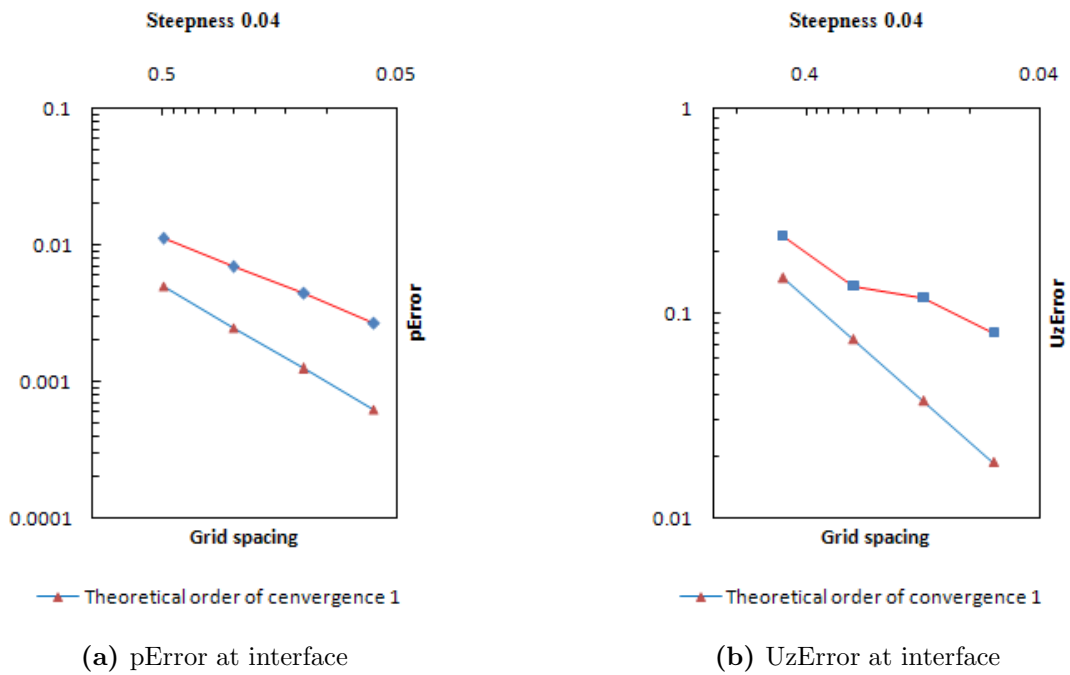


Figure 5.13: Convergence rate, Steepness 0.04

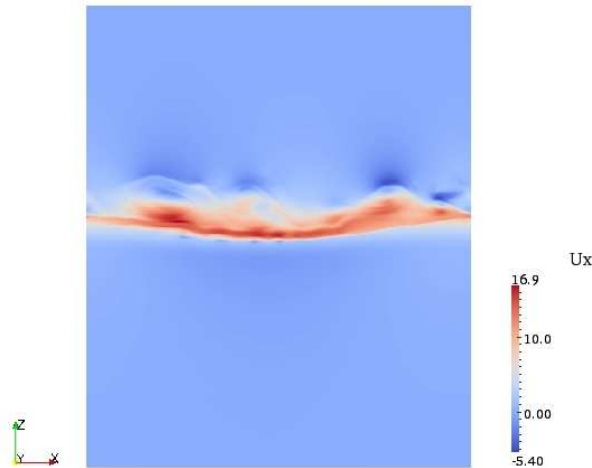


Figure 5.14: Unwanted air velocity over the wave profile

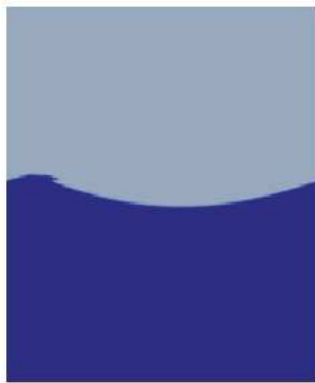


Figure 5.15: Premature Breaker

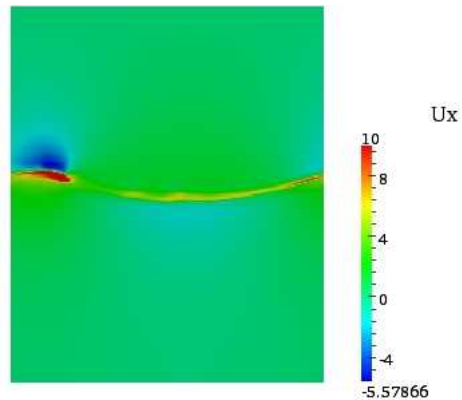


Figure 5.16: Horizontal Velocity

horizontal velocity. Obviously an effective way out of this problem is to get rid of the high velocity in the air phase over the interface as much as possible. Later in this chapter it is tried to propose a method that can enhance the results of *errorCalculator* on horizontal velocities.

Another notable outcome of the study is the oscillation of errors over a considerable part of the run time. This can be seen easily for the waves with lower steepness as their profile diffusion starts quiet late. See for instance figure 5.17a. This oscillation happens as a consequence of the phase lag between computational and analytical solutions. Because of the early dispersion of wave profile, normally it will not happen for steeper waves than 0.06. See figures 5.17b.

5.3 Discussion

Analytical solutions for waves are generally based on a simplified set of fundamental equations, and numerical solution deals with the equations in their whole form. This will leave always an inevitable discrepancy between these two solutions. Obviously the gap will be more close if the numerical solver solves the equations for the conditions for which analytical solutions have been proposed. Stokes waves are basically proposed for deep water conditions, so it is expected to have lower errors in deeper water conditions. So it

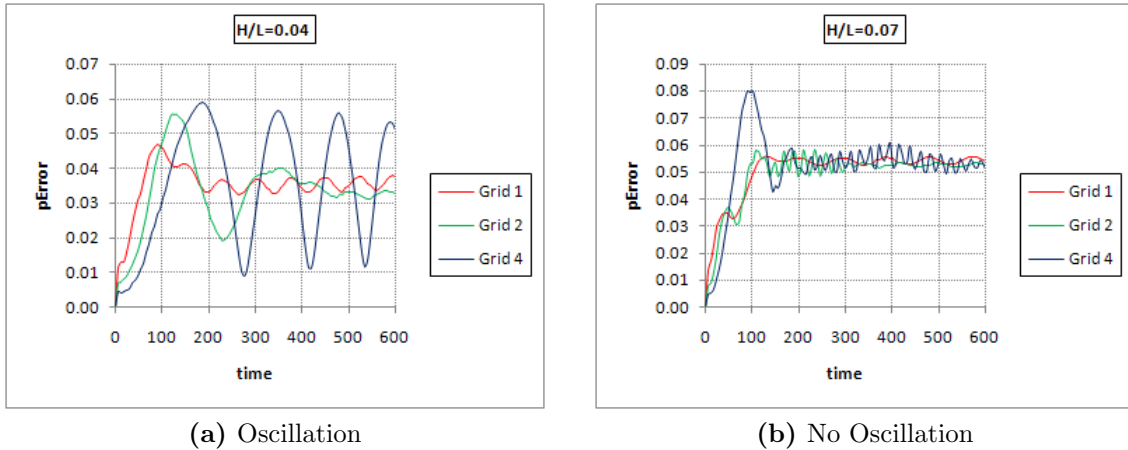


Figure 5.17: Error behavior in long run (100 wave periods)

is interesting to conduct the same grid study but for a higher deep water ratio than 0.6. The study would be more informative if we can get rid of the high air velocity or devise a different method that could circumvent this issue.

The application presented in this chapter for error estimation gives still valuable information for grid resolution influence on the pressure and vertical velocity. Before using the application for horizontal velocity, it is better to remove air velocities from the solution and reduce its adverse effects on the wave profile. Relaxation of solution mentioned above is one of the effective method, but still needs more study to find a proper threshold phase fraction value for each case. This makes error estimation a little time consuming.

A more viable way could be to find the required resolution and corresponding errors, case by case. In this way for each wave a suitable set of $cAlpha$, threshold $alpha1$ and other relevant parameters can be selected in an attempt to have a better error estimation by the *errorCalculator*. In the next sections we presents some remedies that may be applied to reduce the unwanted high air velocity on the wave interface. Moreover, the way of the initialization of the wave field used in this project may cause some adverse effects on the order of convergence. This issue is also shown in section 5.3.3.

5.3.1 Relaxation of air velocity

The total removal of air velocity is both impossible and non-physical, as in reality the air at the immediate distance of the wave gets some motion and velocity. But it is so desirable to have reduced its influence on the wave simulation as far as possible. A quite effective way can be the relaxation of high velocities of the air above the wave profile. The similar approach has been used by [8] in order for reducing the effects of air velocities in the wave tank.

Relaxation can be simply done by replacing the old high velocities with new value of zero before furthering to the next time step. In this manner *interFoam* takes zero velocities for the air phase as the initial condition for the solution of next time level. Air velocity relaxation is implemented by forcing the solver to replace old high velocity in air with the zero at each time step. This requires to find a proper phase fraction value in order to use for distinguishing between air and fluid phase. A value of zero has no efficacy on the relaxation as a major part of the air phase in the numerical solution has values greater than zero. A larger part of the air phase will be captured by selecting higher values for

alpha. In this regard special care should be given as sometimes we will lose the interface at the expense of selecting very large values for the air phase. This means relaxation should be done with a detailed study to find an optimum value for the phase fraction. Figure 5.18 and 5.19 show how this method can considerably improve wave modeling. It gives the horizontal velocity and their corresponding errors for two similar cases but with one of them using air velocity relaxation. The threshold between air and fluid phase is defined by $\alpha_1 = 0.01$, and the velocities are shown after around 9 wave period.

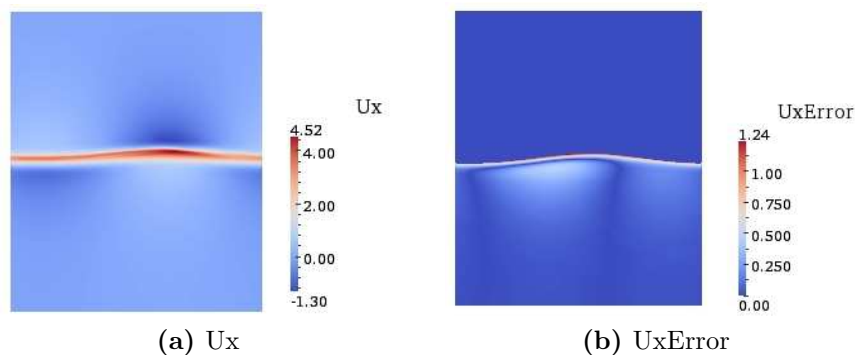


Figure 5.18: Relaxation not implemented

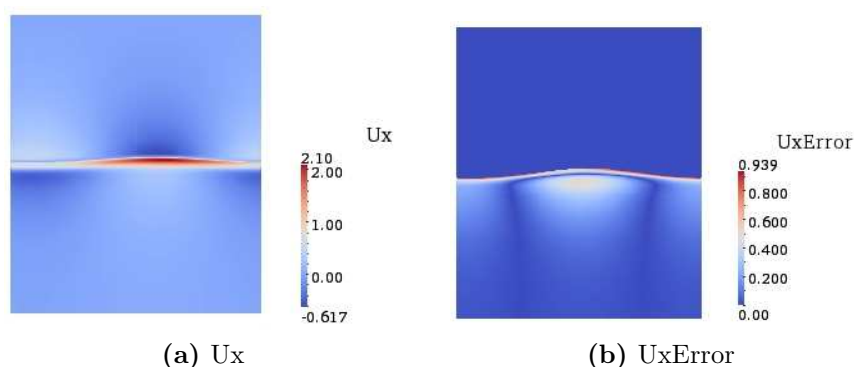


Figure 5.19: Relaxation implemented

Influence of air relaxation will be more considerable for long runs in which air velocity could amount to values up to 5 to 8 times of the maximum horizontal velocity of wave. It is also important to note that very fine grids along the wave profile enables us to select still larger values for $c\alpha$, and can help to prevent premature wave breaking due to high air velocity.

5.3.2 Removing convection in the air phase

Paterson [13] proposed a similar and effective solution to the problem of high air velocity above the free surface. He modifies air convection in the transport equation by multiplying the phase fraction value to the convective term. In this manner convection is kept intact for those parts of the computational cells with the phase fraction value of 1, and reduced for region with lower phase fraction value accordingly. It is obvious that for cells in the air phase a value of zero is multiplied by the convective term, and virtually the non physical effects of the air velocities in that region will be omitted.

5.3.3 Non-Smoothness of initial wave profile

For initialization of wave field by *waveWriter* all cells with their center coordinates lower than those of analytical wave profile are defined as water by setting phase fraction value to 1. As it can be seen from figure below, the use of cell center coordinates for distinguishing of air and water phase produces a staircase shaped interface profile. In this manner the initialized velocity and pressure throughout a computational cell is a constant value and equal to the one at the cell center. This means in order to resolve a close to exact wave profile a fine grid is required for initialization of the field. Put it in another way, this cause a problem for initialization of wave profile using coarse grids as in this case wave profile consists of some wide water columns with constant velocity and pressure at the start. However in this project we alleviate this problem by resorting to the use of a refined mesh around the interface while using wave generation application in next chapter. But still there are remedies that can at the same time save computational expenses and help to initialize a curved profile for the wave field.

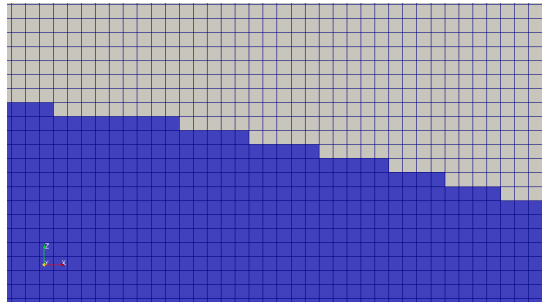


Figure 5.20: Non-Smoothness of wave profile

Chapter 6

Wave Generation and Absorption

6.1 General

Numerical modeling of wave motion, is a very useful tool in the study of forces on ocean structures. Besides all other challenges in a numerical study of wave motion, researchers have to surmount somehow the problems of generation and absorption of the waves in a numerical wave tank. According to the type of desired wave and complexity of the geometry, there are different methods that have been used for wave generation and absorption. For instance, in wave generation the idea of experimental wave makers can be carried over the numerical wave tank. Wave generation in this way requires knowing a given predefined motion of wave maker for each particular wave type. There are analytical solutions proposed for piston and flap wave makers [1] to create linear waves, and can be applied to generate waves by prescribed velocities in the numerical wave tank. Or waves can be generated in another method by inserting internal mass sources in the tank (adding source terms to continuity equation) [9]. In these approaches a relationship is sought between flux of mass over the source region and along the desired wave profile. The source function required for a particular wave type can be found accordingly

Wave damping is as challenging as the wave generation in numerical wave tanks. Since numerical simulation of wave-structure interaction in unbounded spacial domain (like sea water) is only viable by studying the governing equations on a limited region, researchers have to truncate numerical wave tanks by some artificial boundaries that should be introduced in the model. These boundaries are defined in such a way that preclude reflection of scattered outgoing waves back into the computational domain. In contrary to physical boundary conditions like an obstacle in a numerical wave tank, there is not a particular physical law to prescribe an artificial boundary. The same problem in experimental wave tanks can be solved using damping beaches or sponge layers that physically dissipate wave motions by friction and similar mechanisms.

The first and well-known approach to tackle this problem in numerical wave tanks was presented by Sommerfeld on 1964. The idea behind Sommerfeld boundary condition is to force all scatters to act like travelling waves towards infinity without any reflection back into the domain. Simply this can be done by imposition of the first order travelling wave equation at the artificial boundaries as follows:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad (6.1)$$

Where u is the interested property at the boundary and c is the phase velocity of the wave. As far as the celerity of the scattered waves is equal to c , this condition guarantees

the one-way travelling of u with the same phase velocity towards the infinity. Generally it is hardly possible to anticipate a monochromatic wave field throughout the computational region. In effect scattered waves have a wide range of frequency and wavelength, so their phase velocities would be different than something proposed at the start for Sommerfeld boundary condition. This means that the boundary condition will not act as it expected to, and the final solution will not be accurate due to unwanted reflection of scattered waves into computational domain. Moreover the problem is compounded by existence of multi directional wave components in the region, since again the predefined assumptions for the boundary condition are violated. As a result, the attempt to define a general non-reflecting boundary condition for the numerical simulation of wave motion resulted into the development of different methods that have solved the problem to different extent. A comprehensive review of these methods have been explained in a work by [10].

Among them is numerical damping beach that can be implemented for instance by introducing a damping term in dynamic free surface boundary condition in an specified extension in the end of the wave tank. The efficacy of this approach has been proved by many studies like [3], in which a numerical wave tank is developed for generation and absorption of nonlinear and irregular waves.

In this project we make use of a totally different approach called *Relaxation Method* for both generation and damping of the waves in the tank. The method has been widely used by wave scientists and researchers as very efficient numerical wave generation and absorption. For an example see [11]. In following sections the method and its verification is presented.

6.2 Relaxation Method

The idea behind the relaxation method is simple and the scheme is easy to implement. In this approach after each time step in numerical modeling, part of the computational solution of the field is modified according to the desired analytical solution. Generally these parts are called *relaxation zones* and consist of wave generation and absorption regions. Corresponding desired solutions in these regions are a wave theory and the motionless basin. Since the desired solution is forced over the relaxation zones, they should be extended in the wave tank in a way that abrupt changes in computational solution is prevented. Normally in wave generation and absorption by relaxation method, relaxation zones have an extension around 1 to 2 wave length. Figure 6.1 gives schematic representation of the relaxation zones in the wave tank in this project.

Zone I is defined for relaxation of solution towards an analytical wave theory. This is done by neglecting totally the computational values there, and ramping up wave kinematics according to a *relaxation function*. In fact zone I acts like a wave generation region. Relaxation of solution in zone I is implemented according to the following relationships:

$$\begin{aligned}
 U_{relaxed} &= C(x)U_{analytical} + (1 - C(x))U_{computational} \\
 P_{relaxed} &= C(x)P_{analytical} + (1 - C(x))P_{computational} \\
 \alpha_{relaxed} &= C(x)\alpha_{analytical} + (1 - C(x))\alpha_{computational}
 \end{aligned}
 \tag{6.2}$$

where U represents two velocity components, P is the field pressure and α denotes phase fraction value. Relaxation coefficient C has a value of zero at the start of the zone I and ramps up steadily towards one at the end. Computational solution in zone I is that of a still basin with zero velocities and hydrostatic pressure that are gradually modified to

the wave analytical velocities, pressure and profile. Variation of the relaxation coefficient along the zone is prescribed by the relaxation function that will be presented in the next section.

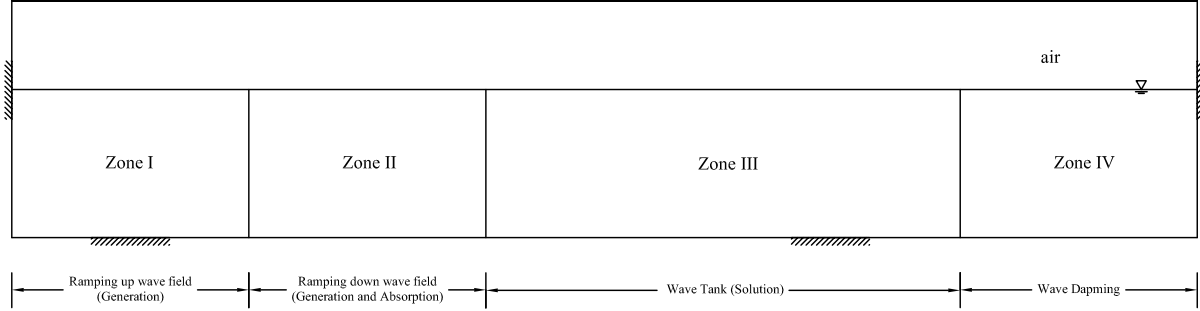


Figure 6.1: Relaxation zones

In zone II, computational solution gradually replaced by an analytical wave solution at the start to the pure computational one at the end of second relaxation zone according to the following relationships:

$$\begin{aligned}
 U_{relaxed} &= C(x)U_{analytical} + (1 - C(x))U_{computational} \\
 P_{relaxed} &= C(x)P_{analytical} + (1 - C(x))P_{computational} \\
 \alpha_{relaxed} &= C(x)\alpha_{analytical} + (1 - C(x))\alpha_{computational}
 \end{aligned} \tag{6.3}$$

C has a value of one at the start of the zone and reaches to zero at the end. This ensures a mild transition from analytical wave filed to the working basin in zone III. In this way all back scattered waves from bodies in the tank or from changes in tank bathymetry are damped gradually along the length of zone II and before reaching to zone I. This is a powerful method to keep wave generation zone away from the influence of back scattered waves.

There is no relaxation in zone III, and computational solutions in this region are kept intact to represent the wave tank working basin.

In order to absorb all outgoing waves, relaxation zone IV is dedicated at the end of the wave tank to gradually modify the computational solution of the basin to the zero. Relaxation in this region is done by:

$$\begin{aligned}
 U_{relaxed} &= C(x)U_{computational} + (1 - C(x))U_{analytical} \\
 P_{relaxed} &= C(x)P_{computational} + (1 - C(x))P_{analytical} \\
 \alpha_{relaxed} &= C(x)\alpha_{computational} + (1 - C(x))\alpha_{analytical}
 \end{aligned} \tag{6.4}$$

In which C starts from one and amounts down to zero at the end of relaxation zone. Analytical solution is that of a still basin with zero velocities and hydrostatic pressure. In fact all generated waves are dissipated gradually along the zone IV, and a motionless basin remains finally at the end.

Generally relaxation towards desired solution is implemented simply at each time-step by "preprocessing" the computational solutions before going further to the next time-step. So in this way the input to the governing equations are changed according to the desired values. Tho whole scheme is easily extendable to three dimensional wave tanks and also for directional and irregular wave generation.

6.2.1 Relaxation Functions

Generally the type of the relaxation functions is highly dependent on the numerical scheme used for the wave tank, and requires a dedicated study to find a proper function that allows mild transition between relaxation zones. In this project we make use of the relaxation functions proposed by [11] for wave generation and damping. For the purpose of wave generation in zone I, the relaxation function is defined as:

$$C(x) = -2x^3 + 3x^2 \quad (6.5)$$

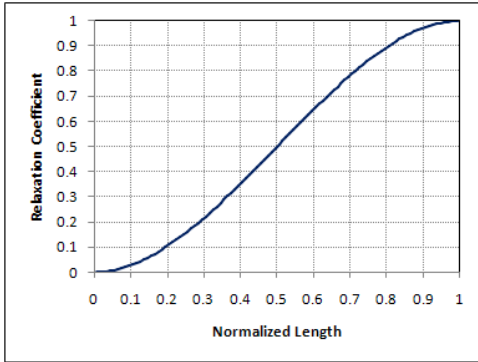
With the change of the argument of the function to $1 - x$, we reach to the relaxation function suitable for absorption of back scattered waves in zone II as follows:

$$C(x) = -2(1 - x)^3 + 3(1 - x)^2 \quad (6.6)$$

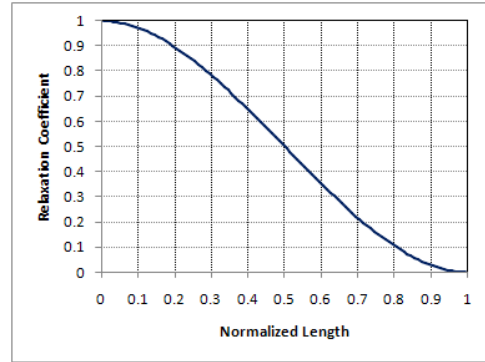
And in zone IV the waves are damped according to the following equation:

$$C(x) = 1 - x^6 \quad (6.7)$$

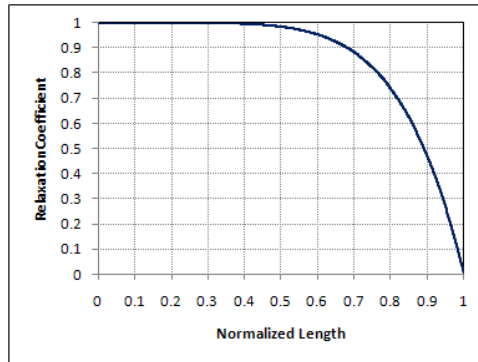
In all equations x denotes the normalized horizontal coordinate along the length of the zone and ranges between 0 and 1 for start and end of the relaxation zone respectively. Graphs of the relaxation functions are presented in figure 6.2.



(a) $-2x^3 + 3x^2$



(b) $-2(1 - x)^3 + 3(1 - x)^2$



(c) $1 - x^6$

Figure 6.2: Relaxation functions

6.3 waveFoam

All procedures mentioned above has been written as a C++ header file called *relaxation.H* and is added to the source code of *interFoam*. The modified solver christened as *waveFoam* enables us to generate and dissipate waves in the wave tank. Moreover an application is prepared (*tankData*) that should be used for giving the wave information and relaxation variables to the solver. By this application we have control over the extension of the relaxation zones in the wave tank and can check wave steepness against breaking according to Fenton expression.

6.4 Wave generation test cases

In order to show the performance of the procedure applied here for the relaxation method, some example run of the numerical wave tank is presented in the coming sections. The mesh used for wave tank modelings is a structured grid created by *blockMesh*. A close-up of the mesh is presented in figure 6.3.

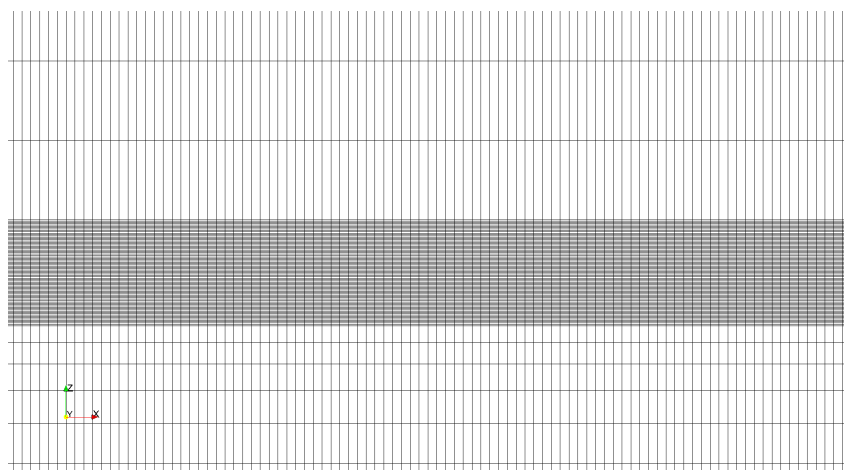


Figure 6.3: wave tank mesh

6.4.1 Linear progressive wave

In the all following runs *blockMesh* is used as the mesh generator. Grid requirements are according to the result of the Chapter 4 that shows number of grids per the wave length for each particular steepness. Generally based on the steepness selected here, between 200 to 400 grids per wave length deemed to be sufficient in order to resolve wave information in the horizontal direction. The mesh size in the vertical direction and over a limited region close to the free surface is chosen to be equal or more than the size in horizontal direction. Far away from the free surface, use of grid with lower resolution is always possible.

First we set a case for a progressive linear wave of steepness 0.02 in the deep water condition $d/L = 1$. Relaxation zones I and II have a extension of one wave length each,

and zone IV extended to three wave lengths. It is important to note that for adequate wave absorption, the scheme used in this project requires at least three wave length extension of zone IV.

Figure 6.4 shows progression of the free surface profile for one wave period along the length of wave tank. It shows how the wave is ramped at the start and dissipated at the end. Although in this test it was tried to remain at the small amplitude range of linear waves, there are some small deviancy between the analytical profile at the start of the tank and the computational ones along the basin. The reason can be sought in the fact that the solver treats the fundamental equations in their whole, whereas the linear theory used for wave generation in this test is based on a simplified set of governing equations. Figure 6.5 also shows a snapshot of wave profile.

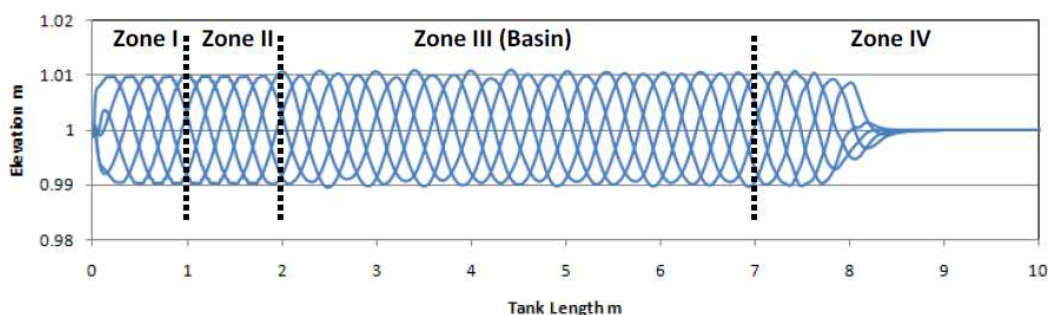


Figure 6.4: One period succession of linear wave profile

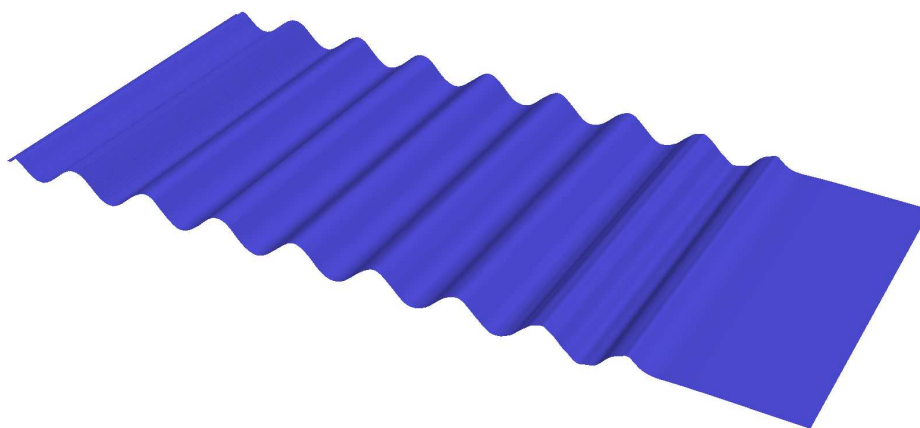


Figure 6.5: A linear wave (Vertical scale exaggerated 15 times)

6.4.2 Linear standing wave

In the next test of the relaxation scheme, a linear standing wave is simulated to show mainly how is the performance of zone II in preventing backscattered waves from entering the wave generation zone. In this case zone IV has been totally removed in order for having a fully reflective wall at the end of the wave tank. This case is easily incorporated in the *tankData* application by entering zero as the extension of zone IV. Zone I extends one wavelength, but in order to have a more mild transition between zone II and the

working basin, zone II has two wave length extension in this case. It is reasonable to set the tank in this manner as the extent of the back scattered waves is remarkable due to full reflection of incident waves backwards.

Incident wave has a steepness of 0.02 and length of 1 m. It propagates in a deep water condition with $d/L = 1$ Figure 6.6 presents four successive profiles of the free surface along the tank. As it is expected from the linear theory, along the tank a standing wave with a height of two times of the incident wave is created. Appearance of the nodes and antinodes at multiples of $L/4$ (points with fully constructive and destructive superposition) is easily recognizable from the graph. According to the figure, back reflected waves are not allowed to enter the wave generation zone and are damped gradually backward along the zone II. As it mentioned before, this zone efficiently can be used while using the tank to study wave-structure interaction. See also figure 6.9 for a snapshot of the tank in this case.

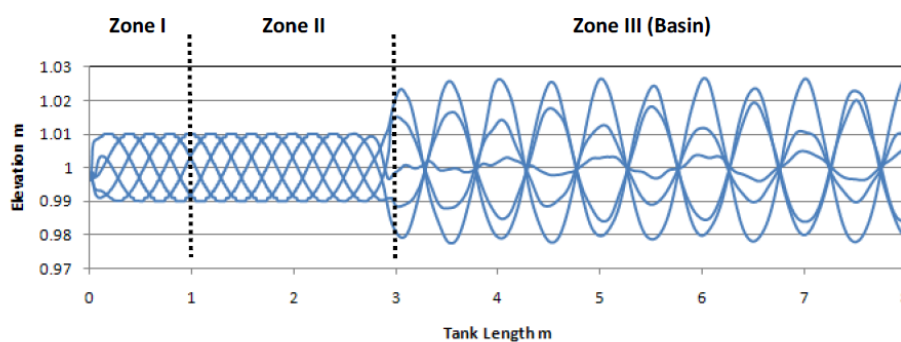


Figure 6.6: One period succession of standing wave profile

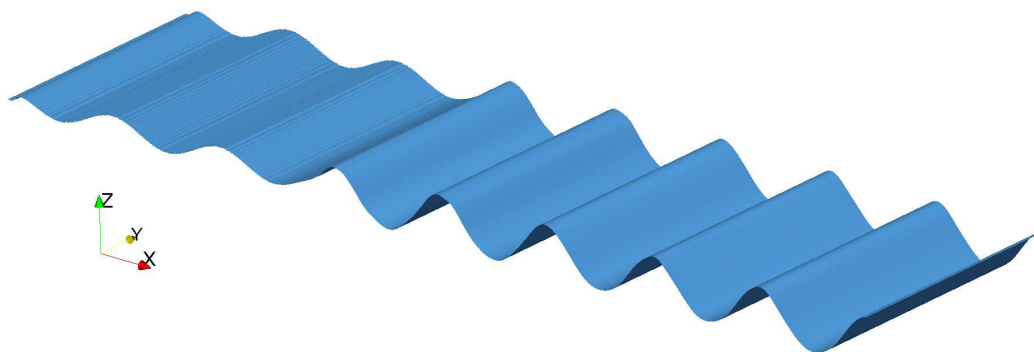


Figure 6.7: Linear incident and Standing wave (Vertical scale exaggerated 8 times)

The difference between the expected analytical and computational standing wave profile is more remarkable in comparison with the linear progressive case in the previous section. Again the same argument regarding the way of treatment of the basic equations can be used to justify the behavior, as in the standing wave case the solver accounts for nonlinearities in the solution which results in some overshoots and undershoots in the superimposed wave profile.

6.4.3 Nonlinear progressive wave

In another test, we used the *waveFoam* to generate a nonlinear progressive fifth order Stokes wave in the wave tank. Like the case with linear wave, a relaxation zone I and II extends one wavelength each. Dissipation of outgoing waves is done along the Zone IV that has an extension of three wavelengths. The wave has a length of 2.5 meter and is generated in the water depth of 30 cm and a tank of 25 meter. Plots of successive wave profile during one wave period is shown in figure 6.8. The minor deviation mentioned in two earlier sections can also be noted here. In order to better present the results of nonlinear wave generation by *waveFoam* a snapshot of the wave profile is given in figure 6.9.

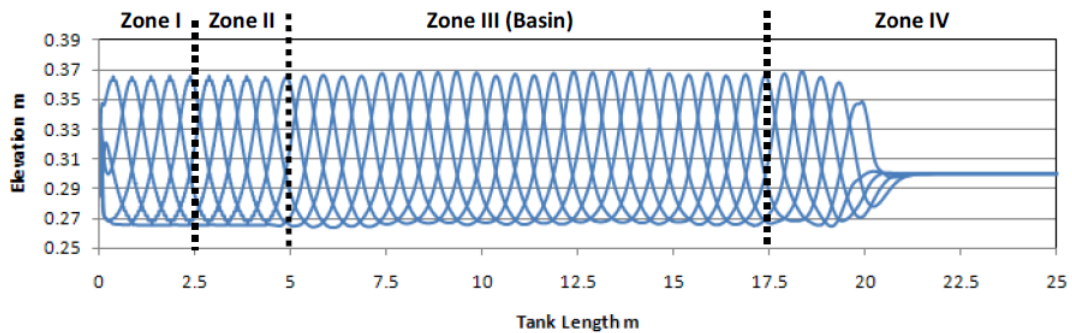


Figure 6.8: One period succession of Stokes wave profile

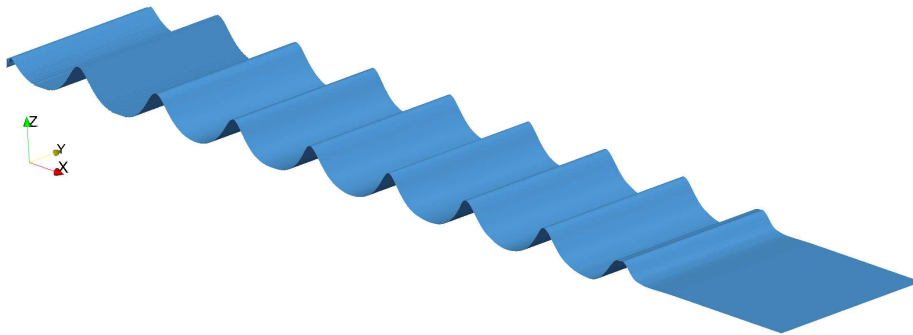


Figure 6.9: A progressive Stokes wave (Vertical scale exaggerated 8 times)

Chapter 7

Verification and validation

7.1 Test description

After preparation of the method for wave generation and absorption in OpenFOAM, the Whalin experimental test [12] was considered to be run in this project to validate the performance of the numerical wave tank created by *waveFoam*. The well-known Whalin experiment is commonly used by the researchers in the wave field to verify the accuracy of the developed numerical method for the wave tanks, for instance refer to [11].

The experiment has been carried out in a wave flume to study the effects of nonlinearities in the shoaling of water waves while propagating over a half submerged cylinder, see figure 7.1. The test is performed by generation of small amplitude linear waves at one side of the tank and damping of the outgoing waves in the end. The measurements done by Whalin in the test reveals contributions due to nonlinearities in the transformed wave profile caused by cylindrical shoal. To validate the numerical wave tank researchers try to prove that the spectrum of the transformed wave at the different locations along the centerline of the tank shows sign of the existence of some other frequencies than that of linear wave at the wave maker side.

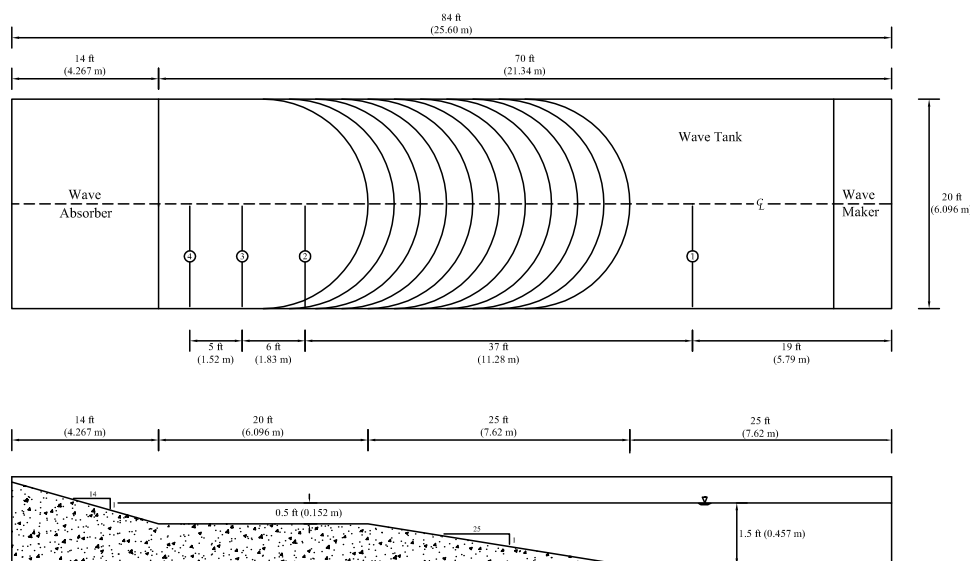
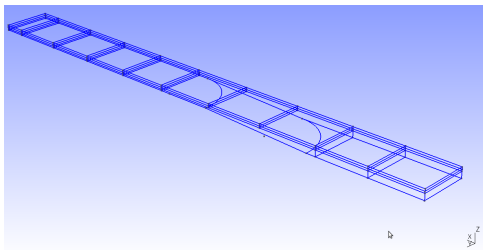


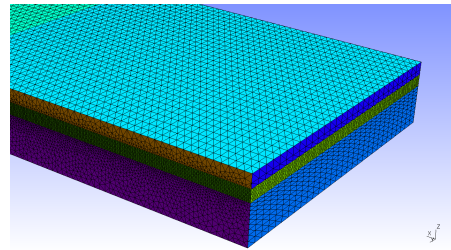
Figure 7.1: Whalin Experiment Set-up

7.2 Mesh generation

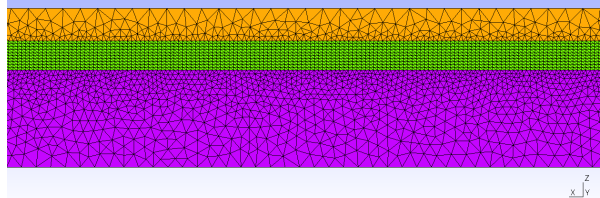
Up to this point in the project we made use of OpenFOAM *blockMesh* application to generate a hexahedral mesh for simple rectangular geometries. Whalin test case is a three dimensional modeling of wave tank and requires considerable amount of computational cells. Since *blockMesh* has a limited capacity for defining mesh refinements in arbitrary regions in model, we tried to utilize an open source mesh generator called *gmsh* [14] for creating a computationally efficient grid for the Whalin test case. It was tried to generate a refined mesh around the interface and according to the grid requirements set forth in chapter 5. The prepared mesh is a hybrid (structured-unstructured) grid of tetrahedras, see figures below. As can be seen a fine grid over a region with the extension of wave height has been defined to resolve the wave profile. Note also that due to symmetry in the problem half of the wave tank is included in the model.



(a) tank overview



(b) mesh in y direction



(c) mesh in x-z plane

Figure 7.2: Whalin test mesh

7.3 Problems with OpenFOAM 1.6

After mesh generation we run *waveFoam* to conduct the Whalin test, but unexpectedly the run process stopped after a short time period. In order to investigate the problem, a simplified Whalin test with highly reduced tank width was created using *blockMesh*. The run showed a very strange behavior of the surface profile in the wave tank. See figure 7.3. There is a huge undulation in the water surface where it is expected to be still and smooth as the waves not yet reached there. The attempt for finding the problem took a considerable time, and finally it was turned out that there are some inadequacies in *interFoam* while dealing with cases with complicated and distorted meshes like in the Whalin test. This idea was more reinforced by the release notes of the then newest version of OpenFOAM, 1.7, [20] which recognized this problem and indicated specifically that it

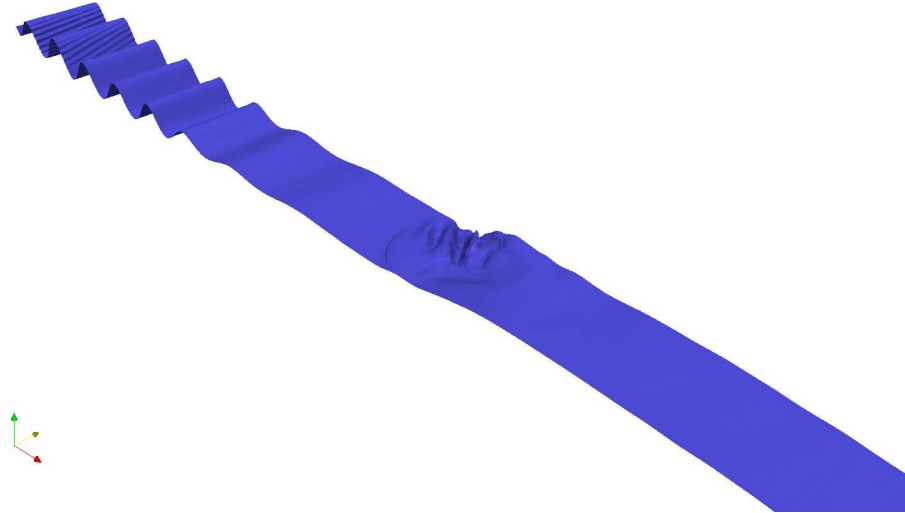
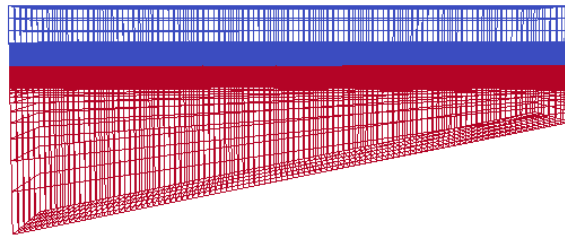
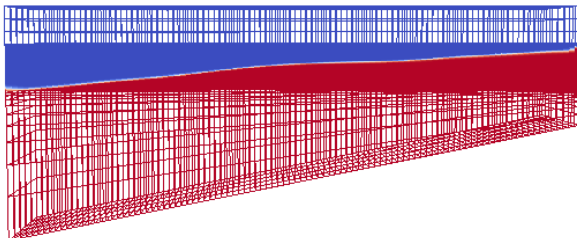


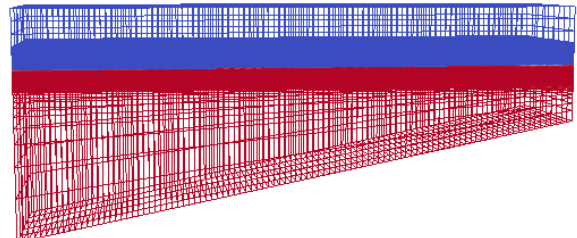
Figure 7.3: Whalin test failure in OpenFoam-1.6.x



(a) Initial still basin



(b) After 2 seconds-OpenFOAM-1.6



(c) After 2 seconds-OpenFOAM-1.7

Figure 7.4: Inadequacy of OpenFOAM-1.6

has been dealt with to some extent in the upgraded package of 1.7. To explain the problem at least visually two similar cases were run by *interFoam* of the old and new version of OpenFOAM. The case is simply a still basin with the water resting on an inclined bed. Models are meshed by *blockMesh*, and there is no initial velocity defined for the model. The expectation is to see a still water basin after running the *interFoam*. Figure 7.4a shows the still basin which has been created by three blocks with different resolutions using *blockMesh*. As can be seen from figure 7.4b, OpenFOAM-1.6 gives totally strange results for the fluid on the inclined bed. The velocity plot in this case shows a very high velocity along the bed, and this is why the initially still interface becomes distorted and moves upward. This unexpected behavior causes the Courant number to become very large to such an extent that the corresponding reductions in the time steps virtually stops

the process from running. Figure 7.4c represents the result of the *interFoam* run on the exactly similar case but using 1.7 package. It is obvious from figure that OpenFOAM-1.7 is able to predict the simple expected results.

In the same manner, due to complexity of the mesh, and mainly due to existence of the inclined bed in the Whalin numerical wave tank, *waveFoam* that is basically a partially modified *interFoam* solver could not run successfully for validation of the relaxation method created in this project.

It seems that the use of OpenFOAM-1.7 for validation by Whalin tank is highly promising as it has been shown by a simple example above.

Chapter 8

Conclusions and recommendations

The thesis has been done in an effort to examine the use of OpenFOAM for the numerical wave generation from a new perspective and in a more detail than the previous works in this field. The research started with the aim of finally applying the prepared numerical wave tank for the study of response of floating structures to the ocean waves. The main achievements of the thesis are the convergence study and the implementation of the relaxation method for wave generation and absorption in OpenFOAM.

The results of grid study in Chapter 5, proved that the computational demands for the wave generation by *interFoam* is rather high. The order of convergence for wave generation in OpenFOAM varies around 1 and is lower than the expected value for the second order discretization schemes that is equal to 2. This is mainly due to the first order type initialization of the wave field in the model. The study showed also that the grid resolution is highly dependent on the wave steepness, and the steeper is the wave the higher is the number of required grids per the wavelength. Normally depending on the steepness, and based on the method used in this project, between 200 to 400 grids are needed for the wave generation in OpenFOAM. This implies use of grid refinement at least around the interface.

In this manner the high mesh resolutions would be sufficient just for a limited part of the computational domain, and because of this structured grids which can be created by *blockMesh* will not be very efficient. For the large and three dimensional wave generation like that in the Whalin experimental test, use of unstructured grids is an efficient way of creating the numerical wave tank in OpenFOAM.

It is of high importance to note that just maximum values of errors that occur normally at the wave interface are included in the process of error calculation. So computational errors in the wave medium itself is well below than those along the wave profile. This may be of importance for the researches in which the interest is mainly on the wave hydrodynamic pressure like study of wave medium on the pore pressure and the scouring in the sea bed.

In the wave generation and absorption part, it has been found that a simple and efficient implementation of relaxation method can be performed just by adding a header file containing the analytical wave solutions to the existing source code of the solver. Depending on the type of desired wave, the analytical solutions can be easily defined in the mentioned header file. The implementation of relaxation method by the use of relaxation functions prescribed by [11], showed that an efficient wave absorption can be achieved by setting the order of the polynomial equation in the damping zone to 6. Moreover multiple tests with different lengths for relaxation zone IV, revealed that damping region

in the wave tank should have an extension of at least three wave lengths. Considerable reflections have been observed for the cases in which zone IV had 1 or 2 wavelength extension. Generally, use of zero for $cAlpha$ parameter is recommended for wave generation by relaxation method as we have seen some unexpected wave reflections back in to wave tank for those cases with $cAlpha$ equal to 1. The study showed also that a reasonable wave generation in zone I and an efficient absorption of back scattered wave in zone II can be achieved by extending the corresponding relaxation zones to just one wavelength each.

Unfortunately, lack of time and unpredicted problems with OpenFOAM-1.6 forced the project to be culminated at its current stage. This means still there are a considerable open subjects that can be studied further in order to have a more efficient wave generation by OpenFOAM. Among them following issues are more highlighted as the probable future studies on this field:

- Implementation of Whalin test in OpenFOAM-1.7.x for validation of relaxation method used for wave generation.
- Removing the high velocities in the air phase by the methods mentioned earlier in this report. They include omitting the convective term in the equation by multiplying it with the phase fraction value and relaxation of air velocity at each time step
- Preparing a new error estimation application that takes in to account computational errors at other regions than wave interface.
- Devising a new method for defining the wave field by the use of other parameters than cell center coordinates. This helps to avoid a jagged wave profile at the relaxation zones I and II.

Bibliography

- [1] Robert G. Dean, Robert A. Dalrymple, (1991), *Water Wave Mechanics for Engineers and Scientist*, World Scientific.
- [2] L. Skjelberia, J. Hendrickson, (1961), *Fifth Order Gravity Wave Theory*, Proc. 7th Conf. Coastal Eng., ASCE, pp. 184-196.
- [3] Ohshima, T. and Nadaoka, K. (1991), *Development of a numerical wave tank for analysis of nonlinear and irregular wave field*, J. Fluid Dynamics Research, 8, pp. 231-251
- [4] Versteeg, H. K. and Malalasekera, W. (2007), *An Introduction to Computational Fluid Dynamics, The Finite Volume Method*, Pearson Education.
- [5] Fenton, J. D. (1990), *Nonlinear wave theories*. In: *Le Mehaute, B; Hanes, D. M. (eds): The Sea*. John Wiley & Sons.
- [6] Banner, M. L, Peregrine, D. H (1993), *Wave Breaking in Deep Water*, Annu. Rev. Fluid Mech. 1993, 25 : 373-97.
- [7] H. Rusche (2002), *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions*, PhD Thesis, Imperial College of Science, Technology & Medicine, Department of Mechanical Engineering.
- [8] X. Liu (2008), *Numerical Models for Scour and Liquefaction Around Object Under Currents and Waves*, PhD Thesis, University of Illinois at Urbana-Champaign.
- [9] P. Lin and Philip L.-F. Liu (1999), *Internal Wave-Maker for Navier-Stokes Equations Models*, Journal of Waterway, port, Coastal and Ocean Engineering, Volume 125, Issue 4, pp. 207-215.
- [10] J. E. Romate (1992), *Absorbing boundary conditions for free surface waves*, Journal of Computational Physics, 99, pp. 135-145.
- [11] Allan P. Engsig-Karup (2006), *Unstructured nodal DG-FEM solution of high-order Boussinesq-type equations*, PhD Thesis, Technical University of Denmark, Department of Mechanical Engineering, Section of Coastal, Structural and Maritime Engineering.
- [12] R.W. Whalin (1971), *The limit of applicability of linear wave refraction theory in a convergence zone*, Research Report H-71-3, U.S. Army Corps of Engineers, WES, Vicksburg, MI, 1971.
- [13] E. Paterson (2008), *Multiphase and free surface flow simulations*, Third OpenFOAM workshop, Politecnico di Milano, Milan, Italy.

- [14] C. Geuzaine and J.-F. Remacle (2009), *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*, International Journal for Numerical Methods in Engineering, Volume 79, Issue 11, pages 1309-1331.
- [15] P. J. Roache (1998), *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, New Mexico.
- [16] G. G. Stokes(1847), *On the Theory of Oscillatory Waves*, Proc. Camb. Philos. Soc, Vol. 8, pp. 441-455.
- [17] J. R. Morison, P. O'Brien, J. W. Johnson and S. A. Schaaf (1950), *The Forces Exerted by Surface Waves on Piles*, Petrol. Trans., AIME, Vol. 189.
- [18] P. M. Gerhart, R. J. Gross and J. I. Hochstein (1992), *Fundamentals of fluid mechanics*, Addison-Wesley Publishing Company.
- [19] C. W. Hirt and B. D. Nichols (1981), *Volume of fluid (VOF) method for the dynamics of free boundaries*, Journal of Computational Physics, 39, pp. 201-225
- [20] <http://www.openfoam.com/archive/1.7.0/docs/release-notes.php> *The open source CFD toolbox*.
- [21] <http://www.openfoam.com>, *The open source CFD toolbox* and *OpenFOAM User Guide*.
- [22] <http://openfoamwiki.net>, *OpenFOAM Wiki Page*.