



Norwegian University of  
Science and Technology

# Numerical study of surface waves on a shear current

Waves behind a surface obstacle

**Ørjan Kvamme**

Master of Science in Mechanical Engineering

Submission date: March 2018

Supervisor: Simen Andreas Ådnøy Ellingsen, EPT

Co-supervisor: Reidar Kristoffersen, EPT

Norwegian University of Science and Technology  
Department of Energy and Process Engineering



EPT-M-2017-45

**MASTER THESIS**

for

Student Ørjan Kvamme

Autumn 2017

Numerical study of surface waves on a shear current: waves behind a surface obstacle

*Numerisk studie av overflatebølger på en skjørstrøm: bølger bak et overflatelegeme***Background and objective**

In coastal waters, surface waves typically interact with currents which vary with depth. Such shear currents will affect wave properties. The research group of Ass. Prof. S. Å. Ellingsen has published a broad range of theoretical work on this subject. The project will aim at studying this system numerically including full nonlinear internal dynamics, for comparison with linear and weakly nonlinear theory, for the geometry of waves forming behind an obstacle on the surface, a “ship”. A large part of the project will be the development and implementation of appropriate absorbing boundary conditions. Conventional absorbing boundary conditions are developed for monochromatic waves, whereas we wish to study initial value problems containing the full spectrum of wavelengths. The plan is that corresponding measurements of the flow field in the same geometry will be performed by a laboratory experiment by another master student. Provided such results are available early enough, a comparison with simulations will be made.

**The following tasks are to be considered:**

The following list is highly ambitious, to be considered a best-case scenario:

1. Developing and implementing appropriate boundary conditions for surface waves, which can absorb a continuous spectrum of waves in 2D geometries.
2. Extend the 2D treatment to including a current of constant shear (linear depth-dependence)
3. Study 2D “ship wave” problems with linear shear current at the inlet and a surface obstacle (for example a cylinder).
4. Analyse the sub-surface structures beneath a nonlinear surface wave.
5. Generalise inflow conditions to allow more general shear flows, and study differences from linear current case.

-- ” --

Within 14 days of receiving the written text on the master thesis, the candidate shall submit a research plan for his project to the department.

When the thesis is evaluated, emphasis is put on processing of the results, and that they are presented in tabular and/or graphic form in a clear manner, and that they are analyzed carefully.

The thesis should be formulated as a research report with summary both in English and Norwegian, conclusion, literature references, table of contents etc. During the preparation of the text, the candidate should make an effort to produce a well-structured and easily readable report. In order to ease the evaluation of the thesis, it is important that the cross-references are correct. In the making of the report, strong emphasis should be placed on both a thorough discussion of the results and an orderly presentation.

The candidate is requested to initiate and keep close contact with his/her academic supervisor(s) throughout the working period. The candidate must follow the rules and regulations of NTNU as well as passive directions given by the Department of Energy and Process Engineering.

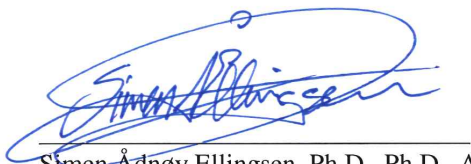
Risk assessment of the candidate's work shall be carried out according to the department's procedures. The risk assessment must be documented and included as part of the final report. Events related to the candidate's work adversely affecting the health, safety or security, must be documented and included as part of the final report. If the documentation on risk assessment represents a large number of pages, the full version is to be submitted electronically to the supervisor and an excerpt is included in the report.

Pursuant to "Regulations concerning the supplementary provisions to the technology study program/Master of Science" at NTNU §20, the Department reserves the permission to utilize all the results and data for teaching and research purposes as well as in future publications.

The final report is to be submitted digitally in DAIM. An executive summary of the thesis including title, student's name, supervisor's name, year, department name, and NTNU's logo and name, shall be submitted to the department as a separate pdf file. Based on an agreement with the supervisor, the final report and other material and documents may be given to the supervisor in digital format.

- Work to be done in lab
- Field work

Department of Energy and Process Engineering, 4. September 2017



Simen Ådnøy Ellingsen, Ph.D., Ph.D., Associate Professor  
Academic Supervisor

Research Advisor/second supervisor: Reidar Kristoffersen

---

# Summary

This master's thesis has made investigation into waves and currents interaction numerically by the use of an oscillating cylinder as a wave-maker. Three cases were investigated: still water, uniform flow and linear shear current. A small parameter study was performed to compare OpenFOAM's damping option to other existing wave damping tools in other competing software. The results showed that, while the damping parameter  $f$  can be estimated through linear scaling law, this serves at best as a first guess. The wave damping was applied in all cases, proving a helpful tool for damping also on a shear current. The still water case validated the use of the oscillating cylinder as a wave-maker, where linear surface waves were created and checked against the dispersion relation. The amplitude of the waves produced by the cylinder proved to be around half of the excitation amplitude. Both the uniform flow case and the linear shear case looked into the non-linear effects of a moving oscillating cylinder at an excitation frequency near to the Doppler resonance frequency,  $\tau = 1/4 = \zeta U/g$ . Non-linearity was in both cases proven to occur upstream of the cylinder. In the uniform flow case, it caused wave breaking and overturning, while linear waves were produced downstream. In the linear shear case, a positive vorticity  $S$ , was shown to assist the waves downstream, while it opposed the waves upstream. This led to smaller wavelengths downstream and larger wavelengths upstream compared to a case of zero vorticity. The dispersion relation showed good concurrence both downstream and upstream with non-linear corrections of the dispersion relation upstream. Downstream, in the linear regime, the relative error for the wavelength was shown to be 4%, 6% and 3.7%, for the respective cases.

---

# Sammendrag

Denne master oppgaven har undersøkt bølge- og strøm-interaksjoner numerisk ved hjelp av en oscillerende sylinder som bølgekilde. Tre tilfeller ble undersøkt: vann i ro, uniform strømning og linær skjærstrøm. En liten parameterstudie ble gjennomført for å sammenligne OpenFoams dempefunksjon med andre eksisterende dempefunksjoner i andre konkurrerende programvarer. Resultatene viste at å estimere dempekonstanten  $f$  ved en linær skaleringslov, kun kan brukes som et første forslag. Bølgedemping ble brukt i alle studier og ble bevist som hjelpsomt verktøy for bølgedemping, også ved skjærstrøm. Vann i ro studiet validerte bruken av en oscillerende sylinder som bølgekilde, der linære bølger ble sammenlignet ved hjelp av dispersjons-relasjonen. Amplituden til bølgene ble vist til å være rundt halvparten av den eksiterte amplituden til cylinderen. Både det uniforme og i det linære skjærstrøm tilfellet undersøkte ikke-linære effekter av en bevegende sylinder ved en eksitasjon frekvens nært Doppler resonans frekvensen,  $\tau = 1/4 = \zeta U/g$ . Ikke-linearitet oppstrøms for sylinderen ble bevist i begge tilfeller. For den uniforme strømningstilfellet resulterte dette i bølge brytning og -velt, mens linære bølger ble produsert nedstrøms. I det linære skjærtilfellet, bidro en positiv virvling  $S$ , til å motarbeide bølger oppstrøms og hadde motsatt virkning nedstrøms. Dette ga mindre bølgelengder nedstrøms og større bølgelengder oppstrøms sammenlignet med ingen virvling. Dispersjons-relasjonen ga god overenstemmelse både nedstrøms- og oppstrøms med ikke-linær korreksjon. Nedstrøms, i det linære regime, ble den relative feilen vist å være 4%, 6% og 3.7% i de respektive sakene.

# Table of Contents

<b>Summary</b>	<b>i</b>
<b>Sammendrag</b>	<b>ii</b>
<b>Table of Contents</b>	<b>iv</b>
<b>List of Tables</b>	<b>v</b>
<b>List of Figures</b>	<b>vii</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Background . . . . .	2
1.2 Motivation and Objectives . . . . .	3
1.3 Outline . . . . .	4
<b>2 Theory</b>	<b>5</b>
2.1 Wave theory and terminology . . . . .	5
2.2 Free surface flows . . . . .	6
2.2.1 Free surface boundary conditions . . . . .	6
2.3 Surface waves . . . . .	7
2.3.1 Linear surface waves . . . . .	7
2.3.2 Non-linear surface waves . . . . .	10
2.4 Oscillating source . . . . .	10
2.5 Water wave regimes . . . . .	12
2.6 Numerical methods . . . . .	13
2.6.1 Finite Volume Method . . . . .	13
2.6.2 Spatial and Temporal discretization . . . . .	14
2.6.3 PIMPLE - Algorithm . . . . .	18
2.7 Numerical Representation of Free Surfaces . . . . .	20
2.7.1 Arbitrary Lagrangian Eulerian - ALE . . . . .	20
2.7.2 Volume of Fluid - VOF . . . . .	22
2.7.3 Level set methods . . . . .	23
2.8 Absorbing Boundary Conditions . . . . .	24
2.8.1 Transparent Boundaries . . . . .	25
2.8.2 Zonal techniques . . . . .	26

---

2.9	Dynamic Mesh Handling . . . . .	31
2.10	Frequency analysis . . . . .	33
2.11	Summary . . . . .	34
<b>3</b>	<b>Method</b>	<b>35</b>
3.1	Software . . . . .	35
3.2	Meshing . . . . .	36
3.2.1	Mesh Quality . . . . .	36
3.3	Solver and numerical settings . . . . .	37
3.3.1	Solver . . . . .	37
3.3.2	Settings . . . . .	38
3.4	Damping layer case . . . . .	39
3.4.1	Mesh . . . . .	39
3.4.2	Boundary conditions and initializations . . . . .	40
3.4.3	Optimizing the damping parameter $f$ . . . . .	41
<b>4</b>	<b>Case Studies</b>	<b>45</b>
4.1	Oscillating cylinder - Quiescent water . . . . .	45
4.1.1	Refined mesh . . . . .	49
4.1.2	Frequency analysis . . . . .	51
4.1.3	Discussion . . . . .	51
4.2	Oscillating cylinder - Uniform flow . . . . .	53
4.2.1	Discussion . . . . .	56
4.3	Oscillating cylinder - Linear shear flow . . . . .	57
4.3.1	Discussion . . . . .	59
<b>5</b>	<b>Conclusion</b>	<b>61</b>
5.1	Concluding remarks . . . . .	61
5.2	Further work . . . . .	62
	<b>Bibliography</b>	<b>63</b>
	<b>Appendix</b>	<b>67</b>



# List of Tables

2.1	A detailed description of the terms in (2.85) . . . . .	29
3.1	List of skewness values and the corresponding cell quality . . . . .	37
3.2	Numerical schemes used in all simulations . . . . .	38
3.3	Quality parameter for the mesh . . . . .	39
4.1	Quality parameter for the original mesh . . . . .	45
4.2	Boundary conditions for quiescent water . . . . .	46
4.3	Boundary conditions for uniform flow . . . . .	54

# List of Figures

1.1	Wake patterns formed behind islands in the North Sea observed by NASA satellites[45]. . . . .	3
2.1	Group and phase velocity of a wave packet . . . . .	6
2.2	Comparison of Stokes 5th order and Airy wave [9]. A Stokes wave has deeper troughs and higher crests and does not oscillate with equal amplitude on each side of the mean water level . . . . .	10
2.3	Water wave regimes . . . . .	12
2.4	Examples of regular, non-regular, equilateral and skewed cells . . . . .	14
2.5	Structured orthogonal grid vs unstructured non-orthogonal grid . . . . .	14
2.6	A usual depiction of mesh notation, where $n$ is the face value at the north face of a volume, $N$ is the cell value at the cell at the north position and $n_e$ is the normal in the east direction . . . . .	15
2.7	An example of the Eulerian, Lagrangian and ALE mesh and particle motion	21
2.8	Volume fraction is captured as fraction from 1 to 0, which can be illustrated by color function from red to blue respectively. . . . .	22
2.9	Implicit representation of the curve $x^2 + y^2 - 1 = 0$ [38] . . . . .	24
2.10	Sphere generating a splash of water[18] . . . . .	25
2.11	Wave height at different timesteps for 1D wave for different sets of boundary conditions . . . . .	26
2.12	An overview of a 2D model with the original domain and the PML layer around[55]. . . . .	27
2.13	Ramping options in OpenFOAM plotted against the functions provided in Star+ CCM . . . . .	31
2.14	Mesh deformation problem[31] . . . . .	32
2.15	Frequency analysis of an Gaussian initial wave deformation over a uniform current . . . . .	34
3.1	Illustration of how ANSYS Mesh calculates aspect ratio . . . . .	36
3.2	Vectors used to compute orthogonal quality for an element . . . . .	37
3.4	Surface height for the entire domain plotted for the last 6 periods for $f = 12$	41
3.5	Surface height for the entire domain plotted for the last 6 periods for $f = 40$	42
3.6	Reflection coefficient for the averaged value over 6 wave periods . . . . .	43

---

4.1	Original mesh around the cylinder . . . . .	46
4.2	The domain setup for the moving cylinder cases . . . . .	46
4.3	Courant number and $\delta t$ plotted for the time interval: [30s 35s] . . . . .	47
4.4	The last 1000 iterations for the hydrostatic pressure, $p_{rgh}$ . . . . .	47
4.5	Surface height plot at $t = 20$ s . . . . .	48
4.6	Surface height plot at $t = 20$ s . . . . .	48
4.7	Volume fraction for two time intervals, where the variance from the initial starting value is given a function of time . . . . .	49
4.8	The surface height at the boundary divided by the amplitude of the initial disturbance and the amplitude of the smallest waves observed at 30s outside the damping zone . . . . .	50
4.9	A small irregularity near the cylinder . . . . .	50
4.10	A comparison of increased resolution amplitude-wise . . . . .	51
4.12	The inlet was split in order to control air and water separately. . . . .	54
4.13	The surface height at two time intervals with equal phase . . . . .	55
4.14	The peak to distance measured from the cylinder for each wavelength. . . . .	55
4.16	The problem description for the linear shear case. . . . .	57
4.17	The surface height at the end of simulation time at the same phase. . . . .	58
4.18	The peak to peak distance for the linear shear. . . . .	58
4.19	The linear dispersion relation and an assumptive correction for weakly non-linear . . . . .	59
4.20	Frequency analysis of the linear shear case . . . . .	60
A1	Vortices were formed in front of the cylinder and trapped as the wave velocity and background flow were in opposite direction. Sometimes these vortices would collide into the surface causing wave overturning and breaking. . . . .	67
A2	An example of vortex hitting the boundary causing wave breaking. . . . .	68
A3	Streamfunction at $t=23$ for linear shear current . . . . .	69
A4	The magnitude of the velocity vector $t=23$ for linear shear current . . . . .	70
A5	The hydrostatic pressure at $t=23$ for linear shear current . . . . .	71

# Nomenclature

$\alpha_s$	Correction factor for the dispersion relation for third order Stokes expansion
$\mathbf{u}$	The velocity vector consisting of $u, v$ and $w$ respectively
$\delta t$	Time step
$\delta_{ij}$	The Kronecker delta
$\gamma$	The surface tension
$\lambda$	Wavelength
$\omega$	The angular frequency
$\phi$	The velocity potential
$\sigma_{ij}$	The stress tensor
$\tau$	Resonance frequency
$\varsigma$	Excitation frequency
$\zeta$	Surface height, which is described as the distance from the mean water level
$a$	Amplitude of the wave
$c_g$	Group velocity or velocity of a wave packet
$C_o$	Courant number
$c_p$	The phase velocity of a wave
$Fr$	Froude number
$Fr$	The Froude number
$Fr_s$	The shear Froude number
$g$	The gravitational acceleration
$h$	The water depth

---

$K$	The curvature of the surface
$k_x, k_y$	The wave number in $x$ and $y$ directions respectively
$l$	Characteristic length
$p$	The scalar pressure
$R$	Radius of the cylinder
$R1, R2$	The principal radii of the curvature
$S$	Vorticity constant
$s$	Wave steepness

# Chapter 1

## Introduction

### 1.1 Background

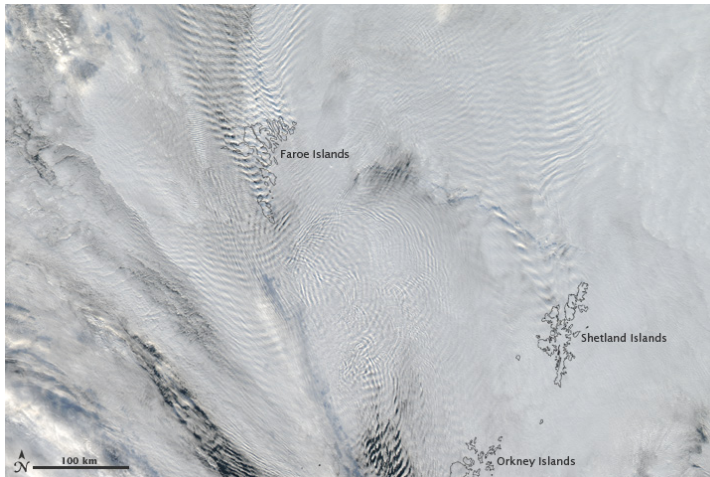
The work in this thesis is performed on behalf of EPT, NTNU as part of the Fluid Mechanics group. A broad range of theoretical work (see [11],[12],[13],[49]) has recently been published concerning shear currents and wave interactions. This work has shed a lot of light on how shear currents can even effect the Kelvin angle of the wake. Shear currents are found in coastal waters, where the current can be very different dependent on the depth. The shear current can be caused by wind, tides or bottom geometry and can be an important factor for say a ferry crossing a river. Even though a lot of theory has been published, are numerical studies on the subject at best limited. The theoretical work is focused on linear and weakly non-linear theory, while a numerical study has no such restrictions. This work is considered as a pursuit in knowledge of numerical studies of shear currents and wave interactions.

An extensive review on wave and current can be found in Peregrine [41]. Shear currents are of interest in oceanography and coastal engineering. Recently work done by Ellingsen and Li [35] showed that the wave resistance upstream of the ship increases due to the shear current, therefore indicating its importance in any ship calculation. Simple models for a ship movements are oscillating sources, where heave and/or sway can be effectively modelled[54]. Linear theory predicts infinite amplitudes in the bow wave for certain frequencies, and these frequencies are affected by shear current(see [21] and [51]). Numerical studies of cylinders fully submerged in a fluid are often used as benchmark studies of numerical solvers, as a lot theory and many experimental studies are available. Close to a free surface, the vortex shedding behind the cylinder is influenced by the free surface [44]. The vortex shedding causes a wave pattern in wake, where the frequency of the waves produced coincide with the frequency of the wave. From this, one can postulate that a cylinder with a given oscillation will produce waves much like a wave-maker, creating waves of correspondent frequencies as the excitation.

The numerical study was performed on the open-source software, OpenFOAM. OpenFOAM has large library of applications and solvers that are very useful for this project. Furthermore, modifications can with caution be made, making the software highly attrac-

---

tive for use beyond the more commonly used CFD practices. Recently, a new framework for wave models with tools for wave initialisation and damping was added to OpenFOAM with the release of version 5.0. This makes the OpenFOAM package fully competitive with the current market of commercial software in terms of wave modelling.



**Figure 1.1:** Wake patterns formed behind islands in the North Sea observed by NASA satellites[45]

## 1.2 Motivation and Objectives

The overall motivation of the thesis is to shine some light on shear currents and wave interactions with the aid of numerical simulation. To accomplish this, several intermediate objectives were set, which is also found in the master's contract:

1. Developing and implementing an appropriate boundary condition for surface waves, which can absorb a continuous spectrum of waves in 2D geometries.
2. Extended the 2D treatment to include a current of constant shear (linear-depth dependence).
3. Study 2D ship wave problems with linear shear current at the inlet and a surface obstacle (for example a cylinder).
4. Analyze the sub-surface structures beneath a nonlinear surface wave.
5. Generalize inflow conditions to allow more general shear flow and study differences from linear current case.

The first objective was an extension of the project done as a prelude to the thesis, where mono-chromatic waves were absorbed and a simplified model was used to linear wave theory. In section 2.8, the current state of the art of absorbing boundary conditions are shown

---

and reviewed in terms of their feasibility. A small parameter study is also performed in section 3.4, to investigate any discrepancy from other software, where Peric and Maksouds model is used for comparison [43]. The second, third and fifth objective, are all dependent on modelling a shear flow which is shown in section 4.3. The second objective was solved in as an intermediate step to objective three, but results are not specifically shown for this step. In order to study a ship wave problem, following objective three, through trial and error a choice was made to look at an oscillating cylinder. The oscillating cylinder proved valuable as both a model for ship as well as a wave-maker. The fifth objective was partly achieved, where a the inflow conditions for an arbitrary shear current were found, while only a linear shear was presented in section 4.3. The fourth objective was not given a special focus, but was treated indirectly in section 4.3 and section 4.2. The linear phenomena of infinite amplitudes due to resonance of a moving source was compared to a numerical solution, which includes non-linear effects. At the time master's contract was drawn up, there was possibility that experiments were to be made for comparison, however this was not performed in time for this master's due date.

### 1.3 Outline

The thesis starts with presenting the relevant theories in the next chapter. At first wave parameters and terminology is explained, then the concept of a free surface is defined. From the free surface boundary condition, linear wave theory derivations are showed before oscillating sources and wave regimes are clarified. That concludes the first part of the theory, where the second part focuses on numerical methods. Numerical methods are introduced with an emphasis on the finite volume method and a detailed explanation of the PIMPLE-algorithm. Then numerical representations of free surfaces are explained, before a review of current absorbing boundary conditions is presented. The theory portion is concluded with dynamic mesh handling specific to the OpenFOAM module. In chapter 3, the numerical setup and pre- and postprocessing are explained explicitly. Quality parameters are defined for the mesh, and numerous solver settings are discussed and explained. The chapter ends with a test of the damping layer boundary condition implemented in OpenFOAM, which is compared to similar setups. Then in the following chapter, three cases are run with an oscillating cylinder. The first case is an oscillating cylinder on quiescent water, which is used to validate the oscillating cylinder as a wave-maker by comparing the simulation data to the dispersion relation found in the theory section. The second case is run with a uniform inflow, modelling a moving source, where the excitation frequency is approximately at the resonance frequency. The same procedure is then repeated for linear shear profile at the inlet. The results and further work are then discussed in chapter 5.



# Chapter 2

## Theory

In this chapter, the general theory is outlined. At first, some general aspect of waves terminology is given, then free surface waves and linear wave theory is presented. The wave part is concluded with theory on oscillating sources and water wave regimes. Then the numerical methods used in the simulation are shown, including the finite volume methodology, discretization schemes, PIMPLE algorithm, free surface modeling and dynamic mesh handling.

### 2.1 Wave theory and terminology

Waves, in the classical definition given in wave theory, are described as a periodic perturbation from an original state. In the case of water waves, this perturbation arises in the interface between two different phases by some sort of displacement from an equilibrium state or the still water level.

Wave theory is applicable to several fields of studies such as acoustics, electromagnetics and elasticity. The terminology, however is generally the same and some of the important terms is characterized here. The general solution for 2D plane waves

$$f(x, y, t) = e^{-i(\omega t + k_x x + k_y y)}, \quad (2.1)$$

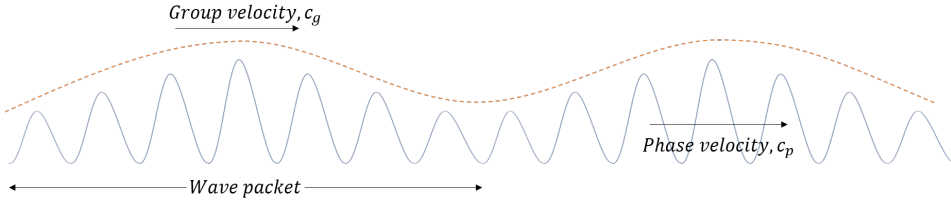
describes the behaviour of the waves in time and space. Here  $\omega$  is the angular frequency,  $k_x$  and  $k_y$  are the wave number in  $x$  and  $y$  directions respectively. An important distinction for waves is whether or not they are dispersed. Dispersion occurs when the propagation velocity of the plane waves is dependent on wavelength. This generally means that it now has a phase and group velocity as shown in Figure 2.1. Phase velocity,  $c_p$ , is defined as

$$c_p = \frac{\omega}{k}, \quad (2.2)$$

while group velocity, which is the velocity of the wave packet or envelope, is

$$c_g = \frac{\partial \omega}{\partial k}. \quad (2.3)$$

It is important to note that though generally phase and group velocities have the same direction as in Figure 2.1, this is not necessarily always the case. The dispersion relation describes the nature of the relationship between the wave number,  $k$  and the angular frequency,  $\omega$ .



**Figure 2.1:** Group and phase velocity of a wave packet

## 2.2 Free surface flows

A general definition of free surfaces is that "... *free surface flows occurs in a deformable solution region whereby the shape and size of the region is part of the solution.*" This introduces one of the reasons why free surface flows are complicated to deal with. The boundary itself is allowed to move, affecting the flow, which in turn affects the movement of the boundary.

This general definition excludes open channel flow and some wave problems, which is often included in an engineering point of view. A more physical definition can therefore be stated as: *Free surface flow occurs in a liquid, where one or more of the boundaries is not physically constrained, but can adjust itself to conform to the flow conditions.* Examples of these types of flows are jets, cavities, flow over weirs, stratified flow and of course problems that involves water and air such as waves propagating on the ocean[23].

### 2.2.1 Free surface boundary conditions

In order to properly model a free surface, boundary conditions are needed to express it mathematically. The free surface has two boundary conditions, the kinematic and the dynamic boundary condition [16]. The kinematic boundary condition states that a particle on the surface stays on the surface. That can be mathematically expressed as:

$$z - \zeta = f(x, y, t) \quad (2.4)$$

where  $f = 0$  on the surface,  $z$  is the coordinate in the gravitational direction and  $\zeta$  is the surface height function. The kinematic relation to express the surface height with respect to motion is

$$\frac{Df}{Dt} = (\partial_t + \mathbf{u} \cdot \nabla)(z - \zeta) = 0, \quad (2.5)$$

where  $\mathbf{u}$  is the velocity vector,  $\partial_t$  is the time derivative. For  $z = 0$  as a reference plane and from (2.5),  $\frac{\partial z}{\partial t} = u \frac{\partial z}{\partial x} = v \frac{\partial z}{\partial y} = 0$  and  $w \frac{\partial z}{\partial z} = w$  since  $z$  is constant. This gives the general kinematic boundary condition.

$$\frac{\partial \zeta}{\partial t} + \mathbf{u} \cdot \nabla \zeta = w(0, t) \quad (2.6)$$

The dynamic boundary condition states that the sum of forces through a surface must be in equilibrium. From the Young-Laplace[17] relation discontinuity due to the surface tension can be expressed. As viscous effects can be ignored the stress can be derived from the stress tensor as a simple pressure relation. The stress relation is described as

$$\sigma_{ij} = -p\delta_{ij}, \quad (2.7)$$

where  $\sigma_{ij}$  is the stress tensor and  $\delta_{ij}$  is the Kronecker delta, which is 0 for  $i \neq j$  and else equal to 1 and  $p$  is the pressure. Then the Young-Laplace relation and (2.7) gives

$$\frac{1}{R1} + \frac{1}{R2} = -\frac{\zeta_{xx}}{\sqrt{1 + \zeta_x^2}} - \frac{\zeta_{yy}}{\sqrt{1 + \zeta_y^2}} \approx -\zeta_{xx} - \zeta_{yy} \quad (2.8a)$$

$$p_{\text{above}} = p_{\text{below}} - \gamma K \approx p_{\text{above}} = p_{\text{below}} - \gamma(\zeta_{xx} + \zeta_{yy}) \quad (2.8b)$$

where  $K = \frac{1}{R1} + \frac{1}{R2}$  is the curvature of the surface,  $R1$  and  $R2$  are the principal radii of the curvature and  $\gamma$  is the surface tension. The pressure below is found through the Euler equation as formerly defined. The full dynamical boundary condition, where the viscous terms are neglected for simplicity, is then

$$\left. \frac{D\mathbf{u}}{Dt} \right|_{z=\zeta} = -\rho g \hat{\mathbf{z}} + \nabla(\gamma(\zeta_{xx} + \zeta_{yy})). \quad (2.9)$$

In this thesis, the effects of surface tension are negligible and can be simply removed from (2.9). By non-dimensionalization of (2.9),  $Fr$  or the Froude number, naturally appears. The Froude number is an expression for the ratio of the flow inertia compared to the external field (gravity) and is defined as

$$Fr = \frac{U}{\sqrt{gl}}, \quad (2.10)$$

where  $l$  is a characteristic length such as the length of a boat, diameter of cylinder or similar.

## 2.3 Surface waves

### 2.3.1 Linear surface waves

In order to simplify analytical analysis, surface waves are often restricted to being linear. Linear refers in this case to the fact that any non-linearity arising from the NS-equations or boundary conditions are neglected. Surface waves are considered as linear if the amplitude of the wave is negligible in comparison to the wave length. Therefore the assumption that

---

the waves are linear is in the order of  $s = \frac{\lambda}{a}$ , where  $\lambda$  is the wavelength,  $s$  is the steepness and  $a$  is the amplitude of the wave. Wavelength is the distance from crest to crest or trough to trough. This is an approximation, but it is an effective way of modelling many wave-phenomena, except of course the non-linear effects such as for instance breaking waves. Starting from the governing equations one can assume that the Reynolds number will be high as the viscosity of water is in the order of  $10^{-6}$ , which unless boundary layer present, and fairly long waves are assumed is an valid assumption. In boundary layers viscosity becomes very important for the attachment of the flow to the geometry, which means that the flow is no longer inviscid. Inviscid flow can be described by the Euler equations, and in the case that the flow is irrotational, then velocity potential can be applied. The relation between the velocity and the velocity potential is

$$\mathbf{u} = \nabla\phi, \quad (2.11)$$

where  $\phi$  is the velocity potential.  
This holds true if

$$\nabla \times \mathbf{u} = 0. \quad (2.12)$$

The dynamic and kinematic can then be simplified, if no external flow is present and  $z$  is the direction of gravity, to

$$\frac{\partial\zeta}{\partial t} + g\frac{\partial\phi}{\partial z} = 0 \quad (2.13a)$$

$$\frac{\partial\phi}{\partial t} + g\zeta = 0, \quad (2.13b)$$

where  $g$  is the gravitational acceleration. The continuity equation then becomes a Laplace-equation

$$\nabla^2\phi = 0. \quad (2.14)$$

A wave travelling in the x-direction can now be expressed in the way common to linear wave theory

$$\zeta = a \sin(kx - \omega t) \quad (2.15)$$

The relationship between the wave number and the wave frequency is called the dispersion relation and can be derived from solving Equation 2.14 and Equation 2.13. Landau [34] and Peregrine[41] are among those who treat this problem. The dispersion relation is in this case

$$\omega = \sqrt[2]{gk \tanh(kh)} \quad (2.16)$$

where  $h$  is the water depth. In the case of a uniform background flow in the x-direction, the solution is Doppler-shifted so that

$$\zeta = a \sin(kx - (\omega + U_0k)t). \quad (2.17)$$

---

If the background flow is a linear shear flow and the flow is considered to be 2D, the argument of using potential flow still holds according to Ellingsen and Brevik [13]. The surface waves are superimposed on a mean flow  $U$ , where the mean flow is

$$U(z) = U_0 + Sz, \quad (2.18)$$

and the vorticity,  $S$ , is

$$S = U_0/h. \quad (2.19)$$

One of the main differences is that the phase velocities becomes direction dependent. In other words, the phase velocity of the wave moving upstream will be different from the phase velocity of the wave moving downstream. Upstream and downstream here is defined according to the direction of the mean flow. Therefore a new expression is formed for the liquid surface and  $\zeta$  can be written as

$$\zeta(x, t) = \beta_u(k)e^{ik(x-U_0t+c_u t)} + \beta_d(k)e^{ik(x-U_0t+c_d t)}. \quad (2.20)$$

The expression has two undetermined amplitudes,  $\beta_u$  and  $\beta_d$ , which must be determined from initial conditions. The phase velocities are then

$$c_u(k) = \sqrt{\frac{g}{k} \tanh kh + \left(\frac{S}{2k} \tanh kh\right)^2} + \frac{S}{2k} \tanh kh \quad (2.21a)$$

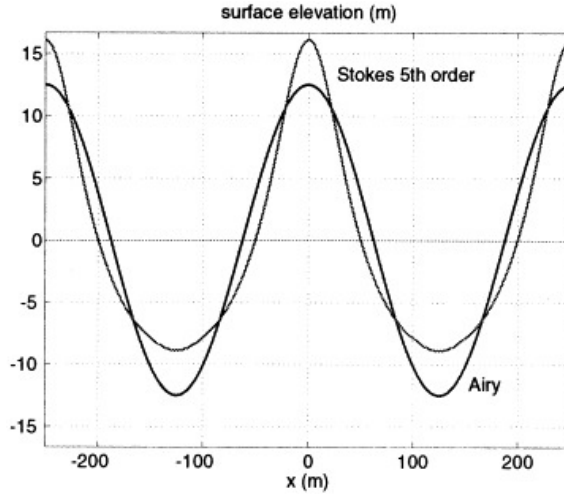
$$c_d(k) = \sqrt{\frac{g}{k} \tanh kh + \left(\frac{S}{2k} \tanh kh\right)^2} - \frac{S}{2k} \tanh kh, \quad (2.21b)$$

and the dispersion relation is

$$\omega_u(k) = \sqrt{gk \tanh kh + \left(\frac{S}{2} \tanh kh\right)^2} + \frac{S}{2} \tanh kh \quad (2.22a)$$

$$\omega_d(k) = \sqrt{gk \tanh kh + \left(\frac{S}{2} \tanh kh\right)^2} - \frac{S}{2} \tanh kh. \quad (2.22b)$$

For arbitrary shear currents, the math becomes much more involved and solutions are moved into the realm of approximations. By assuming a weak current in the order of  $\epsilon$ , Kirby and Chen [33], found a way to approximate the dispersion to the order of  $\epsilon$  for an arbitrary current of finite depth. Weak shear current are defined as  $U(z) \ll c$ , where  $c$  is the absolute phase velocity. Another method to solve the arbitrary shear profile was presented by Smeltzer and Ellingsen [49]. They used a piecewise linear approximation, where there is no need for a weak shear assumption. The approximation was shown to be accurate down to 1% for 4 to 5 layers.



**Figure 2.2:** Comparison of Stokes 5th order and Airy wave [9]. A Stokes wave has deeper troughs and higher crests and does not oscillate with equal amplitude on each side of the mean water level

### 2.3.2 Non-linear surface waves

Linear waves are assumed to be of a low steepness  $s = \lambda/a = ka$ , but for moderately steeper waves corrections can be added. These corrections do not affect the velocity potential formulation, but must be added to correct for the added steepness. The weakly-non-linear waves are often known as Stokes wave, and for detailed derivations, the reader is referred to Stokes [50]. Stokes based his approach on a perturbation theory, where  $\phi$  and  $\zeta$  are expanded in a series.

$$\phi = \sum_{i=0}^{\infty} \varepsilon^i \phi_i \quad \zeta = \sum_{i=0}^{\infty} \varepsilon^i \zeta_i \quad (2.23)$$

Here  $\varepsilon = s$  and is expanded to the  $i$ -th order. For a second order solution, the linear dispersion relation still describes the relation between angular velocity and wave number, however in the third-order expansion for infinite depth, a correction factor appears. The correction factor,  $\alpha_s$ , is related to the dispersion relation by

$$\omega = \omega_0 \alpha_s \quad \alpha_s = \left[ 1 + \frac{1}{2}(ka)^2 \right], \quad (2.24)$$

where  $\omega_0$  now is the linear dispersion relation.

## 2.4 Oscillating source

In the land of velocity potentials, flow singularities can be used to model floating and submerged bodies. Bodies such as these can represent ships or marine structures and is

a highly used method in marine hydrodynamics. The obvious advantage of using potential theory to describe this type flow is that the use of sources allow for panel methods, which have been very successful in aerodynamics and there already exists any number well tested approaches. The general problem of a time-dependent submerged source of a time-dependent source in uniform flow was outlined by Wehausen and Laitone[54]. It was predicted that there exist a certain frequency and velocity relationship leads to resonance for upstream waves. Grue and Palm showed that a source distribution due to a 2D oscillating body in a uniform flow[21], has waves of finite amplitude. Dagan and Miloh discovered that the amplitudes become bounded once higher order terms are considered[6]. For a body oscillating in a uniform current, several wavelengths are generated. For weak currents,  $\tau = U\omega_0/g < 1/4$ , four waves are generated. For stronger currents,  $\tau > 1/4$  two waves are generated. They are termed as  $k_1$ ,  $k_2$ ,  $k_3$  and  $k_4$  wave types, where  $k_1$  have a group velocity less then the strength of the uniform current,  $U$ , and is found downstream, while  $k_2$  have a group velocity larger than  $U$ , and is found upstream. The  $k_3$  wave has a positive phase velocity smaller than  $U$  and the  $k_4$  has a negative phase velocity and is located downstream. As  $\tau$  approaches  $1/4$ , the  $k_1$  wave and  $k_2$  wave merge into one wave with a group velocity equal to  $U$ . This creates a singularity at  $\tau = 1/4$ , which can shown through the Green function as described by Grue and Palm [21]. The resonance frequency,  $\tau$  is defined as

$$\tau = \frac{U\zeta}{g}, \quad (2.25)$$

where  $\zeta$  is the excitation frequency of the source. From Grue and Palm, an expression is shown for the far-field solution of the surface height:

$$\zeta_\infty = a_2 \sin(k_2x - \zeta t + \theta_2) \quad (2.26a)$$

$$\zeta_{-\infty} = a_1 \sin(k_1x - \zeta t + \theta_1) + a_3 \sin(k_3x - \zeta t + \theta_3) - a_4 \sin(k_4x - \zeta t + \theta_4). \quad (2.26b)$$

Here  $\theta$  is the angular shift by a sway motion, while any numeral subscript refer to the wave type associated with the same subscript. The amplitudes  $a_q$ , ( $q = 1, 2, 3, 4$ ), was found to be:

$$a_q = \begin{cases} \frac{2\pi(k_q R)^2 \epsilon}{(1 - 4\tau)^{1/2}} e^{-k_q d} & (\tau < 1/4, q = 1, 2) \\ 0 & (\tau < 1/4, q = 1, 2) , \\ \frac{2\pi(k_q R)^2 \epsilon}{(1 + 4\tau)^{1/2}} e^{-k_q d} & (q = 3, 4) \end{cases} \quad (2.27)$$

where  $R$  is the radius of the cylinder,  $d$  is the depth and  $\epsilon$  is the amplitude of the heaving motion. This result has two special cases: Firstly, as mentioned earlier, the amplitude of the  $k_1$  and  $k_2$  wave merge and create a infinite amplitude for  $\tau = 1/4$ , but also secondly that the amplitude for the downstream wave is zero when  $\tau > 1/4$ .

Tyvand and Lepperød made use of the line source and included a linear shear, where they showed that shear affects the resonance frequency [51]. Even though their method

was physically inconsistent as showed by Ellingsen and Tyvand [14], the dispersion problem is treated correctly. The frequency-velocity relationship at resonance was showed to be dependent of shear current and expressed as

$$\tau = \frac{(1 + Fr_s)^2}{4} \quad (2.28)$$

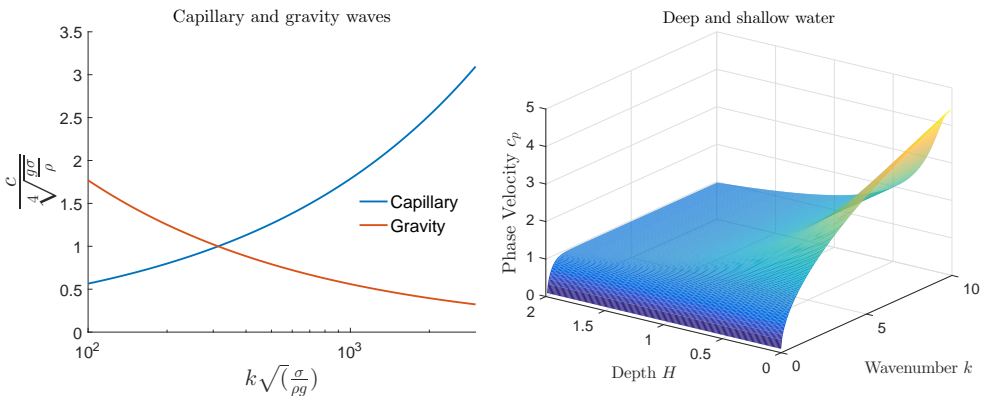
for a 2D source travelling in shear assisted motion. The shear Froude number,  $Fr_s$ , is defined as

$$Fr_s = \frac{VS}{g}, \quad (2.29)$$

where  $S$  is the uniform vorticity. It comes from the choice of sub-surface shear current as  $U(z) = Sz$ .

## 2.5 Water wave regimes

Water waves can be categorized by different regimes, which generally indicates what type of flow to expect and can be used to simplify analysis. Water can first of all be divided into capillary and gravity waves. Capillary waves are often seen as ripples and are caused by the surface tension acting between interfaces of different densities, such as water and air. In Figure 2.3a the phase velocity of capillary and gravity waves are compared. The phase velocity gives information on which type of waves are dominant for different wave numbers. For a small wavelength, the capillary waves propagate at a much larger velocity then gravity waves and therefore gravity waves are the dominant wave type. While for smaller  $\lambda$  the gravity waves propagate at a larger velocity then capillary waves and opposite effect occurs.



(a) Phase velocity of capillary and gravity waves (b) Phase velocity of shallow and deep water

**Figure 2.3:** Water wave regimes

The next regime to characterise water waves is divided between shallow, intermediate and deep water waves. The phase velocity for shallow water is only affected by depth,



while for deep water the wave number is the important parameter. Intermediate depth is anything in between, where both parameters are important. So for gravity waves the phase velocity is

$$c_p = \sqrt{gk \tanh kh} \quad (2.30)$$

and is plotted in Figure 2.3b. The phase velocity,  $c_p$ , for water waves, can be found by combining (2.16) and (??). For large depths,  $kh \gg 1$ , the phase velocity becomes independent of  $h$

$$c_p = \sqrt{gk}. \quad (2.31)$$

For smaller depths,  $kh \ll 1$ , the phase velocity becomes independent of the wave number,  $k$  and the phase velocity is in that case

$$c_p = \sqrt{gh}. \quad (2.32)$$

## 2.6 Numerical methods

In the thesis, OpenFOAM has been chosen to be used for simulation. Within the OpenFOAM framework, there is support both for finite volume and finite element methods, which are the most prominent solution methods for fluid simulation.

### 2.6.1 Finite Volume Method

The finite volume method is based on the integral form of conservation equations. Starting from the momentum equation in its compact form with tensor notation

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_i \partial x_j} + f_i, \quad (2.33)$$

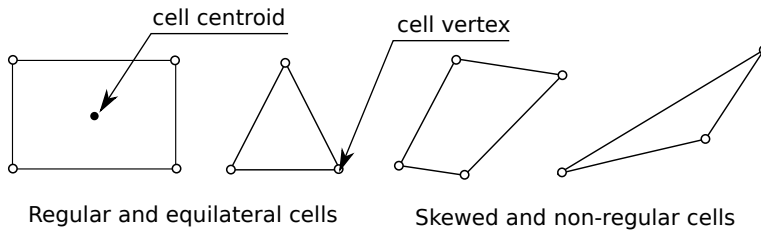
where  $u_i$  is the velocity for directions  $i = 1, 2, 3$ ,  $P$  is the pressure divided by the density,  $\nu$  is the kinematic viscosity,  $x_i$  is the spatial coordinates and  $f_i$  is the body force term, which is also divided by the density. Then integrating over a control volume,  $V$ , gives

$$\iiint_V \left[ \frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} \right] dV = \iiint_V \left[ -\frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_i \partial x_j} + f_i \right] dV, \quad (2.34)$$

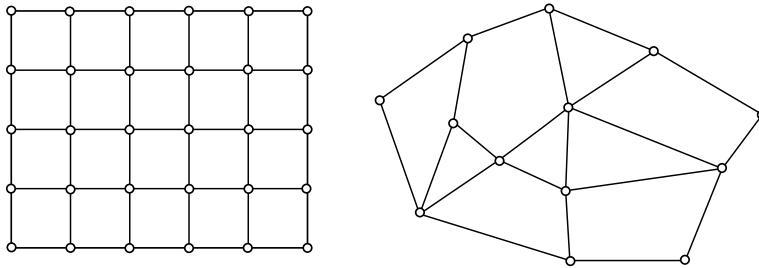
and applying the divergence theorem to the advection, pressure gradient and diffusion terms allows it to be rewritten as

$$\frac{\partial u_i}{\partial t} V + \iint_S u_i u_j n_j dS = - \iint_S P n_i dS + \iint_S \nu \frac{\partial u_i}{\partial x_j} n_j dS + f_i V. \quad (2.35)$$

where  $S$  is the surface of the control volume and  $n_j$  is the normal of the surface. The control volume,  $V$ , can be any type of volume and is usually applied to a small computational



**Figure 2.4:** Examples of regular, non-regular, equilateral and skewed cells



**Figure 2.5:** Structured orthogonal grid vs unstructured non-orthogonal grid

ensuring local conservation and thereby ensuring global conservation for a large volume divided into cells. A computational cell can in theory be any shape, but in 2D quadrilateral or triangular cells are the most common.

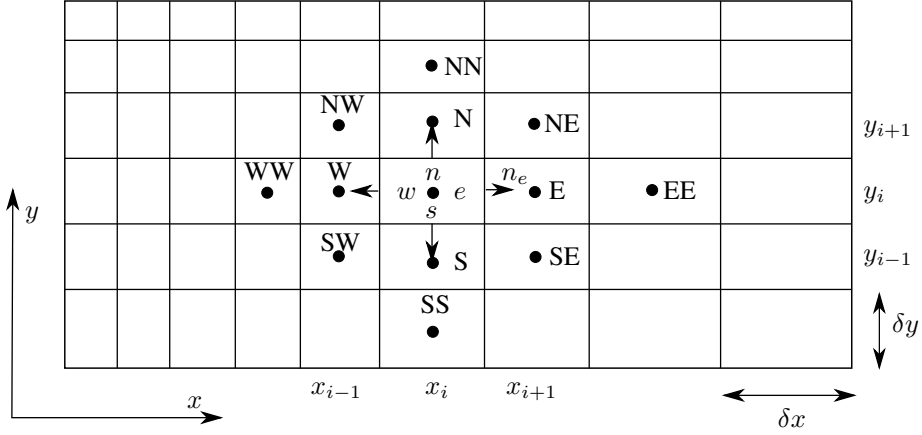
A grid of cells can be structured, unstructured, orthogonal, non-orthogonal and any combination of these depending on the geometry and problem. Structured meshes are defined by every interior vertex in the domain has the same number neighbouring cell vertexes [40]. Block structured meshes are divided into blocks where each block has its own block structure. Orthogonal meshes have grid lines perpendicular to the intersection, whereas non-orthogonal meshes have no such restriction. Another distinction is done by considering body-fitted or curvilinear grids which are based on curvilinear coordinates instead regular Cartesian coordinates. In general, structured orthogonal meshes are easier to deal with numerically, but unstructured and non-orthogonal meshes are much easier to deal with when complicated geometries are present.

## 2.6.2 Spatial and Temporal discretization

The finite volume version of the momentum equation as shown (2.35) is solved as a set of linear equations, where each term is discretized in space and time. Cell-face values used in these linear equations are approximated in terms of the nodal values. For instance a surface integral can be approximated by the midpoint rule for an arbitrary term [42],  $f$ , as

$$F_e = \int_{S_e} = \bar{f}_e S_e \approx f_e S_e \quad (2.36)$$

This approximation is valid if the value at 'e' is known, but this is generally not the case.



**Figure 2.6:** A usual depiction of mesh notation, where  $n$  is the face value at the north face of a volume,  $N$  is the cell value at the cell at the north position and  $n_e$  is the normal in the east direction

Therefore interpolation is needed in order to find this value. The two main ways of finding the interpolated value are upwind differencing scheme (UPS) and linear differencing scheme (LDS).

### Upwind Interpolation

Upwind differencing scheme is a backward or forward approximation for the first derivative in the upstream direction. A quantity  $\varphi_e$  at 'e' is approximated as

$$\varphi_e = \begin{cases} \varphi_P & \text{if } (\mathbf{v} \cdot \mathbf{n})_e > 0 \\ \varphi_E & \text{if } (\mathbf{v} \cdot \mathbf{n})_e < 0 \end{cases} \quad (2.37)$$

A first order Taylor expansion around P:

$$\varphi_e = \varphi_P + (x_e - x_P) \left( \frac{\partial \varphi}{\partial x} \right)_P \quad (2.38)$$

$$\left( \frac{\partial \varphi}{\partial x} \right)_P = \frac{\varphi_P - \varphi_e}{x_P - x_e} \quad (2.39)$$

The UDS is a first order scheme, which is the only scheme that satisfies the boundedness criterion and can never yield oscillatory solutions. However, the scheme is highly diffusive and accurate results can only be acquired for very fine grids.

### Linear Interpolation

A more accurate scheme is the central differencing scheme (CDS), which interpolates the value at CV-face center by linear interpolation. Again at 'e'

---


$$\varphi_e = \varphi_E \Lambda_e + \varphi_P (1 - \Lambda_e), \quad (2.40)$$

where the linear interpolation factor  $\Lambda_e$  is defined as

$$\Lambda_e = \frac{x_e - x_P}{x_E - x_P} \quad (2.41)$$

and a second order Taylor expansion of  $\varphi_E$  about  $x_P$  is

$$\varphi_e = \varphi_E \Lambda_e + \varphi_P (1 - \Lambda_e) - \frac{(x_e - x_P)(x_E - x_e)}{2} \left( \frac{\partial^2 \varphi}{\partial x^2} \right)_P. \quad (2.42)$$

Since CDS are second order, they are accurate to the square of the grid spacing, which applies for uniform and non-uniform grids [42]. Despite gain in accuracy, the scheme reduces stability and can produce oscillations like any higher order method. Many other differencing schemes exist, but they are usually a combination of these schemes or a higher order version of these. In OpenFOAM many of these schemes are implemented, among others the linear upwind scheme (LUDS). LUDS approximate  $\varphi_e$  linear extrapolation from two upstream nodes. From OpenFOAM's User Guide[20], LUDS is a "second order, upwind-biased, unbounded scheme that requires discretization of the velocity to be specified." However, boundedness can still be achieved. Bounded variant of the schemes relate to the treatment of the material time derivative which is equals the left hand side of the momentum equation if the derivative is the velocity. Therefore from

$$\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) - (\nabla \cdot \mathbf{u})\mathbf{u} \quad (2.43)$$

boundedness can be accomplished if the last term on the right hand side of (2.43) is kept. Normally,  $(\nabla \cdot \mathbf{u})$  is zero when continuity is fulfilled, however numerically this never happens as residuals are always present, therefore this can be included and thereby ensuring boundedness. These types of schemes are normally used for steady state calculation as it helps with convergence.

LUDS is second-order space scheme [53], which switches between a linear or an upwind biased formulation depending on the gradient. Large gradient usually causes wiggles or oscillations in linear schemes and this can be negated by introducing a weighted upwind scheme. In OpenFoam, there are two such schemes; linearUpwind and linearUpwindV. The only difference between them is that linearUpwindV calculates a single value for the weighted scheme and applies that value in all directions, while linearUpwind finds value in all directions. The single value is calculated in the direction of the largest gradient.

Limits can also be applied make sure a scheme does not overshoot or wiggles. The Van Leer scheme [52] is a monotonic conservation scheme that is more diffusive than a linear scheme, but that can handle jumps and shocks fairly well. This scheme is normally used for conservation equation for energy or phase fraction, which has a jump across the surface as the phase fraction goes to from 1 to zero. Wiggles would create artificial values of larger than 1 and less than zero, which can cause serious problems.

---

## Time schemes

The same type of differencing can be applied to the time derivative. Forward and backward differencing are known as backward or forward Euler and can be written as

$$\left(\frac{\partial\varphi}{\partial t}\right)^{n+1} = \frac{\varphi^{n+1} - \varphi^n}{\delta t} \quad (2.44)$$

$$\frac{\partial\varphi_{n+1}}{\partial t} = \frac{\varphi^n - \varphi^{n-1}}{\delta t}, \quad (2.45)$$

for a time step  $\delta t$ , where the backwards Euler is implicit, while the forward is explicit. In terms of stability, backward Euler is preferred as the boundedness ensures stability. Forward Euler has limited use as the stability requirements causes time-steps to be very small as well accuracy is only first-order.

Crank-Nicolson is a second order time differencing scheme [5], where the time derivative is

$$\left(\frac{\partial\varphi}{\partial t}\right)^{n+1} = \frac{\varphi^{n+2} - \varphi^n}{2\delta t}. \quad (2.46)$$

In OpenFoam, it is implemented as a hybrid scheme with the backward Euler for stability. The blending must be specified, where 0 is Euler and 1 is Crank-Nicolson, normally a value of 0.9 ensures accuracy and robustness according to the OpenFoam documentation [20].

An important parameter for scaling the time step in terms of stability is the Courant number. The Courant number is defined as

$$C_o = u \frac{\delta t}{\delta x}. \quad (2.47)$$

Time-dependent stability of schemes are usually measured in terms of the Courant number, where for explicit schemes  $C_o$  must be lower than 1 or 1/2 depending on the scheme. Implicit schemes allow larger time-stepping,  $C_o > 1$ , however larger time-steps can decrease the accuracy of the flow resolvability. Therefore care must be taken keep time-stepping at sufficient level so that all important flow properties are resolved sufficiently.

## Linear equations

All of the schemes involved with finite volume method must be solved as a system of linear equations of the form

$$A\mathbf{x} = \mathbf{b} \quad (2.48)$$

where  $A$  is the coefficient matrix,  $\mathbf{x}$  is the vector of unknowns and  $\mathbf{b}$  is the source vector.  $A$  and  $\mathbf{b}$  are filled by the discretization schemes. The system of equations are then on the

---

form

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}. \quad (2.49)$$

In order to solve (2.48), the coefficient matrix must be inverted and the solution vector is then

$$\mathbf{x} = A^{-1}\mathbf{b}. \quad (2.50)$$

In OpenFoam, three groups of iterative linear solvers are available for matrices; smooth solvers, conjugate gradient solvers and multigrid solvers.

Smooth solver are used to decrease high frequency errors, which can arise from the solving the linear equations. Smooth solvers are usually based on Gauss-Seidel methods, but can also be diagonal incomplete-Cholesky or incomplete-LU. Details on these linear methods are best explained in resources among which Youssef Saad[47] is recommended. The difference between them are usually related to solution time and types of matrices. For symmetric matrices symmetric solvers are usually much faster then asymmetric solvers, while symmetric solvers cannot be used for asymmetric matrices.

Multigrid methods are somewhat self explanatory in that they utilize a series of grids, where the first grid is coarse and then the resolution is increased until the final fine mesh is reached. Mapping from grid levels can cause oscillatory behaviour and therefore smoothing should be applied.

Conjugate gradient solvers are not used in the simulation of the text and are therefore beyond the scope of the text.

### 2.6.3 PIMPLE - Algorithm

Solving the Navier-Stokes requires numerical strategies for coupling the pressure and momentum quantities. The coupling arises by the fact that the Navier-Stokes has four unknowns and three equations to be solved and including the mass conservation gives a fourth equation. Applying the divergence to the momentum and using the mass conservation to get rid of excessive terms one can end up with a Poisson equation for the pressure  $p$ [42].

$$\frac{\partial}{\partial x_i} \left( \frac{\partial p}{\partial x_i} \right) = -\rho \frac{\partial}{\partial x_i} \left[ \frac{\partial u_i u_j}{\partial x_i} \right] \quad (2.51)$$

There are three main algorithms for solving the coupling of momentum and pressure in OpenFoam:

- SIMPLE - Semi-Implicit-Method-Of-Pressure-Linked-Equations
- PISO - Pressure-Implicit-Split-Operator
- PIMPLE - Merged SIMPLE and PISO

---

For an implicit method, the Poisson equation for the pressure is

$$\frac{\partial}{\partial x_i} \left( \frac{\partial p^{n+1}}{\partial x_i} \right) = -\rho \frac{\partial}{\partial x_i} \left[ \frac{\partial (u_i u_j)^{n+1}}{\partial x_i} \right], \quad (2.52)$$

and the momentum equation rewritten as

$$A_P^{u_i} u_{i,P}^{n+1} + \sum_l A_l^{u_i} u_{i,l}^{n+1} = Q_{u_i}^{n+1} - \left( \frac{\partial p^{n+1}}{\partial x_i} \right)_P. \quad (2.53)$$

$P$  is as before the index of the arbitrary velocity node and index  $l$  denotes the neighbour points that appear in the discretized momentum equations. The source term  $Q$  contains all of the terms that may be explicitly computed in terms of  $u_i^n$  as well as any body force or linearized terms that may depend on  $u_i^{n+1}$ . Non-linearity makes only iterative approaches possible. The iterations in a single time-step, where the coefficient and source matrices are updated, are called outer iterations, to separate from inner iterations done to solve the linear equations. For every outer iteration, the equations solved are:

$$A_P^{u_i} u_{i,P}^{m*} + \sum_l A_l^{u_i} u_{i,l}^{m*} = Q_{u_i}^{m-1} - \left( \frac{\partial p^{m-1}}{\partial x_i} \right)_P, \quad (2.54)$$

where the time step index  $n + 1$  is skipped for the outer iteration counter  $m$ . At the beginning of every outer iteration the right hand side is evaluated at the previous outer iteration. To enforce the continuity equation the velocities need to be corrected, because the velocities used are estimated. The velocity at  $P$  is expressed as:

$$u_{i,P}^{m*} = \frac{Q_{u_i}^{m-1} - \sum_l A_l^{u_i} u_{i,l}^{m*}}{A_P^{u_i}} - \frac{1}{A_P^{u_i}} \left( \frac{\partial p^{m-1}}{\partial x_i} \right)_P. \quad (2.55)$$

The SIMPLE algorithm does not use the actual pressure, but a pressure-correction. The velocities computed from the linearized momentum equations and the pressure  $p^{m-1}$  are taken as a starting value and a correction is then added:

$$u_i^m = u_i^{m*} + u' \text{ and } p^m = p^{m-1} + p' \quad (2.56)$$

These are substituted into (2.54) and a relation between the corrections is obtained:

$$u'_{i,P} = \tilde{u}'_{i,P} - \frac{1}{A_P^{u_i}} \left( \frac{\partial p'}{\partial x_i} \right)_P \quad (2.57)$$

where  $\tilde{u}'_i$  is defined by (2.55):

$$\tilde{u}'_{i,P} = - \frac{\sum_l A_l^{u_i} u'_{i,l}}{A_P^{u_i}}. \quad (2.58)$$

Then using the discretized continuity equation

$$\frac{\partial(\rho u_i^m)}{\partial x_i} = 0, \quad (2.59)$$

---

and (2.57) to produce the pressure-correction equation:

$$\frac{\partial}{\partial x_i} \left[ \frac{\rho}{A_P^{u_i}} \left( \frac{\partial p'}{\partial x_i} \right) \right]_P = \left[ \frac{\partial(\rho)u_i^{m*}}{\partial x_i} \right]_P + \left[ \frac{\partial(\rho)\tilde{u}'_i}{\partial x_i} \right]_P \quad (2.60)$$

The velocity corrections  $\tilde{u}'_i$  are unknown and is neglected. According to [42], this is why the method has convergence problems. The SIMPLE algorithm is normally used for steady state problems only. In the PISO algorithm, the first correction step is the SIMPLE algorithm and the second correction  $u''$  is:

$$u''_{i,P} = \tilde{u}'_{i,P} - \frac{1}{A_P^{u_i}} \left( \frac{\partial p''}{\partial x_i} \right)_P. \quad (2.61)$$

Then the second pressure-correction equation is:

$$\frac{\partial}{\partial x_i} \left[ \frac{\rho}{A_P^{u_i}} \left( \frac{\partial p''}{\partial x_i} \right) \right]_P = \left[ \frac{\partial(\rho\tilde{u}'_i)}{\partial x_i} \right]_P \quad (2.62)$$

The PIMPLE algorithm lets these outer corrections and pressure corrections be set to an arbitrary number. It also has the advantage of larger time steps ( $C_o$ ) than the PISO algorithm.

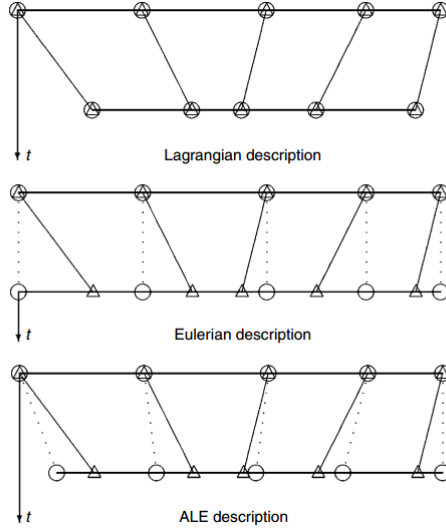
## 2.7 Numerical Representation of Free Surfaces

This section reviews current methods for representing free surfaces. Depending on the nature of the problem there are different approaches. In general methods are divided into two main categories, the Eulerian and the Lagrangian formulation. The Eulerian approach is well rooted in fluid mechanics and keeps the node of the mesh constant, while letting the fluid move within the mesh. The Lagrangian description can be thought of as a structural mechanics approach, where the mesh nodes follow the material and thereby changing their initial form. The Lagrangian algorithm is considerably better at resolving interfaces with high degree of accuracy. However it struggles with large deformations as this can lead to distorted meshes, which in turn can have convergence and stability issues as well as effect the total accuracy. This can in some cases be handled by remeshing, but for large internal movements as those experienced by fluids this causes a whole range problems and therefore is ill suited for this type problem. The Eulerian algorithms, on the other hand, has no problem in coping with large deformations, but they struggle with capturing an accurate representation of the surface, which often leads to a smeared out solution or has a high computational cost due to high grid refinement. Generally most methods for handling free surfaces are versions or combinations of these approach and the most important are given in this section.

### 2.7.1 Arbitrary Lagrangian Eulerian - ALE

The Arbitrary Lagrangian Eulerian algorithm (ALE) is a combination of the two previously mentioned methods in a way that uses the strength of each method. It can be set to





**Figure 2.7:** An example of the Eulerian, Lagrangian and ALE mesh and particle motion

Lagrangian at surfaces and be locked at other positions where large movements in the material occur. One of the major drawbacks of the method is that it cannot handle topological changes, for example breaking of water or the burst of a droplet. The differences between the three methods are illustrated in Figure 2.7.

In mathematical terms, there exist two domains in continuum mechanics describing the motion of particles. The material domain,  $R_X \subset \mathbb{R}^{n_{sd}}$ , with  $n_{sd}$  spatial dimensions and  $R_x$  is the spatial domain, where  $\mathbf{x}$  is the spatial dimension. This follows a specific notation, which will be used in this section [10]. For the ALE description there also exist a third domain, the reference domain  $R_{\mathcal{X}}$ . The reference domain can be considered as the distance relative to a stationary observer, and therefore does not change

If we first look at the conservation equations (energy is omitted for simplicity) in its normal Eulerian form

$$\frac{d\rho}{dt} = \left. \frac{\partial \rho}{\partial t} \right|_{\mathbf{x}} + \mathbf{u} \cdot \nabla \rho = -\rho \nabla \cdot \mathbf{u} \quad (2.63a)$$

$$\rho \frac{d\mathbf{u}}{dt} = \rho \left( \left. \frac{\partial \mathbf{u}}{\partial t} \right|_{\mathbf{x}} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \quad (2.63b)$$

where  $\rho$  is the density,  $\mathbf{v}$  is the material velocity vector,  $\boldsymbol{\sigma}$  is the stress tensor and  $\mathbf{b}$  is the specific body force vector.

To obtain the ALE form of the conservation equation the material velocity,  $\mathbf{u}$ , in the convective terms needs to be replaced with the convective velocity  $\mathbf{c} = \mathbf{u} - \hat{\mathbf{u}}$ .  $\hat{\mathbf{u}}$  is the mesh velocity defined as

$$\hat{\mathbf{u}}(\boldsymbol{\chi}, t) = \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\mathbf{x}}. \quad (2.64)$$

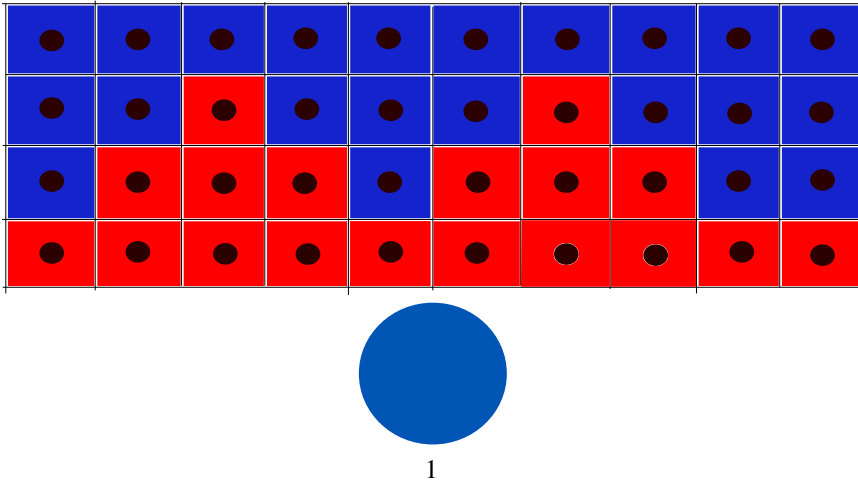
Then the conservation equations can be expressed in the ALE form as:

$$\frac{d\rho}{dt} = \left. \frac{\partial \rho}{\partial t} \right|_{\mathbf{x}} + \mathbf{c} \cdot \nabla \rho = -\rho \nabla \cdot \mathbf{u} \quad (2.65a)$$

$$\rho \frac{d\mathbf{u}}{dt} = \rho \left( \left. \frac{\partial \mathbf{u}}{\partial t} \right|_{\mathbf{x}} + (\mathbf{c} \cdot \nabla) \mathbf{u} \right) = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \quad (2.65b)$$

## 2.7.2 Volume of Fluid - VOF

Volume of Fluid (VOF) method as developed by Hirt and Nicholas is an Eulerian approach [26]. VOF can be considered as an extension of the MAC method in some sense as it uses the somewhat the same methodology in marking cells. However, Hirt and Nicholas describes fluids as a volume fraction instead merely denoting them as 1 or zero i.e containing fluid or empty. This allows for a more accurate expression for the amount of each fluid in each cell, as this has been previously assumed either one or zero leading to a relatively high number of cells. Another important aspect is that it is more general method that allows for multiphase flows as it can handle different fluids in the same cell.



**Figure 2.8:** Volume fraction is captured as fraction from 1 to 0, which can be illustrated by color function from red to blue respectively.

The method belongs to the class of Eulerian advective schemes, a numerical recipe to track the shape and position of the surface. The Volume of Fluid method is based on the idea of a fraction function  $\alpha$  on the form:

$$\frac{\partial \alpha}{\partial t} + \mathbf{u} \cdot \nabla \alpha = 0 \quad (2.66)$$

---

The discontinuous phase boundary is replaced with a soft boundary governed by the phase fraction  $\alpha$ . Thus the interface is tracked indirectly through the phase fraction  $\alpha$  and not explicitly. A sufficient resolution in the boundary zones is a requirement for accurate boundary tracking. This can be done by increasing the mesh resolution which gives sharper boundary.

In an air-water system the phase fraction parameter  $\alpha$  can be defined as

$$\alpha = \begin{cases} 1, & \text{water} \\ 0, & \text{air} \\ 0 < \alpha < 1, & \text{at the interface} \end{cases} \quad (2.67)$$

The density and viscosity of a fluid in a cell is found by summing over the product of the phase fraction and phase property in a given cell

$$\rho = \sum \alpha_i \rho_i \quad (2.68)$$

$$\mu = \sum \alpha_i \mu_i \quad (2.69)$$

In order to retain boundedness a limit must be set on the value of volume phase fraction in the cell. The limit is set by

$$\alpha_1 + \alpha_2 = 1. \quad (2.70)$$

This limits the total phase fraction in any cell to be max 1. The way to negate this is by using flux limiters to ensure boundedness. In OpenFOAM, this is done by using a bounded scheme such as the van Leer scheme.

### 2.7.3 Level set methods

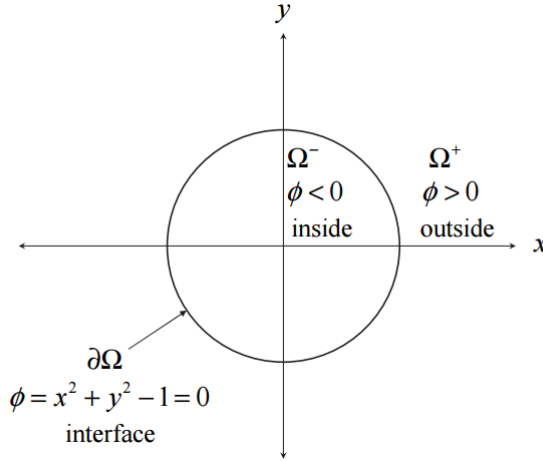
Free surfaces can also be represented by level set methods. Level set was developed in the 1980s in a series of papers by Osher and Sethian [39]. Level set methods are based on dynamic implicit representations of surfaces. An explicit representation is done by writing the explicit points of a curve, whereas the implicit representation is made up of a series of iso-contours. Each iso-contour has a set of points that crosses the interface. For example the zero contour of  $\phi(x) = x^2 - 1$  is the set of points that equals  $\phi(x) = 0$ . This can just as easily be done for a curve instead of points. For  $\phi(x, y) = x^2 + y^2 - 1$  a zero contour is any points on the curve, while contours inside the zero contour has values  $\phi < 0$  and outside they are  $\phi > 0$ .

This provide an excellent tool for judging whether a point lies on the inside, outside or on the interface simply by checking on values for  $\phi$ .

If the Eulerian approach is kept, the  $\phi$  function can be used both to represent and evolve the interface. The evolution is represented by the simple advection equation

$$\phi_t + \mathbf{v} \cdot \nabla \phi = 0. \quad (2.71)$$

The advection equation is a hyperbolic equation and therefore share any number of similarities to existing equations in fluid mechanics. Numerical approximations to the



**Figure 2.9:** Implicit representation of the curve  $x^2 + y^2 - 1 = 0$  [38]

equations are therefore similar methods such as a simple first-order upwind scheme or more complex scheme such as Hamilton-Jacobi ENO, Hamilton-Jacobi WENO or Runge-Kutta TVD [38].

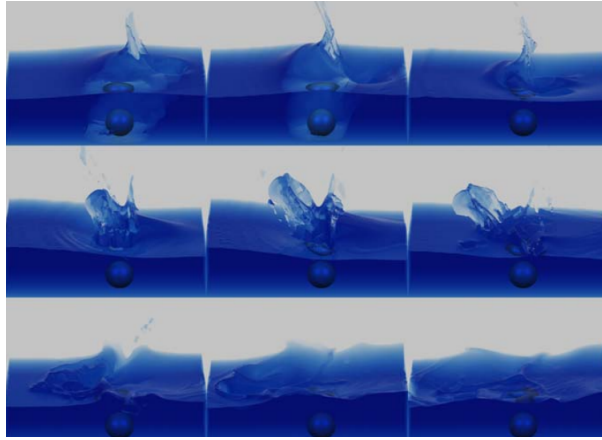
Despite being a powerful tool for describing free surfaces and multiphase flow, especially effective with topological changes, level set methods suffer from leakage of mass in underresolved regions of the flow. Foster and Fedkiw [18] suggest an improvement hybridizing the particle and level set method and use the local interface curvature as a diagnostic. This technique reduces much of the mass lost and has been used as realistic animation tool for water, a result can be shown in Figure 2.10.

## 2.8 Absorbing Boundary Conditions

Special care is needed near the outlet regions as waves hitting normal boundary conditions gets reflected and can then affect the solution area. This can be illustrated by looking at the scalar 1D wave equation

$$\frac{\partial^2 \varphi}{\partial t^2} - c^2 \frac{\partial^2 \varphi}{\partial x^2}, \quad (2.72)$$

where  $\varphi$  is the advected quantity and  $c$  is the velocity of the advection or phase velocity of the wave, in this case constant. The wave equation can be solved by a finite difference approach and the solution is shown in Figure 2.11. Here a initial wave propagates towards boundaries in the x-direction. Two approaches are compared; a zero gradient boundary and a transparent boundary condition. For a zero gradient boundary condition the entire wave is reflected and returns with same amplitude and propagation speed. For a radiating boundary condition, the wave height is predicted at the boundary in such a way that the wave appears to pass perfectly through the boundary. This shows the importance of being



**Figure 2.10:** Sphere generating a splash of water[18]

able to effectively treat waves hitting boundaries in order to minimize their effect on the solution.

### 2.8.1 Transparent Boundaries

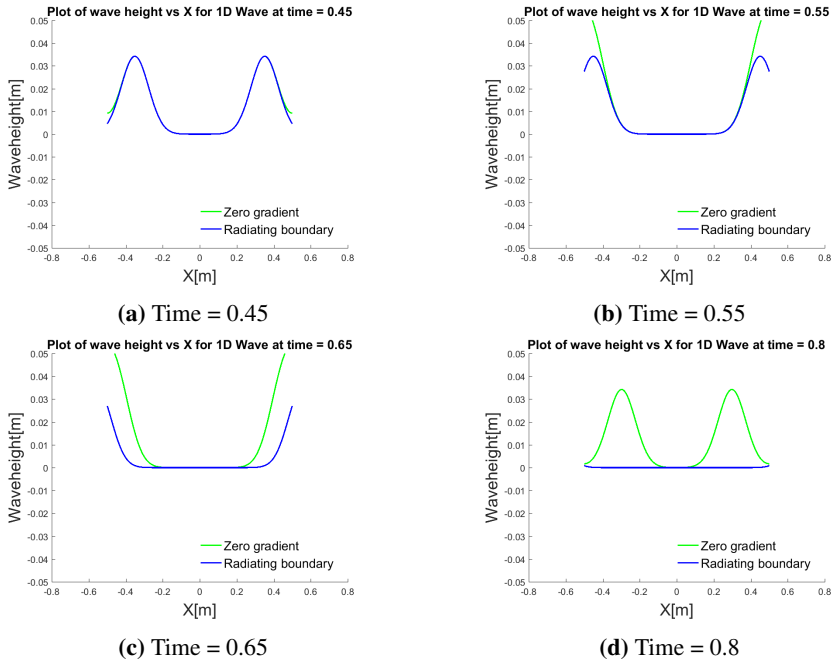
For the scalar wave equation, there exist several version of transparent boundary conditions. Enquist and Majda[15] to make a formal formulation of a transparent boundary condition for the 2D scalar wave equation with  $c = 1$

$$\left( \frac{\partial}{\partial x} + i\sqrt{k_y^2 - \omega^2} \right) \varphi_{x=0} = 0, \quad (2.73)$$

where  $k_y$  is the  $y$  component of the wavenumber. The global formulation in half space (2.73) can be approximated (given in first order for simplicity) by Padè or Taylor approximation and moved back into real space, can then be written as

$$\left( \frac{\partial}{\partial x} + \frac{\partial}{\partial t} \right) \varphi_{x=0}. \quad (2.74)$$

This is the same formulation as the Sommerfeld radiation condition. Orlanski[37] had already provided a finite difference approximation by extrapolation for the same radiation condition, including a way to estimate the phase velocity. Higdon[25] extended the formulation to dispersive waves and the Klein-Gordon equation, including the phase velocity and group velocity in the formulation. Higdon[24] also accounted for waves at an angle of incidence to the boundary. This was extended by Romate[46] to include free surface waves with a velocity potential formulation. The BC was formulated in first and second



**Figure 2.11:** Wave height at different timesteps for 1D wave for different sets of boundary conditions

order, where the first order BC was expressed as

$$\left( \frac{\partial}{\partial x} + c \frac{\partial}{\partial r} \right) \varphi_{x=0} \quad (2.75a)$$

$$c = \frac{\omega}{k} \quad \mathbf{r} = \begin{cases} \cos \beta \\ \sin \beta \end{cases}, \quad (2.75b)$$

where  $\beta$  is the incident angle between the wave and the boundary. Radiating boundary conditions give in theory no reflection at the boundary, but when numerical approximation is applied the conditions are no longer valid and therefore some reflection is always present. Formulation of these kinds are very useful for the advection of scalar quantities such as the pressure, acoustic, elastic or electromagnetic waves[22]. These are all linear approximations, that being said the boundary can be set at a location, where non-linearity is avoidable. However, introducing a non-uniform background flow makes the use of velocity potentials inapplicable as well as making the phase velocity hard to predict, and other methods should be considered.

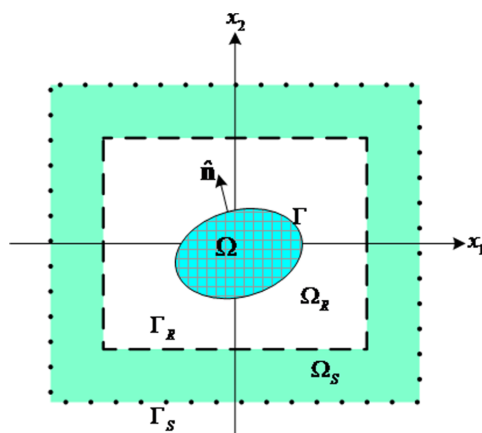
## 2.8.2 Zonal techniques

Instead of formulating an exact boundary condition, strategies exist for developing zones or layers for dampening or absorbing waves in domains. One of the main benefits of

these methods are that they can be quite easily expressed and implemented. Generally they work by adding extra mesh points that stretch, absorb or dampen the wave in some fashion to reduce or avoid reflection at the boundary. In the literature these techniques are often referred to as "absorbing layers", "sponge layers", "buffer zones", "numerical beaches" or similar. As mentioned all the methods in this section increase domain size and are therefore considered as computationally costly. The benefits of using of these methods has to be weighed against this extra cost.

### Perfectly matched layer

In 1994, Perfectly Matched Layer (PML), was formulated by Bèrenger[3] to dampen electromagnetic waves propagating towards boundaries. The idea is that an absorbing layer attenuates the solution and therefore can make simpler to set correct boundary condition on the outer domain. A simple outline of the idea is given in Figure 2.12.



**Figure 2.12:** An overview of a 2D model with the original domain and the PML layer around[55].

Bèrenger developed method for electromagnetic waves, which have behaves as according to the scalar wave equation. The Laplace transformation of (2.72) is

$$s^2 \hat{\varphi} = c^2 \frac{\partial^2 \hat{\varphi}}{\delta x^2}, \quad (2.76)$$

where  $\hat{\varphi}$  is the Laplace transform of  $\varphi$ . This holds true for any waves inside the regular domain, now we introduce a coordinate transformation as is the essence of the PML.

$$\tilde{x}_i = x_i + \frac{i}{\omega} \int_{a_i}^{L_i} \varpi(x_i) dx_i \quad (2.77)$$

where  $\varpi(x_i)$  is the complex stretching factor. The PML starts at a point  $a_i$  and ends  $L_i$  at the end of the domain in any given direction  $i = 1, 2, 3$ . This stretching contributes the actual absorbing of the wave and can have properties of an exponentially decaying function, but of matching impedance as of the system in question. For the PML, the

---

assumption is that the same equation expressed in terms of this coordinate transform, holds true. In terms Bèrangers paper [3], this is in fact matching of PMLs. This can be illustrated looking at the effect the transformation has on the general solution.

$$u(x, t) = e^{-i\left(\omega t + \int_{\alpha_i}^{L_i} \varpi(x_i) dx_i\right)} \quad (2.78)$$

As seen in (2.78), the PML works as a exponentially decaying function unlike the general solution which has a purely oscillating nature. However, it is important to distinguish between oscillating and evanescent modes of waves and how they respond to the PML. Evanescent waves are exponentially decaying and when exposed to the PML they become oscillating and can cause potential problematic behaviour. This is important to consider when designing PML's, given a bigger distance between the PML and the starting point of the evanescent waves it is likely that the waves have decayed substantially enough.

Perfectly matched layers have been modified to be able to account for a mean flow, making able to be used to for instance the linerised Euler [27]. Hu extended his approach for use to both the non-linear Euler as well as the Navier-Stokes equation and thereby making it a very viable option for use in CFD[28]. However, for non-linear equations the perfectly matched is no longer perfectly matched due to the conversion from Fourier transformed expressions to time domain equations. Non-linearity in the flux vector excludes the formally perfectly matched layer. It is still effective, but the performance is decreased if there is strong non-linearity present inside the perfectly matched layer.

### Artificial dissipation and damping

In the added parts of the domain one can customize the dissipation and/or add a damping term to reduce waves travelling toward the exit. This can be added directly into the governing equations and thereby likely keeping them well-posed. Effectiveness has for the one-dimensional wave-equation has been considered by Israeli and Orzag [29]. They posed it as a system of two equations

$$\frac{\partial u}{\partial t} = \frac{\partial v}{\partial x} + \mu(x) \frac{\partial^2 u}{\partial x^2} - \nu(x)u \quad (2.79)$$

$$\frac{\partial v}{\partial t} = \frac{\partial u}{\partial x} \quad (2.80)$$

where both  $\mu, \nu \geq 0$ . Dissipation alone was found to be rather ineffective, needing over 15 wavelengths to achieve a reflection of 1%. Damping was found to much more effective and was considered only to need about one and a half wavelength to achieve minimal reflection.

Damping is therefore a much more used technique and has been used effectively also for the Navier-Stokes equation. Freund used this for aerodynamic soundwaves simply by adding the damping as a pressure term in governing equation[19].

$$\frac{\partial \mathbf{u}}{\partial t} = L(\mathbf{u}) - \nu(\mathbf{u} - \mathbf{u}_0) \quad (2.81)$$



---

Term	Description
$\hat{z}$	vertical direction along gravity
$\hat{x}$	flow direction
$S$	Momentum sink term in the $z$ direction
$C_1$	linear damping resistance [1/s] (default value of 10)
$C_2$	quadratic damping resistance [1/m] (default value of 10)
$V$	velocity in the $z$ direction
$x$	distance from free surface
$z$	distance along the flow direction $\hat{x}$
$f(x)$	damping function in the $\hat{x}$ direction
$f(z)$	damping function in the $\hat{z}$ direction

**Table 2.1:** A detailed description of the terms in (2.85)

Here,  $\mathbf{u}_0$  is a aimed value of the velocity field, which depend on the nature of the problem. A similar approach is to damp out the disturbance directly in every time loop with an appropriate function.

$$\mathbf{u} = \mathbf{u}_{ref} + \xi(x) (\mathbf{u} - \mathbf{u}_{ref}) \quad (2.82)$$

This damping technique is used in commercial software such as Ansys Fluent and Star CCM+ for dealing with waves travelling out of the domain. According to Peric and Maksoud most methods of this kind can be generalized as either linear or quadratic damping[43]. Linear wave damping is obtained by adding a source term,  $q_i^{d,lin}$ , to the momentum equation:

$$q_i^d, quad = \rho C_{i,lin} u_i, \quad (2.83)$$

where the linear damping coefficient,  $C_{i,lin}$ , often has a spatial dependency.  $C_{i,lin}$  regulates the strength and gradual blend-in of the damping as this is important to reduce reflections. Reflections still occur at entrance and exits of these layers, however tuning the parameters can decrease the reflections substantially.

Quadratic damping is usually described as

$$q_i^{d,quad} = \rho C_{i,quad} |u_i| u_i. \quad (2.84)$$

This is very similar to the linear damping, however it is weighted by the magnitude of the velocity, causing particles of higher velocities to experience more damping. In the Fluent literature the sink term is defined

$$S = -\rho f(x) f(z) \left[ C_1 V + \frac{1}{2} C_2 |V| V \right] \quad (2.85)$$

The damping functions are determined by scaling factors according to

$$f(x) = r_x^2 \quad (2.86a)$$

---


$$f(z) = 1 - r_z, \quad (2.86b)$$

where the scaling factors,  $r_x$  and  $r_z$ , are determined by

$$r_x = \frac{x - x_s}{x_e - x_s} \quad (2.87a)$$

$$r_z = \frac{z - z_{fs}}{z_b - z_{fs}} \quad (2.87b)$$

In Fluent, linear damping is used in flow direction, while quadratic damping is used in the direction of the gravity. Perić and Maksoud asses the use of these in-built functions in their 2016 paper [43]. They also discuss important aspects concerning design of the wave tank and absorber, as well as discussing the evaluation of the data received. In particular, since the waves cannot be perfectly absorbed, unlike in the analytical cases, standing waves will oscillate in time around the actual value.

They further discovered that the quadratic damping did not offer any actual improvement compared to the linear. In fact, the linear seemed to work efficiently for a broader range of wave frequencies than the quadratic. The width,  $x_d = x_e - x_s$ , seemed to have a great effect on the damping quality. Normally, the damping length is recommended as  $1.5\lambda \leq x_d \leq 2\lambda$ , but for set-ups where the parameters of the wave is not known in advance such as waves reflected of bodies, a longer damping zone is recommended.

Perić and Maksoud[43] also introduced a scaling law for the length of the damping layer:

$$x_d = x_{d,ref} \cdot \frac{\lambda}{\lambda_{ref}}, \quad (2.88)$$

and similarly for the damping coefficient

$$f_1 = f_{1,ref} \cdot \frac{\omega}{\omega_{ref}}. \quad (2.89)$$

Based on the results, they also recommend a damping setup:

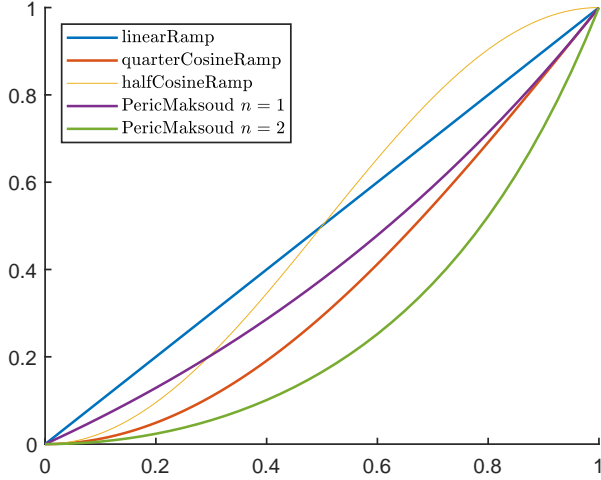
$$f_1 = \Psi_1 \omega, \quad (2.90)$$

with  $\Psi = \pi$ , wave frequency  $\omega$ ,  $x_d = 2\lambda$  and  $n = 2$

In OpenFoam, similar ramping options exist for wave damping. The fvOption verticalDamping can be added as a source term to the momentum, where a set constant is multiplied with the vertical velocity similar to in Fluent and Star+ CCM. Ramping function can be used to blend the source term in order to cause minimal reflections. In Figure 2.13, three of these ramping function are plotted against the ramping function used by Perić and Maksoud. Perić and Maksoud reported best results for their function

### Grid stretching and numerical filtering

Another way of attenuating the solution can be achieve in a simple numerical way. Stretching the grid in the wave propagating direction to the extent that the grid becomes coarse



**Figure 2.13:** Ramping options in OpenFOAM plotted against the functions provided in Star+ CCM

enough to under-resolve the wave. Grid stretching is computationally equivalent to coordinate modification. In the x-direction this can be written as

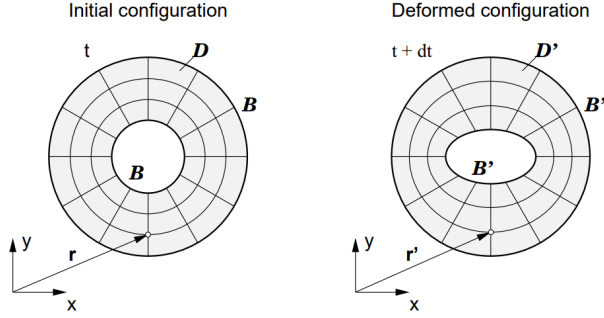
$$\frac{\partial}{\partial x} \rightarrow \alpha(x) \frac{\partial}{\partial x}. \quad (2.91)$$

This must however, be used with caution as sharp transition to a coarser grid can cause numerical reflection in itself. Grid stretching can also cause high frequency grid-to grid oscillations and therefore high-order numerical filters should be considered.

Filtering techniques similar to what is used in Large-Eddy Simulations (LES) has been developed as an absorbing layer for compressible flows [4]. By using a Fourier transform on the governing equations and then filtering the equation in Fourier space, flow disturbances can be dampened out. The filtering area is controlled by a window function in analogy to signal processing, which is restricted to a smooth function as for instance the tanh function. A second filtering is performed on the physical grid and a model for the unresolved scales is also suggested.

## 2.9 Dynamic Mesh Handling

Dynamic meshes involves any automatic changes performed by the solver during a simulation, including moving and adaptive meshes. These types of problems are normal to find in any type moving boundaries such as a moving piston or a prescribed motion of wing actuators. One separates between two main dynamic mesh actions, mesh deformation and topological changes. Mesh deformation is caused by moving boundaries that moves points by a prescribed velocity. Topological changes are more complicated as they involve changes in the number and/or connectivity of points, faces and cells.



**Figure 2.14:** Mesh deformation problem[31]

Methods for dealing with dynamic meshes are dependent on the discretization. The Arbitrary Lagrangian-Eulerian Finite Element Method (ALE-FEM) is considered a textbook subject, while extensions to finite volume methods have been present since the 1980s[7]. Moving mesh FVM takes advantage of the integral form of the governing equations over an arbitrary moving volume  $V$  bounded by a closed surface  $S$ . The general formulation for an tensorial property  $\phi$  is then

$$\frac{\partial}{\partial t} \int_V \rho \phi \partial V + \oint_S \rho \mathbf{n} \cdot (\mathbf{u} - \mathbf{u}_S) \phi \partial S + \oint_S \rho \gamma_\phi \mathbf{n} \cdot \nabla \phi \partial S = \int_V s_\phi \partial V, \quad (2.92)$$

where  $\rho$  is the density,  $\mathbf{n}$  is the outward pointing normal vector on the boundary surface,  $\mathbf{u}$  is the fluid velocity,  $\mathbf{u}_S$  is the velocity of the boundary surface,  $\gamma_\phi$  is the diffusion coefficient and  $s_\phi$  the volume source/sink of  $\phi$ . The rate of change of the volume  $V$  and the velocity  $\mathbf{u}_S$  are related through the space conservation law (SCL)[31]:

$$\frac{\partial}{\partial t} \int_V \partial V - \oint_S \mathbf{n} \cdot \mathbf{u}_S \partial S = 0 \quad (2.93)$$

A general mesh deformation problem can be stated as follows. Let  $\mathcal{D}$  represent a domain configuration at a given time  $t$  with its bounding surface  $\mathcal{B}$  and a valid computational mesh as given in Figure 2.14. During a time interval  $\delta t$ ,  $\mathcal{D}$  changes shape into a new configuration  $\mathcal{D}'$ . A mapping between  $\mathcal{D}$  and  $\mathcal{D}'$  is sought such that the mesh on  $\mathcal{D}$  forms a valid mesh on  $\mathcal{D}'$  with minimal distortion of control volumes.

Solving the mesh deformation is non-trivial and there are several strategies to accomplish. One way to consider this by a spring analogy which aims to link each point of the mesh by fictitious springs. However, this can lead to failure modes as shown by Jasak and Tukovic[32]. These failure modes can be avoided introducing non-linear and torsional springs. The cost however, is huge and this method suffers greatly from that. The Laplace smoothing equation offers a totally different strategy. In  $\mathcal{D}$

$$\nabla \cdot (\gamma \nabla \mathbf{u}_D) = 0 \quad (2.94)$$

where  $\mathbf{u}_D$  is the mesh deformation velocity and mesh deformation can be found from

---


$$x_{new} = x_{old} + \delta t \cdot \mathbf{u}_D \quad (2.95)$$

An important thing too note that the Laplace smoothing equation does not take into account coupling of motion due to rotation. To achieve this one must use a pseudo-solid equation, which are simplifications of the non-linear spring problem. In the thesis, only Laplace smoothing equation has been used as there is no rotational mesh motion.

The diffusivity,  $\gamma$ , can have a large effect on the distribution of the movement of the cells. Too high values can lead to self intersecting or other ill conditioned meshes due to too large displacements of the grid points [1]. The diffusivity can also be varied from the moving boundary, which is usually done by making it inverse or square inverse proportional with the distance from the moving boundary. If the mesh movement is large, one must consider making topological changes to keep the mesh at an acceptable quality. This will not be considered further in this text as mesh deformation is kept at a significantly low level.

## 2.10 Frequency analysis

In signal processing, it is very useful to perform frequency analysis, which can be made through the Fourier transform of the signal. Fourier transforming the signal is done by using a fast Fourier transform (FFT) algorithm. The FFT algorithm computes the discrete Fourier transform (DFT) of a sequence or its inverse. This a very helpful tool for quick switching between the time domain and the frequency domain. The discrete Fourier is defined by:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N}, \quad (2.96)$$

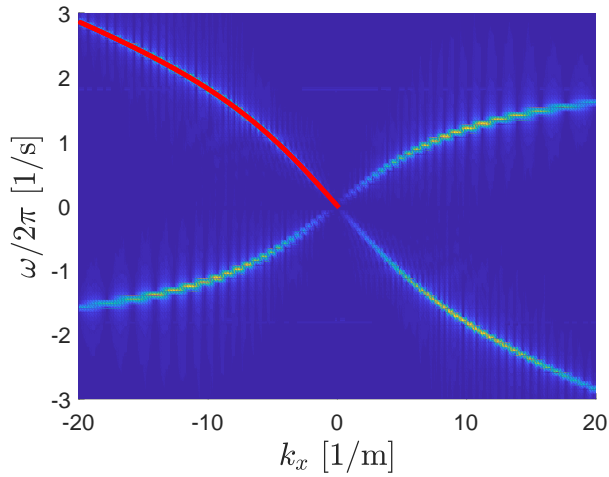
where  $N$  is the amount of numbers in a sequence. An example is shown here, as this process is much used in the post-processing stages of the work done in this text. Assume a surface is has an initial Gaussian deformation and over a uniform current then the surface as a function of time and space. The expression can be easily derived by combining (2.20),(2.21) and (2.22) with the vorticity  $S$  set to zero. The expression is then :

$$\zeta(x, t) = \beta(k)e^{ik(x-U_0t+ct)} + \beta(k)e^{ik(x-U_0t+ct)}, \quad (2.97)$$

where  $c = \sqrt{\frac{g}{k} \tanh kh}$ . The amplitude  $\beta$  can is found from

$$\zeta(x, 0) = 2\beta e^{ikx} = (e^{-(x/a)^2}). \quad (2.98)$$

Then the dispersion relation is plotted in Figure 2.15. In this example, a large number of frequencies are available from the analytical solution and it distributes nicely. Since the example has a uniform current, the dispersion has different shape upstream then downstream. This has do with that upstream of the initial disturbance, the waves have a lower wavelength, while downstream the current "stretches" the wave and therefore the wave is longer. If the current is set to zero, both sides are identical.



**Figure 2.15:** Frequency analysis of an Gaussian initial wave deformation over a uniform current

## 2.11 Summary

In the theory section, it was uncovered the necessary relations, parameters and equations that are to describe waves and wave theory. A broad presentation was also given of numerical representations of free surfaces as well as the absorbing boundary conditions. The VoF-method was selected as in OpenFoam, the interFoam solver is fully integrated with this. However, a Euler-Lagrangian approach is also implemented for the dynamic mesh handling used to oscillate cylinder in chapter 4. A damping layer was chosen to handle reflections on the boundaries. The damping layer was chosen both due to the fact that its already implemented in OpenFoam, but also because it has abilities to dampen broad wave spectrum's. It also is universally defined, without considering inflow or outflow conditions.

# Chapter 3

## Method

Numerical methods are used in this thesis to perform analysis of surface waves. The open-source software OpenFOAM, which is an open-source CFD program, is the software of choice to perform the analysis. This section will quickly cover into what kind of software and hardware that has been used in the thesis, before explaining about meshing and mesh quality. Then the solver and solver settings will be discussed, before doing a test case to determine an optimal setting for the wave damping parameter  $f$ .

### 3.1 Software

OpenFOAM is the main tool for the CFD analysis, in addition Paraview and Matlab is used for post-processing. Paraview is also open-source software, which is the most common tool for viewing results from OpenFOAM and is easily accessed via the terminal window. In OpenFOAM, most cases are generally set-up through three folders; 0, constant and system. The 0-folder contains the boundary conditions, the constant folder includes fluid properties and the polyMesh-folder, which describes the mesh. In addition, the system-folder includes a number of files that controls the solvers, schemes and other important settings. Commands are then executed in a terminal window, instead of using an interactive tool as is common in most commercial CFD softwares. The CAD model and mesh are created with the ANSYS package, by the use of Design Modeler and ANSYS Meshing. Meshing can be performed directly in OpenFOAM, as well by using the blockMesh tool or from a CAD model with snappyHexMesh. However, due to familiarity with the ANSYS package, this was chosen to ease the work flow. A mesh can easily be imported by using the fluentMeshToFoam command, however care must be taken as to which patch types and boundary conditions are important as these may not be what was originally intended.

Post-processing was done in Paraview and MATLAB, where Paraview generally was used to visualize the flow during simulation time as well as used for filtering. The filters used were a contour set at  $\alpha = 0.5$ , and a cell-center filter to extract only the value at a cell center, as Paraview interpolates to cell corner values. This was exported as a .csv file, which could then be reread into MATLAB. In case of any multiple entries, linear interpolation was used to find  $\alpha$ , and then the surface height was simply  $\zeta = \alpha \cdot y$ .

Simulation time varied from 8 hours to 24 hours and was performed on a Dell Latitude E6420.

## 3.2 Meshing

### 3.2.1 Mesh Quality

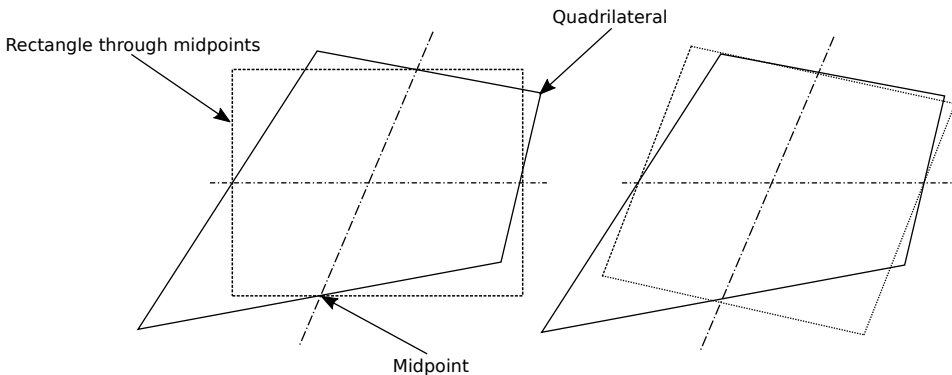
In order to determine whether a mesh has been properly or poorly designed, mesh quality parameters are used for a quantifiable . ANSYS Mesh uses element quality, aspect ratio, Jacobean ratio, warping factor, parallel deviation, maximum corner angle, skewness and orthogonal quality as mesh metrics. The definition has been gathered from ANSYS Documentation, Mesh Metric. The most parameters used to measure the quality are explained below.

Element quality is a composite quality metric ranging from 0 to 1. It is based on the ratio of the volume to the sum of the square of the edge length for 2D quad elements. Element quality is expressed as

$$\text{Quality} = C \frac{\text{Area}}{\sum (\text{Edge length})^2} \quad (3.1)$$

for 2D elements, where  $C = 4$  for quad elements.

Aspect ratio is simply the ratio between sets of orthogonal edges, when the edges are not aligned they are calculated according to Figure 3.1.



**Figure 3.1:** Illustration of how ANSYS Mesh calculates aspect ratio

Skewness is a measure how close an ideal i.e equilateral or equiangular a face or cell is. Highly skewed elements have values close to 1 and at exactly 1 the element is completely degenerate. If the skewness of an element is close to 0, the element is almost equilateral and considered an ideal element.

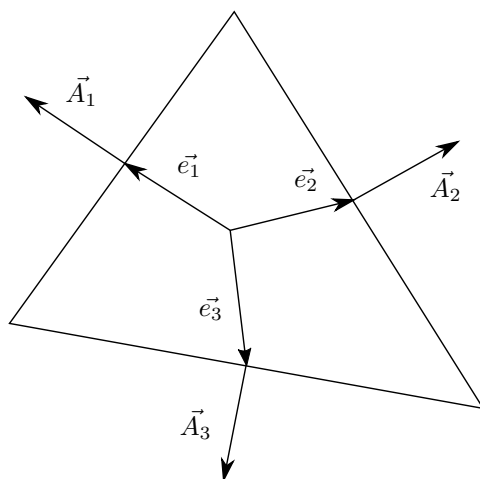
Orthogonal quality for a face is computed as the smallest cosine of the angle between the edge normal vector,  $A_i$  for each edge and the vector from the face centroid to the centroid of each edge,  $e_i$ .



---

Value of Skewness	Cell Quality
1	degenerate
0.9 - < 1	bad(sliver)
0.75 - 0.9	poor
0.5 - 0.75	fair
0.25 - 0.5	good
> 0 - 0.25	excellent
0	equilateral

**Table 3.1:** List of skewness values and the corresponding cell quality



**Figure 3.2:** Vectors used to compute orthogonal quality for an element

### 3.3 Solver and numerical settings

#### 3.3.1 Solver

In openFoam, the standard solver for multiphase simulations is interFoam. This solver utilizes the VOF method, by solving an extra equation for the phase fraction  $\alpha$ . The interFoam solver has been thoroughly tested in literature [8], where performance tests were done on the solver. Deshpande, Anumolu and Trujillo also cover a large section of references to the use of the solver not based on the actual performance, but measured against experimental, previous numerical and analytical data. These include, but are not limited to; a comparison of the droplet impact and crater formation with their experiments [2], microfluidic flow in rectangular channels with an analytical solution [48] and a comparison of the velocity profiles for a flow around a partially submerged cylinder with experimental data [36]. As a part of the verification they tested a standing wave in 2D, which showed

---

excellent agreement in terms of frequency and small overshoots in the wave amplitude in the inclusion surface tension effects, which also could be found in other numerical solvers.

For the oscillating cylinder cases, the `interDyMFoam` is used because of its ability to handle dynamic meshes. The `interDyMFoam` solver is essentially an extension of the `interFoam` solver, solving the multiphase equations equally, but has an inclusion for handling mesh operation such as node movement, removal and addition.

### 3.3.2 Settings

#### Solver

The `interFoam` solver utilizes the PIMPLE or PISO algorithm depending on the number of outer corrections in the `nOuterCorrectors` in the `fvSolution` file found in the system folder. For a value of 1 the PISO-algorithm is used and for larger value the PIMPLE-algorithm is chosen. For each iteration of `nOuterCorrectors` the velocity matrix pressure corrections are made and from these pressure corrections a new velocity matrix is built based on the new flux field, then pressure corrections are again calculated and the flux field is corrected. The number of cycles the pressure is corrected based on the old flux field is `nCorrectors`. By specifying a tolerance based on the initial residual for each variable, the number of `nOuterCorrectors` can be set to a large number and the solver finished the loop when all tolerance criterion are met. Relaxation factors were used for stability purposes and they were set for the velocity vector and pressure field at 0.6 and 0.4 respectively. The viscosity and density are inherently different in water and air. This jump causes convergence and numerical instability due to the phase fraction. Therefore  $\rho_{water}/\rho_{air} = 100$  and  $\mu_{water}/\mu_{air} = 100$  was chosen following Reichl et.al [44]. This only affected the air properties and water was chosen to have its regular values for the density and viscosity.

#### Numerical schemes

The numerical schemes in OpenFoam are set in the `fvSchemes` file also in the system folder. The schemes are chosen for; time, gradients, advection terms, diffusion terms and interpolation. The numerical schemes used in the simulations are listed in Table 3.2.

Term	Scheme
Time	Crank Nicholson
Gradients	Linear
Advection, Velocity	linearUpwindV
Advection, Phase Fraction	van Leer
Diffusion	Linear corrected
Interpolation	Linear

**Table 3.2:** Numerical schemes used in all simulations

All schemes, except time, are second order accurate and are linear schemes, except for the advection schemes. The advection of velocity is a delicate non-linear term, which

---

can become unstable for a linear scheme, therefore the linearUpwind scheme was chosen. The van Leer scheme, was chosen because it is a bounded scheme, which also helps with stability and ensures a smooth transition over the surface, where phase fraction jumps.

### Time stepping

Time stepping was set by a calculated time-step, where the max Courant number was a set value. OpenFoam uses cell volumes instead of grid spacing for calculating the Courant number, in order to account for variable grid spacing in all directions. For easy start-up the simulation was initiated by an Euler time-stepping scheme (value of 0), which was ramped up during the first second of simulation time to a value of 0.9 in the Crank-Nicholson weighting scheme, which was kept throughout the simulation.

## 3.4 Damping layer case

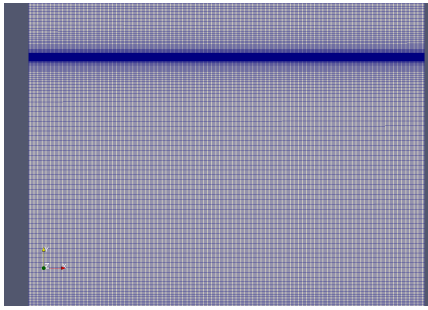
In order to treat waves approaching the boundaries a damping layer was deemed necessary. The damping layer was added by the fvOption tool in openFoam, verticalDamping. This function adds a force term which works in the direction of gravity and is intended to damp vertical motion of an interface in the region approaching an outlet. A constant of proportionality has to be determined. This is determined from an initial case, where waves of amplitude  $a = 0.08m$  and wavelength  $\lambda = 4m$  enters from the left and propagates towards the exit. The length of the mesh,  $L_{tot} = 6\lambda$  and the damping layer starts at  $x = 4\lambda$  and its length is  $L_{damp} = 2\lambda$ . The vertical domain size was  $L_{vert} = 4.5\lambda$  and the depth is,  $d = 4\lambda$ .

### 3.4.1 Mesh

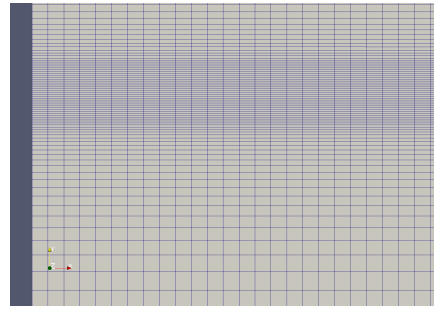
An orthogonal, quadrilateral and structured mesh was created in Fluent Mesh, where 10 cells per amplitude and 50 cells per wavelength were used. The number of cells were decided from Peric and Maksoud's paper [43] and ITTC's guidelines which states that "Use no less than 40 grid points per wavelength on the free surface ... Use no less than 20 grid points in the vertical direction where the free surface is expected" [30]. Resolution is most important where gradients are large i.e where the flow changes fastest. In this case that is only close to the surface. The bottom was given a no-slip boundary condition, but the boundary layer was not of interest and therefore not resolved. The growth rate away from the free surface and bottom was 1.1 to achieve good resolution on the free and have excellent mesh quality. The mesh size was therefore at 46800 cells.

Quality parameter	Value
Aspect Ratio	$Min = 1$ $Max = 8$
Mesh non-orthogonality	$Max = 0.012$ $Avg = 0.005$

**Table 3.3:** Quality parameter for the mesh



(a) An overview of the mesh



(b) Close-up in the free surface zone

The quality parameters in Table 3.3 are taken from OpenFoams checkMesh utility. The definition for mesh non-orthogonality is different from the orthogonal quality in Fluent Mesh. Generally speaking, here a value close to 0 is good quality, while Fluent have good quality close to 1.

### 3.4.2 Boundary conditions and initializations

All simulations in this section was initialized using the openFoam tool funkySetFields, where  $\alpha = 0$  for  $y \leq 0$ , which is equal to an undisturbed water surface. The hydrostatic pressure and the velocity field was initialized as zero in the entire field.

#### Inlet

The velocity and phase-fraction boundary conditions at the inlet was, as mentioned, given by an fifth order Stokes wave model, with no background flow. Values are gathered from a table and mapped onto the inlet patch. The hydrostatic pressure  $p_{rgh}$  is set by a fixedFlux-Pressure. This boundary condition sets the pressure gradient to the provided value such that the flux on the boundary is that specified by the velocity boundary condition.

#### Outlet

At the outlet a zero gradient boundary condition is used for the velocity and the phase fraction. A fixed value of 0 was used for the hydrostatic pressure.

#### Top

The velocity at the top patch was set as an pressureInletOutletVelocity condition. It is mixed inlet/outlet boundary condition, where pressure is specified from the pressure boundary condition. Outflow or inflow is decided from the sign of the flux, in case of outflow, zero-gradient is applied. For inflow the velocity is found from the patch-face normal component of the internal-cell value. The hydrostatic pressure is set according to a totalPressure boundary condition:

$$p_p = p_0 - \frac{1}{2}|\mathbf{u}|^2. \quad (3.2)$$

This pressure BC describes the the patch-pressure,  $p_p$  by subtracting the dynamic pressure from the total pressure  $p_0$ , which is in this case  $p_0 = 0$ . The phase-fraction is a mixed condition inletOutlet, where the inlet value is set as 0 or in other words air.

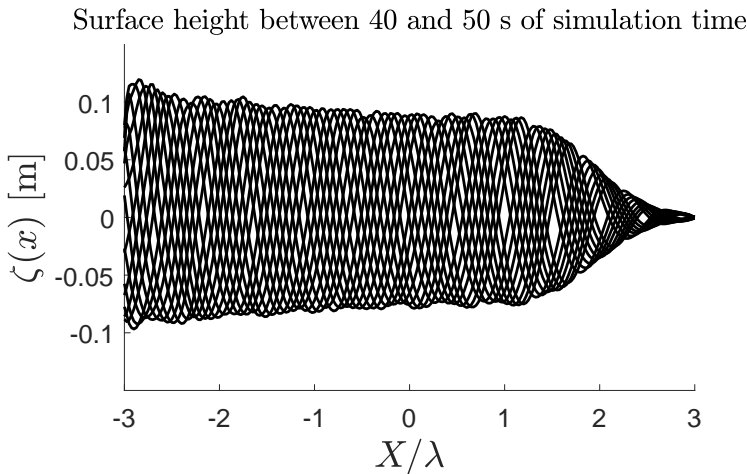
### Bottom

The bottom has a no-slip boundary condition for the velocity,  $U = 0$ . The phase-fraction has a zero-gradient boundary condition. The pressure is set by the fixedFluxPressure.

### 3.4.3 Optimizing the damping parameter $f$

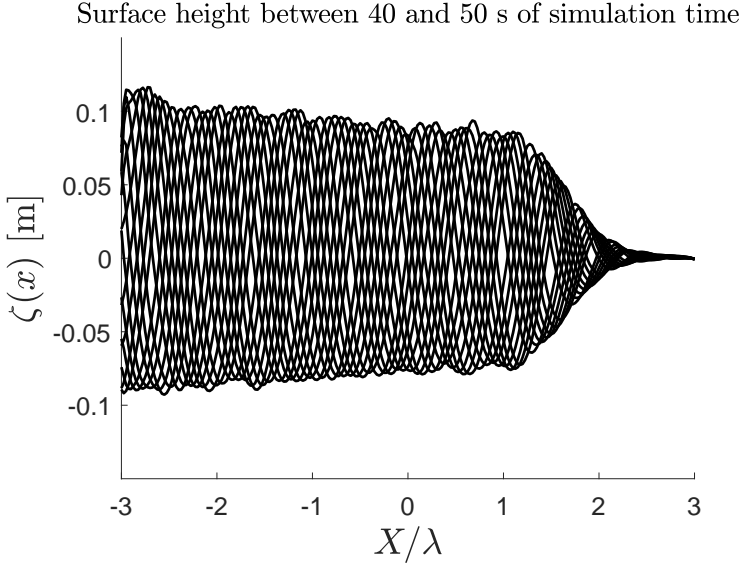
This setup is very similar to the one Peric and Maksoud [43] used to propose a range of guidelines and a scaling law for practical purposes of a damping layer. The idea was to compare the quarterCosineRamp with Peric and Maksouds result for the ramp functions in Star CCM+.

Initially, the simulations was run for 20s of simulation time, with the PISO-algorithm ( $n_{\text{OuterCorrectors}} = 1$ ). The number of inner loops ( $n_{\text{Correctors}}$ ) was three, an extra loop was added for non-orthogonal corrections ( $n_{\text{NonOrthogonalCorrectors}}$ ), and no relaxation factors were added. After this start-up phase, relaxation factors were added (0.4 for  $p_{rgh}$ , 0.6 for  $U$ ) and the PIMPLE-algorithm was run with tolerance at  $1 \cdot 10^{-4}$  for the pressure and  $1 \cdot 10^{-5}$  and the number of  $n_{\text{OuterCorrectors}}$  was increased such that the tolerance was reached until 50s of simulation time. Following Peric and Maksoud [43], the damping parameter,  $f$  was initially set to  $\Psi\omega$ , where  $\Psi = \pi$ . The frequency was calculated from the dispersion relation, which for  $\lambda = 4m$  gave an omega  $\omega = 3.92\text{rad/s}$ . This gave a value for the damping coefficient at  $f \approx 12\text{rad/s}$ .



**Figure 3.4:** Surface height for the entire domain plotted for the last 6 periods for  $f = 12$

From Figure 3.4, it was observed that reflections were present, which indicated that the damping coefficient was not at an optimal value. This led to the conclusion that Peric and Maksoud's parameters were not directly applicable to OpenFoam's quarterSineRamping function, despite the fact that show very similar behaviour as seen in Figure 2.13. Therefore, the simulation was rerun with  $f = 40\text{rad/s}$  to establish an appropriate value, which could be used as a basis for the rest of the simulations. The simulation for the new damping coefficient is portrayed in Figure 3.5.

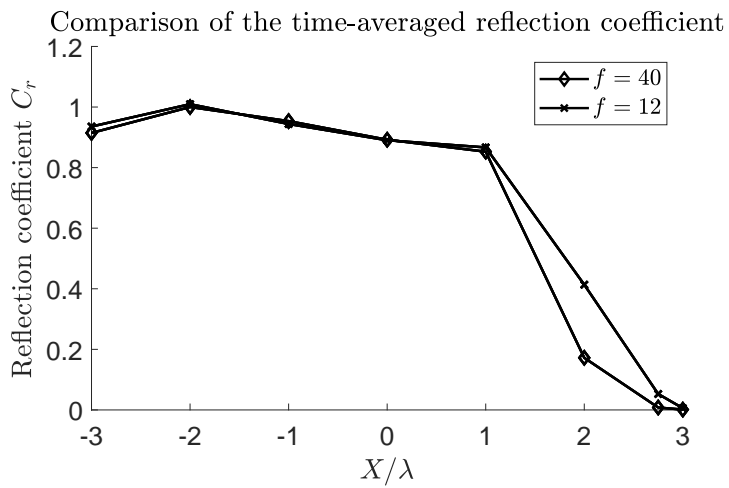


**Figure 3.5:** Surface height for the entire domain plotted for the last 6 periods for  $f = 40$

In order to look quantitatively on the effect of the damping coefficient, a reflection coefficient,  $C_r$ , was calculated from the largest and smallest amplitude measured at 8 different location starting at  $x = -12$  with a distance of  $\lambda$ . The largest and smallest amplitude is taken to the most positive and negative value at any given point during the course of a wave period. The reference amplitude was taken to be the amplitude at  $x = -8$ . The coefficient,  $C_r$  is given by:

$$C_r = \frac{\max(a) - \min(a)}{\max(a_{ref}) - \min(a_{ref})}. \quad (3.3)$$

The computation of the wave model at the inlet boundary seemed to struggle with accurately predicting the wave at the boundary. This is shown in Figure 3.6, where one can observe that the shape of the wave is visibly different as well as that the water level seems to be at a slightly different height. The reason for this was suspected to relate to the boundary condition for the hydrostatic pressure. The wave model was not accounted for in the definition of the pressure boundary condition, which may have been the reason for the slightly off-put value. Therefore the reference was chosen to be set at  $x = -8$ .



**Figure 3.6:** Reflection coefficient for the averaged value over 6 wave periods





# Chapter 4

## Case Studies

These case studies encompasses three different cases; oscillating cylinder with quiescent water, oscillating cylinder with uniform velocity and an oscillating cylinder moving at uniform velocity over a shear current. For still water, the simulation validates the use of an oscillating cylinder as a wavemaker by comparing the solution to the dispersion relation for linear surface waves.

### 4.1 Oscillating cylinder - Quiescent water

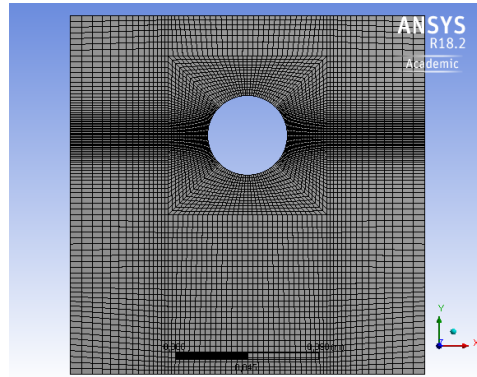
This case was performed on a domain of  $(x, y) = [-3.33.3 - 0.150.075]$ , with an initially at rest surface at  $y = 0$ . At the origin a cylinder of diameter 0.05 m was placed and oscillated following a sinusoidal trajectory  $y_{cylinder} = a \sin(\omega t)$ , where  $a = 0.0075m$  is the amplitude and the frequency is run at  $\omega = 4\pi$ .

The meshing was divided into two different parts, where the mesh around the cylinder was performed in ANSYS Meshing and then imported into openFoam. In openFoam, the mesh was extruded in the x-direction for  $6 \lambda$  and stretched in the damping zone for  $2 \lambda$  at both sides. In the extruded part there were 50 cells per wavelength and 20 cells per amplitude and in the damping zone the cells were stretched with a growth rate of 1.2. The number of cells then numbered 70240 in total. In Figure 4.1, the mesh around the cylinder is shown and Figure 4.2 shows the outline of the domain used.

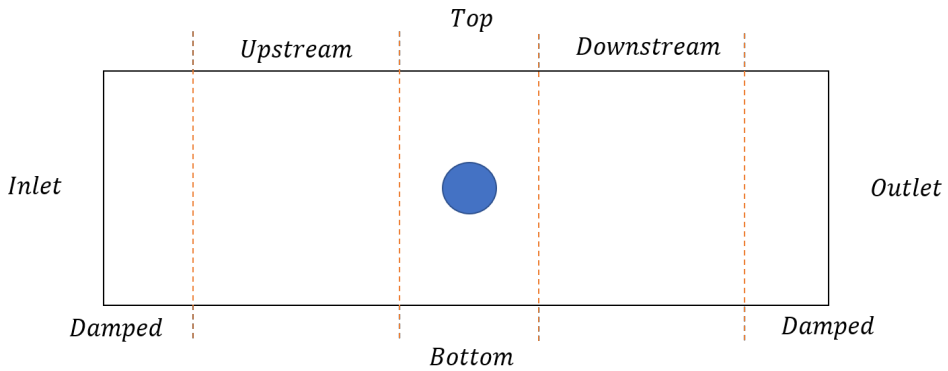
Quality parameter	Value
Element Quality	$Min = 0.2 \ Max = 0.99 \ Avg = 0.75$
Aspect Ratio (original mesh)	$Min = 1 \ Max = 9.8 \ Avg = 2.2$
Aspect Ratio (stretched mesh)	$Max = 186.169$
Orthogonal Quality	$Min = 0.71 \ Max = 1 \ Avg = 0.98$

**Table 4.1:** Quality parameter for the original mesh

In Table 4.1, quality parameters are listed for the mesh. The stretched part had a lot worse aspect ratio due to the stretching, which was expected. The stretching was used for



**Figure 4.1:** Original mesh around the cylinder



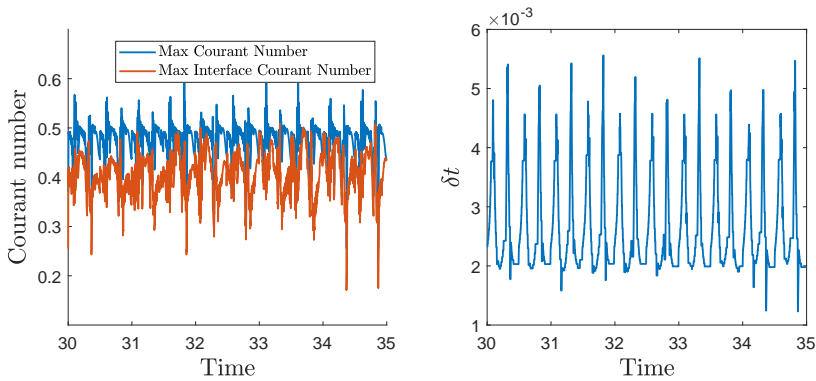
**Figure 4.2:** The domain setup for the moving cylinder cases

two reasons; firstly to damp out incoming waves by poorly resolving them causing dispersion and secondly by reducing the number of cells for faster computing. The boundary conditions are given in Table 4.2.

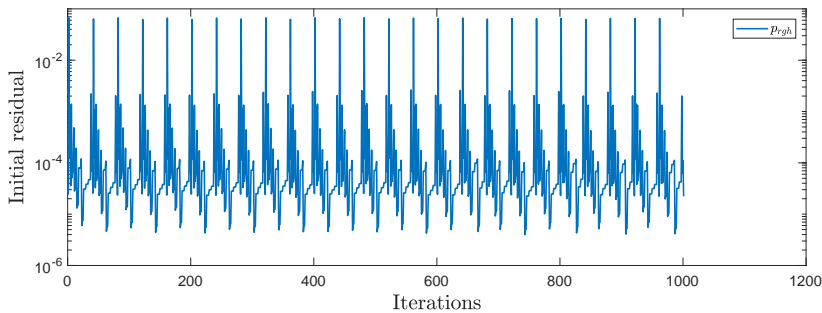
Boundary	$U$	$\alpha$	$P_{rgh}$	Displacement
Top	pressureInletOutletVelocity value (0 0 0)	inletOutlet	totalPressure	fixedValue 0
Bottom	slip	zeroGradient	fixedFluxPressure	fixedValue 0
Cylinder	movingWallVelocity	zeroGradient	fixedFluxPressure	oscillatingDisplacement $y_{cylinder} = a \sin(\omega t)$
Inlet	zeroGradient	zeroGradient	fixedFluxPressure	fixedValue 0
Outlet	zeroGradient	zeroGradient	fixedFluxPressure	fixedValue 0

**Table 4.2:** Boundary conditions for quiescent water

The simulation was run for 20s with a variable timestep with the PISO algorithm, where the timestep was calculated from the last Courant number value and the maximum



**Figure 4.3:** Courant number and  $\delta t$  plotted for the time interval: [30s 35s]

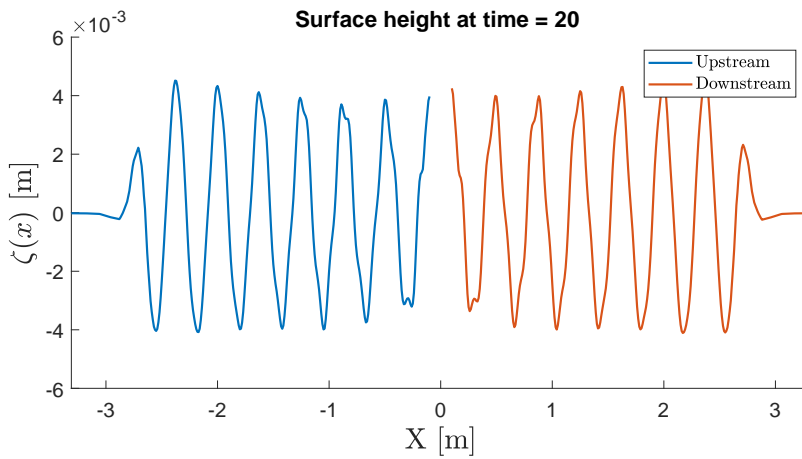


**Figure 4.4:** The last 1000 iterations for the hydrostatic pressure,  $p_{rgh}$

Courant number was set at 0.5. The ITTC guidelines [30], recommend "100 time steps per period for regular waves". The wave period in this case was 0.5 s, which would lead to a recommended minimum of  $\delta t = 0.005$ . From Figure 4.3,  $\delta t$  was found in general to be under the requirements. Usually the flow requirements or restrictions on the Courant number are more strict than the restrictions on the number of time steps per period, which was shown to be an accurate prediction in this simulation and was chosen as the restrictive time parameter for the rest of the simulations.

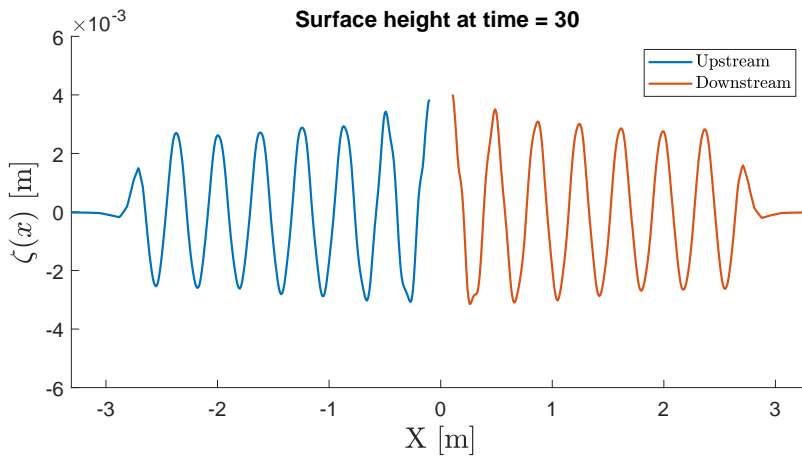
From simulation time 20s to 30s, the number of nOuterCorrectors was increased to 10. The solution was deemed to be periodically time independent at 30s, and was run another 5s for data sampling. In Figure 4.4, a sample of the residuals for the hydrostatic pressure is plotted. This corresponds to approximately 26 time steps. For each time step, a peak value is observed for the first PISO iteration, the initial residual then reduces down below the tolerance criteria, which was for the pressure was set at  $1 \cdot 10^{-4}$ . The tolerance for the velocity was set at  $1 \cdot 10^{-5}$  and then when both criteria was met or if the number nOuterCorrectors was reached, the solution progressed in time.

In Figure 4.5, the surface height is plotted at 20s of simulation time. The wave height is clearly increased towards the damping layer, which gives no physical meaning. One might



**Figure 4.5:** Surface height plot at  $t = 20$  s

expect that the surface height could potentially increase close to the start of the damping layer, but this effect is evident over the entire solution range. This is likely due to two main causes; Firstly, the PISO algorithm is used causing large residuals for the pressure and velocity, secondly the solution has not yet reached a steady, periodically varying state, due to the start-up.



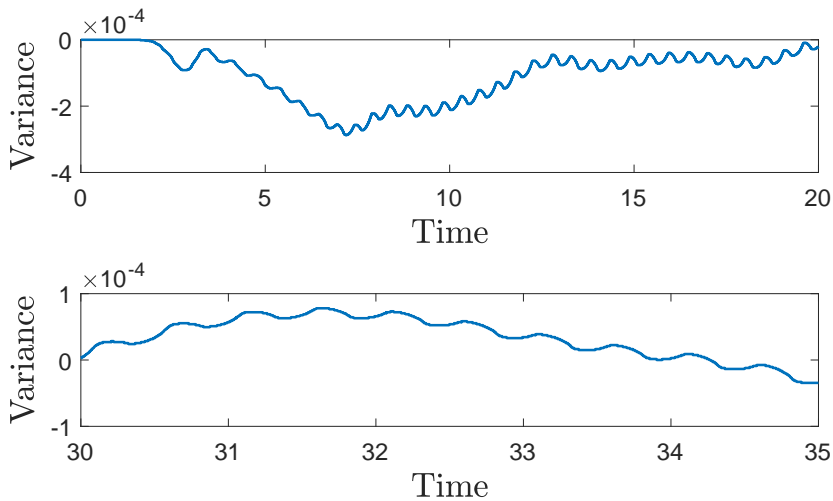
**Figure 4.6:** Surface height plot at  $t = 30$  s

In Figure 4.6, the wave amplitude is largest close to the cylinder and propagating with dispersion towards the exits. This is an indicative sign that the solution have been solved to an adequate accuracy. This can also be observed for the change in volume fraction with respect to time, which is plotted in Figure 4.7.

Another thing of note in Figure 4.6 is that the amplitude of the initial disturbance

caused by the moving cylinder is not directly translatable in the amplitude of the waves in travelling away from the cylinder. This is expected, because the cylinder will only act as crude wave-maker and cannot directly create wave of the same amplitude as it is moved with. Since the amplitude of the waves were smaller, the vertical resolution of the waves was less than optimal. Therefore in subsection 4.1.1, a second case was used to refine the cells across the free surface by a factor of 2.

Near the cylinder a small irregularity is observed, which can be seen in Figure 4.9. The irregularity can also be seen during the first wavelength close to the cylinder in Figure 4.6, but dissipates beyond this due to the grid resolution, and is therefore not considered to influence the solution significantly.

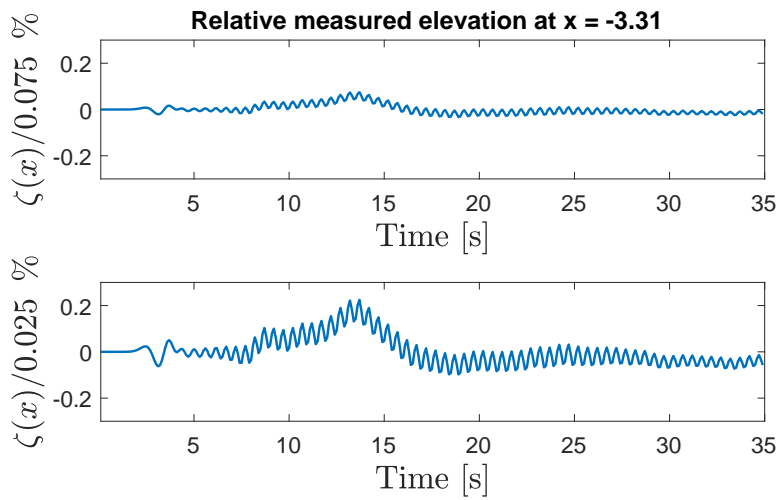


**Figure 4.7:** Volume fraction for two time intervals, where the variance from the initial starting value is given as a function of time

Another thing to consider is how much of the wave reaches the boundaries. In Figure 4.8, the relative surface height in percent is shown. The wave is still present, but the disturbance at the boundary never constitutes to more than a small contribution during the start up. Between 30s and 35s, the mean value of the disturbance was 0.0451% and the variance was 0.0038%.

### 4.1.1 Refined mesh

Since the elevation was lower than originally expected, a new mesh with a higher amplitude-wise resolution was created. The number of cells were increased from 20 to 40 cells, which gave a new mesh size of 84080 cells. Then the simulation was run from 30 to 35 s for the coarse and fine resolution. The effect of increasing the mesh size can be seen in Figure 4.10. There is little effect in the waves of larger amplitude as expected, but a small effect can be seen for the waves close to the damping zones, which are somewhat smaller.

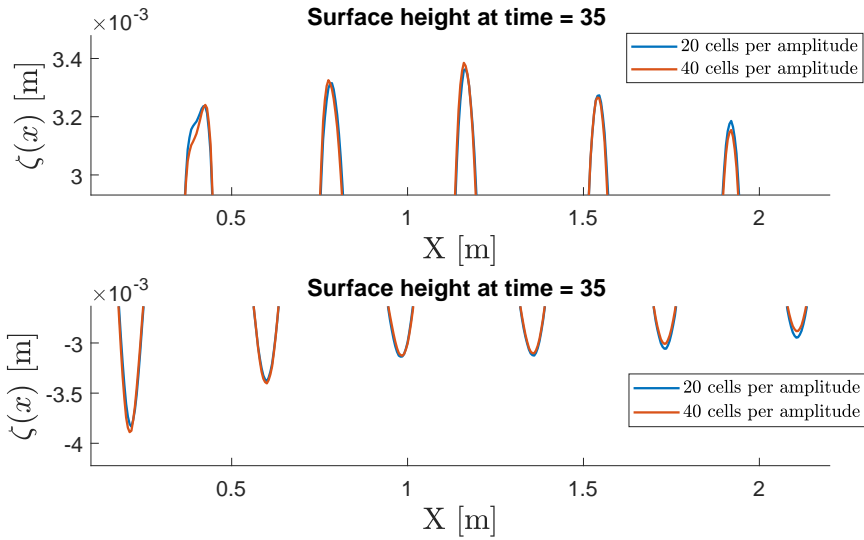


**Figure 4.8:** The surface height at the boundary divided by the amplitude of the initial disturbance and the amplitude of the smallest waves observed at 30s outside the damping zone



**Figure 4.9:** A small irregularity near the cylinder

More of the amplitude is resolved, which causes less smearing of the amplitude.



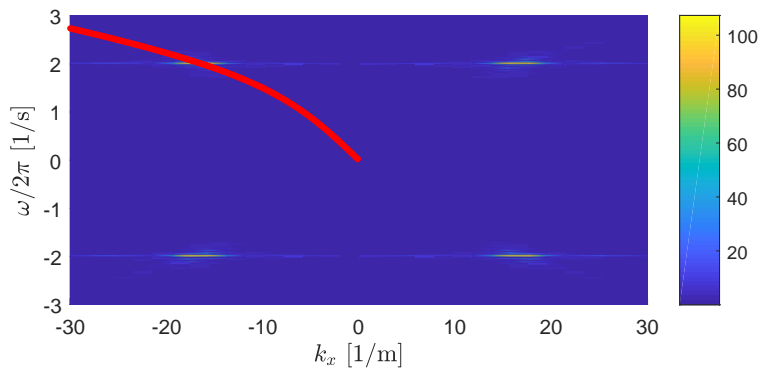
**Figure 4.10:** A comparison of increased resolution amplitude-wise

### 4.1.2 Frequency analysis

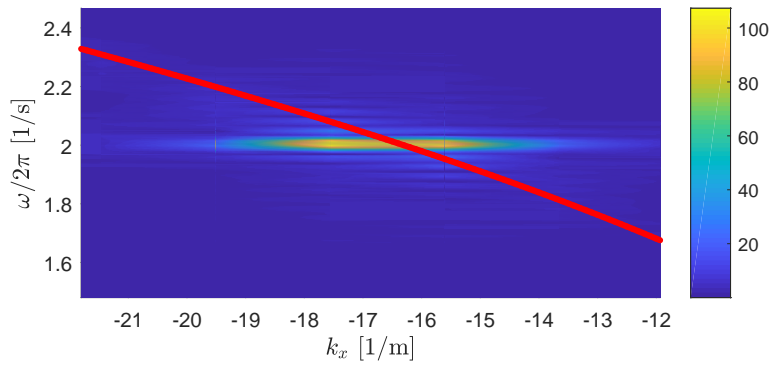
Frequency analysis was used to look at how the solution behaved in terms of the dispersion relation. The surface height, which was calculated from the assumption that it was located at  $\alpha = 0.5$  was transformed by FFT (Fast Fourier Transform), and then compared to the dispersion relation in Figure 4.11a. As the waves were concentrated for a single frequency, there was a lot of information for the excitation frequency and little data for all other frequencies found in the dispersion relation. This did, however, show that the frequency of the waves corresponding to the excitation frequency was accurately portrayed by the oscillating cylinder. From Figure 4.11b, the link between the wavenumber and frequency, can be witnessed more clearly. The FFT-algorithm caused the spectral leakage or smearing seen in Figure 4.11b and Figure 4.11a. From the FFT, a fast assumption could be made to find the largest element in the matrix, which gives  $k \approx 17.56$ . This was checked against the peak to peak distance, which was found at the end of the simulation at 35s. The peak to peak distance gave an average wavelength of  $0.37m$ , which then implies a value for  $k$  around 17. From the dispersion relation, the wavenumber is slightly less,  $k = 16.33$ , which led the peak to peak distance to be about 4% less than the analytical value.

### 4.1.3 Discussion

This case study helped show that an oscillating cylinder could be an effective wave maker, despite the fact that the amplitude of the waves was not one to one with the amplitude of the cylinder oscillations. The waves had closer to half the size of the amplitude of the cylinder oscillation, but a parameter study was not performed to see if this was a proven trend. Instead, a new simulation was run on a refined mesh in the amplitude-direction, and



**(a)** Frequency analysis of the surface height compared to the dispersion relation



**(b)** A closer look on the dispersion.



---

the comparison gave a good correspondence between the two meshes, except for a slight under-prediction for the smallest amplitudes.

Valuable insight was also given comparing the PISO - and PIMPLE - algorithms, where the PISO proved to be too inaccurate causing large residuals for the pressure and velocity. Increasing the number of correctors remedied the problem, however solution time was naturally increased. Generally, the PISO-algorithm was found to be useful to progress faster in time and a switch was made to the PIMPLE- algorithm for time-accuracy.

The damping layer was deemed capable of handle the incoming waves at both ends without causing to much reflections. The amplitude at the boundaries was found to be less then 0.2% of the smallest amplitudes in the domain, which was an excellent result. The volume fraction for the total domain was also used as measure of convergence, where it showed a slight periodic oscillation around the starting value at the end of the simulation, which was taken to be at an acceptable value.

## 4.2 Oscillating cylinder - Uniform flow

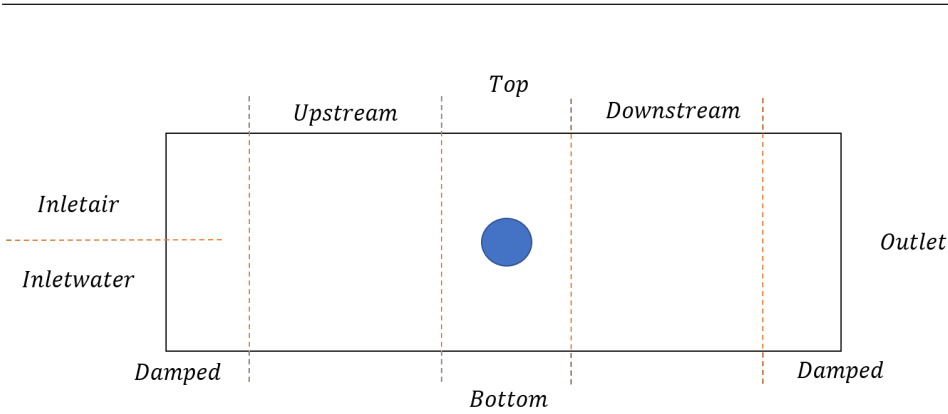
In this section, an oscillating cylinder on uniform flow will be considered. The simulation was performed on a similar mesh as the simulation for still water, but the damping zone near the outlet was extended to be twice as long as for the quiescent water case. This was done to accommodate for the longer wavelength expected at the outlet zone.

The boundary conditions were changed from two outlets at each end to one inlet and one outlet. The inlet was split in two, where in the top end of the inlet  $\alpha$  was set to 0 (air) and in the lower end  $\alpha$  was set to 1 (water). The water was given a velocity  $U = 0.2$  m/s. This was chosen from (2.25), where linear theory predicts a resonance frequency. The boundary condition for the velocity of the air was not that trivial. One could give the air velocity  $U$ , however this could lead to high velocities close to the cylinder, which in turn could increase solution time. Another alternative was to set the inlet to zero, but this an non-physical condition, because the interface between the air works as boundary layer, where the air is given a velocity by sticking to the water surface. Therefore, the air velocity was unknown at the boundary which implied zero gradient condition. From simulations, observations was made that while the air velocity seemed to be properly estimated in the interface, air velocities in other parts of the of the boundary kept increasing. This happened due to the fact that the air was swirling as it hit boundary suggesting inflow. The inlet air boundary was therefore changed to an mixed boundary, where any inflow was accounted for by setting it to zero velocity and this effectively solved the issue.

In addition, the variableHeightFlowRate was introduced at the outlet boundary as it was known to maintain the water level better then a zeroGradient condition. The variable-HeightFlowRate differs from the zeroGradient by having an upper and lower bound. The upper bound was set as 0.8 and the lower bound was set as zero.

$$\alpha = \begin{cases} 0 & \alpha < \text{lowerBound} \\ \text{zeroGradient} & \text{lowerBound} \leq \alpha \leq \text{upperBound} \\ 0 & \alpha > \text{upperBound} \end{cases} \quad (4.1)$$

The cylinder was oscillated at  $\omega = 4\pi$ , and the inflow velocity was  $0.2m/s$ . In order



**Figure 4.12:** The inlet was split in order to control air and water separately.

Boundary	$U$	$\alpha$	$p_{rgh}$	Displacement
Top	pressureInletOutletVelocity value (0 0 0)	inletOutlet	totalPressure	fixedValue 0
Bottom	slip	zeroGradient	fixedFluxPressure	fixedValue 0
Cylinder	movingWallVelocity	zeroGradient	fixedFluxPressure	oscillatingDisplacement $y_{cylinder} = a \sin(\omega t)$
Inletair	InletOutlet	fixedValue 0	fixedFluxPressure	fixedValue 0
Inletwater	fixedValue (0.2 0 0) m/s	fixedValue 1	fixedFluxPressure	fixedValue 0
Outlet	zeroGradient	variableHeightFlowRate	fixedFluxPressure	fixedValue 0

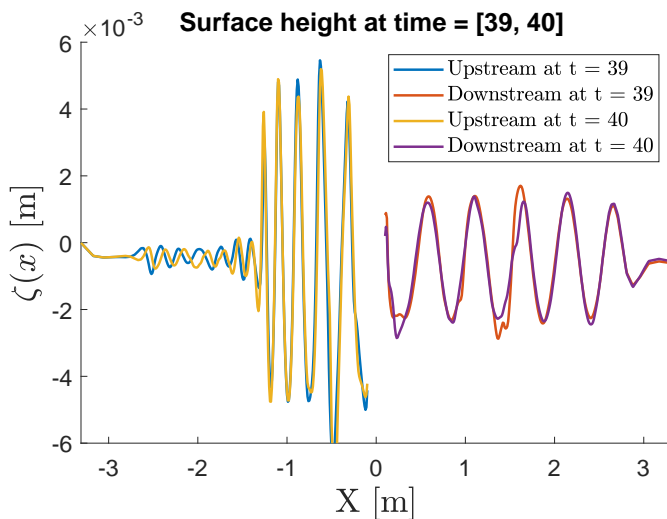
**Table 4.3:** Boundary conditions for uniform flow

to obtain reliable results, the simulation was first run with the cylinder at rest on a coarse mesh. Then when the solution was deemed as time-independent, the cylinder was set to oscillate and run again until time-independence. The solution was then mapped onto a fine mesh, and then run for 40 seconds of simulation time, corresponding to 80 periods of oscillations. There was still some degree of unsteadiness in the solution, which can be seen in Figure 4.13, where the surface height is plotted for two time instances with equal phase.

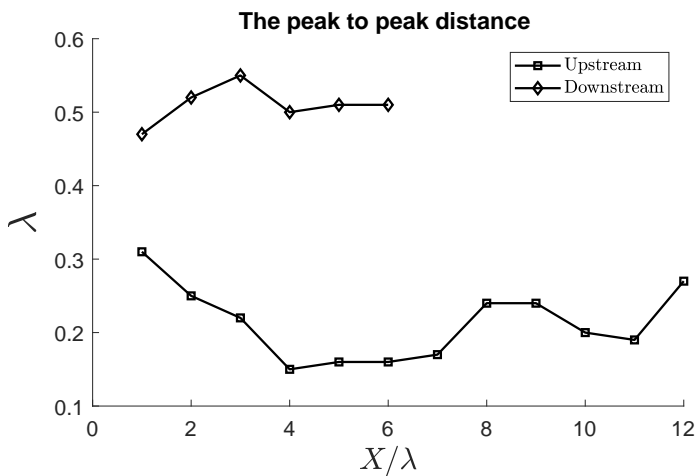
From Figure 4.13, it was clear that the downstream and upstream behaviour was generating very different types of waves. The wavelength was much longer downstream than upstream as expected. Downstream there only existed one wave, and the wavelength was quite uniform as can be witnessed in Figure 4.14. Upstream, the wave pattern was more complicated, as close to the cylinder the amplitude was large until it suddenly dropped significantly from the fifth to the sixth wavelength without any significant change in wavelength.

The mean peak to peak value downstream was  $\lambda_m = 0.51$  m, which compared quite well with the dispersion relation, which gave a value of  $\lambda = 0.5478$  m, giving a . Upstream, the dispersion relation gives a value for  $\lambda \approx 0.11$  m, while the peak to peak measurement gave a mean  $\lambda_m = 0.2133$  m and a lowest value at  $\lambda = 0.15$  m. This was further reflected in the frequency analysis shown in Figure 4.15a, where the dispersion relation was plotted alongside for comparison.

The results were even more clear, when plotted alongside the dispersion relation. The

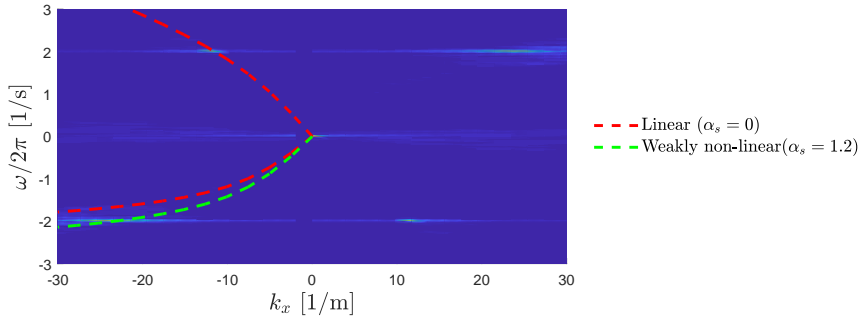


**Figure 4.13:** The surface height at two time intervals with equal phase

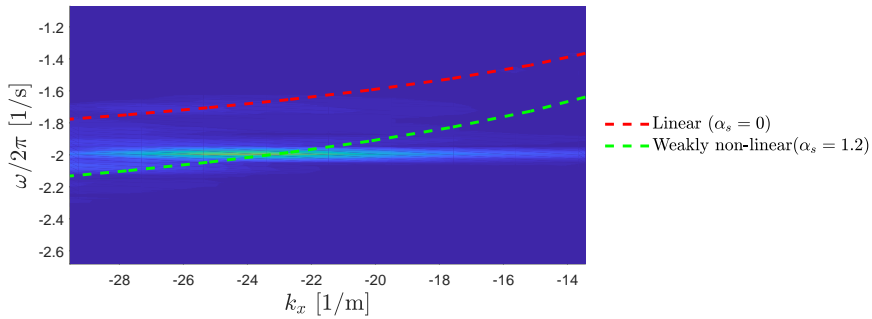


**Figure 4.14:** The peak to distance measured from the cylinder for each wavelength.

accuracy downstream were very good, while the solution was more smeared in the upstream direction, which was a direct consequence of that for the same frequency several wavelengths were measured. In addition, the incline of dispersion was at this point small, and therefore a small difference in frequency leads to a large difference in the wavenumber and wavelength. Therefore a weakly non-linear expansion of the dispersion relation was added to illustrate that the difference between the linear dispersion relation and the simulation might have been caused due to non-linearity. An expansion factor of  $\alpha_s = 1.2$ , is indicative of  $ka = 0.6325$ .



(a) The frequency analysis of the results from the uniform inflow simulation.



(b) A closer look on the dispersion upstream.

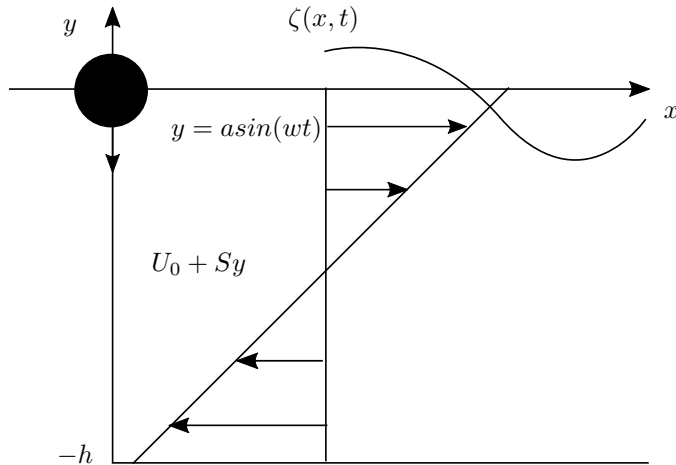
### 4.2.1 Discussion

One of the main issues was being able to keep a steady inlet and outlet condition so that the water level remained constant. Fixing the water level at the inlet was considered necessary, as this should create a steady boundary condition for the inflow of the water. However, unsteady behaviour was observed at one cell length into the mesh in terms of the phase fraction. This did appear to have correlation with damping parameter, as it was observed to react to changes in the parameter. During simulations the damping parameter close to the inlet was set to a low value, which was possible due the long incubation time of the

waves travelling against the main flow hitting the boundary. For the outflow condition, it helped greatly to switch from a zeroGradient to a variableHeightFlowRate condition. But still there were some water leakage, which is why from Figure 4.13 one can notice the waves are oscillating around a value, slightly less than the original still water level  $y = 0$ . As mentioned in the results, the downstream wavelength corresponded well with linear dispersion relation giving a relative error of 6.9%. The upstream effects were considered to be quite non-linear, which was clear from frequency plot, where a weakly non-linear correction to the dispersion relation gave good correspondence with measured data. The upstream data was necessarily imprecise to the very low gradient in the dispersion relation or low group velocity in other words. Another observation from Figure 4.13 was the sudden jump in amplitude between the fifth and sixth wavelength upstream of the cylinder, similar to a hydraulic jump. However, unlike an hydraulic jump the wave does not, increase in velocity, but rather the opposite, almost stagnates.

### 4.3 Oscillating cylinder - Linear shear flow

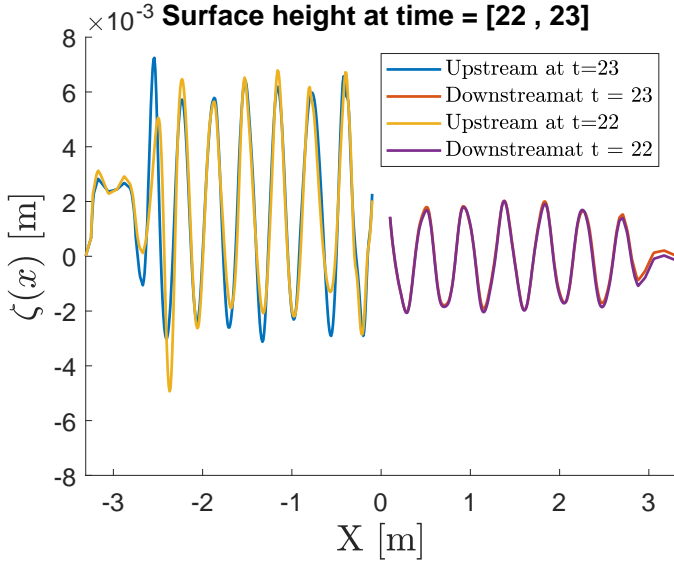
The objective of this case was to look into potential non-linear-effects caused by the resonance frequency of a moving source on linear shear current. This was done by superimposing the velocity of the oscillating cylinder,  $U$ , on a shear current. The problem description is shown in Figure 4.16.



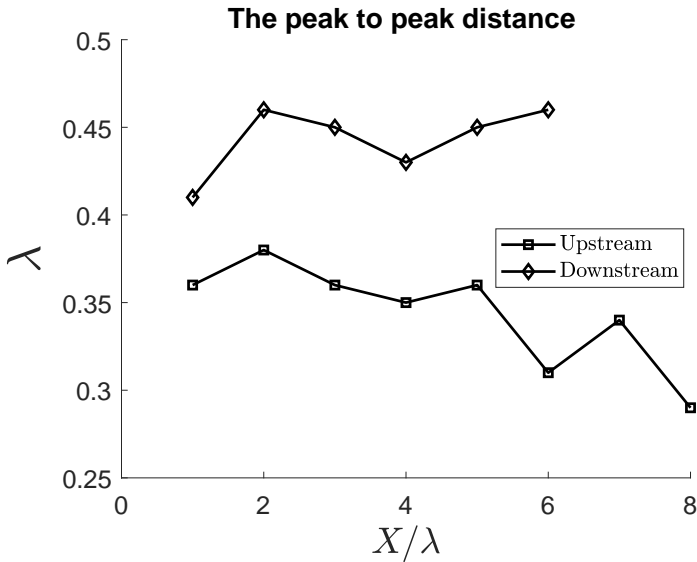
**Figure 4.16:** The problem description for the linear shear case.

The inlet velocity was  $U = 0.25$ , while the vorticity was  $S = 5.2$ ; Several changes had to be made to the boundary conditions. The velocity boundary was changed to a linear profile using the groovyBC library, while all the pressure condition containing fixedFlux-Pressure was changed as especially on patches it was incompatible with a shear current and any attempt to use it led to crashing the solver. Therefore the inlet and outlet boundaries were changed to inletOutlet to account for backflow setting any inlet values to zero, while the cylinder and bottom was changed to zeroGradient. The simulation pattern was

identical to that for the uniform flow case, where the cylinder is first placed in the flow until the transients were died out and then allowed oscillate on a coarse mesh and then mapped onto a finer mesh. The mesh was the same the one used in all three cases.



**Figure 4.17:** The surface height at the end of simulation time at the same phase.

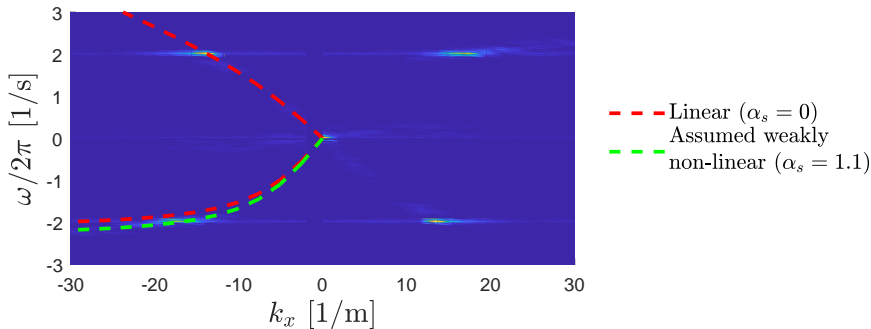


**Figure 4.18:** The peak to peak distance for the linear shear.

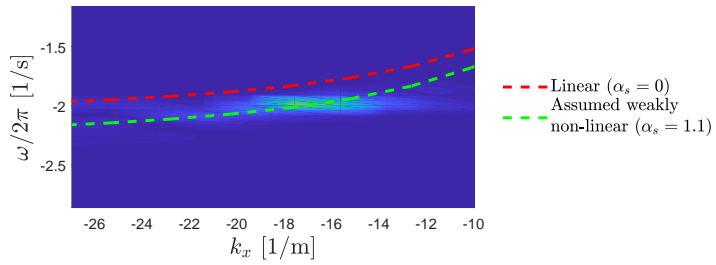
---

### 4.3.1 Discussion

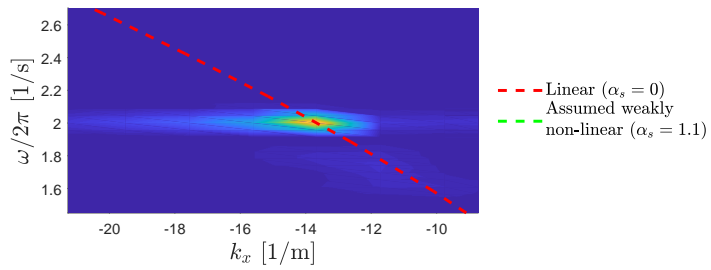
From Figure 4.17 and Figure 4.18, the conditions were quite unlike the moving cylinder case, the wavelength has increased significantly to a mean of  $\lambda_m = 0.3438$ , while maintaining a larger amplitude. This happened due to the introduction of vorticity, where the flow counteracts the rotation for the particles causing large amplitudes and larger wavelengths upstream. Downstream, it has the opposite effect, more vorticity caused the waves to shorten and the amplitude is lessened. This corresponds well with behaviour as seen in the frequency analysis plot in Figure 4.19. Upstream, the wave number from the frequency seemed to correlate better with an assumptive non-linear expansion. In the uniform flow case, this Stokes order assumption is valid for the dispersion relation, however to the author's knowledge there exist no extension for shear flow. This assumed behaviour was merely shown as possible explanation for the discrepancy between the simulation data and the linear dispersion relation. This assumption was based on that the third order Stokes expansion, which causes correction for the dispersion relation without shear flow exhibits the same behaviour with shear flow. The mean steepness of the waves was found to be  $s_m = k_m a_m = 0.06$ , which would also indicate a non-linear regime. Another typical sign of non-linear regime can be observed in Figure 4.17, where the linear wave downstream oscillates around a zero still water level, while upstream the wave oscillates around a level higher than the mean water level. The relative error in the wave length was in this case calculated as 3.74%.



**Figure 4.19:** The linear dispersion relation and an assumptive correction for weakly non-linear



(a) A closeup of the upstream



(b) A closeup of the downstream

**Figure 4.20:** Frequency analysis of the linear shear case



# Chapter 5

## Conclusion

### 5.1 Concluding remarks

Returning to the main objectives that was put forward in the master's contract. The developing and implementing of an appropriate boundary conditions able to absorb continuous wave spectrum's was achieved by implementing the quarterCosineRamp and tested in the damping layer case in chapter 3. Overall this exhibited a good behaviour being able to effectively dampen waves close to outlets. Some issues were discovered close to inlets such as in uniform and linear shear flow case. Oscillations due to a fixed value on the boundary, may require a deeper study, but it was manageable with trial and error to find appropriate values that gave stable conditions. Especially at inflow for the shear case, only low values  $f$  was permitted, therefore some reflections were unavoidable upstream of the cylinder for the linear shear flow case.

The 2D "ship wave" problem was studied through an oscillating cylinder moving at constant velocity. Two cases were shown, where the first case had no background flow and the second had shear assisted flow. The two cases gave both non-linear behaviour upstream and linear behaviour downstream. This was shown analytically by comparing the dispersion relation with a frequency analysis of the surface. Visually it was also found classical signs of non-linear behaviour such as steep troughs and crests, waves that risen from oscillating with equal height on each side of the mean still water level. For the uniform flow case, vortices was also observed crashing into the surface boundary causing wave breaking and overturning(see section 5.2). The linear shear profile was chosen as it is well treated in literature, and has a rather simple dispersion relation, but with the current set-up any velocity profile could be chosen. Though due to time restrictions no effort was made to look at other shear flows. The accuracy in the linear regimes were measured in terms of the wavelength, which had a relative error of 4% in the still water and 6% uniform case and 3.74% in the linear shear case for the linear part.

---

## 5.2 Further work

This thesis has shown that OpenFoam can be used to simulate surface waves in the linear regime with accuracy and has also showed that it can describe non-linearity, and therefore can be a useful tool in further research. Limits were set on the degree of accuracy as number of cells closer to 100,000 slowed the computation fairly done. One of the main reasons was due to the mesh movement, which had a solution time of up to one second per time step. This was the main limiting feature of this kind of simulations, therefore future work could be facilitated towards making the mesh movement more efficient, optimizing the mesh structure by increasing the resolution near the surface, without causing an increase in the global mesh size. Originally, an idea was to set the surface boundary as a moving mesh. However, the gain in not solving the air could easily be lost in effort of solving the mesh movement itself. Further research could immediately investigate more thoroughly the effects between shear flows and non-linear waves as seen in linear shear flow case. Parameter study could be done of the linear shear flow case, which also could work towards establishing how shear flow and non-linearity interact. Data from several simulation run at different frequencies could put together in the frequency analysis to give a broader picture, of the surface that was barely scratched in the thesis.

# Bibliography

- [1] T. Alexandru. *Data Visualization : Principles and Practice*. Taylor and Francis Group, second edition, 2015.
- [2] E. Berberović, N. Van Hinsberg, S. Jakirlić, I. Roisman, and C. Tropea. Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution. 79:036306, 04 2009.
- [3] J. P. Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(1):185–200, 1994.
- [4] T. Colonius and H. Ran. A super-grid-scale model for simulating compressible flow on unbounded domains. *J. Comput. Phys.*, 182(1):191–212, Oct. 2002.
- [5] J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Advances in Computational Mathematics*, 6(1):207–226, 1996.
- [6] G. Dagan and T. Miloh. Free-surface flow past oscillating singularities at resonant frequency. *Journal of Fluid Mechanics*, 120:139–154, 1982.
- [7] M. Demirdzic, I. Peric. Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries. *International Journal for Numerical Methods in Fluids*, 10:771–790, 1990.
- [8] S. S. Deshpande, L. Anumolu, and M. F. Trujillo. Evaluating the performance of the two-phase flow solver interfoam. *Computational Science and Discovery*, 5(1):014016, 2012.
- [9] O. Ditlevsen. Stochastic model for joint wave and wind loads on offshore structures. *Structural Safety*, 24(2):139 – 163, 2002.
- [10] J. Donea, A. Huerta, J. Ponthot, and A. Rodríguez-Ferran. *Arbitrary Lagrangian–Eulerian Methods*. Number 14. Encyclopedia of Computational Mechanics, 2004.
- [11] S. Ellingsen. Initial surface disturbance on a shear current, the cauchy-poisson problem with a twist. *Physics of Fluids*, 26:82–104, 2014.

- 
- [12] S. Ellingsen. Ship waves in the presence of uniform vorticity. *Journal of Fluid Mechanics*, 72(2), 2014.
- [13] S. Ellingsen and I. Brevik. How linear surface waves are affected by a current with constant vorticity. *European Journal of Physics*, 35(2), 2014.
- [14] S. Ellingsen and P. Tyvand. Oscillating line source in a shear flow with a free surface: critical layer-like contributions. *Journal of Fluid Mechanics*, 798:201–231, 2016.
- [15] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Mathematics of Computation*, 31(139):629–651, 1977.
- [16] O. Faltisen. *Sea Loads on Ships and Offshore Structures*. Cambridge University Press, 1993.
- [17] R. Finn. Capillary surface interfaces. *Notices of the AMS*, 46(7):770–781, August 1999.
- [18] N. Foster and R. Fedwick. Practical animation of liquids. *Siggraph 2001 Annual Conference*, pages 15–22, 2001.
- [19] J. Freund. Proposed inflow/outflow boundary condition for direct computation of aerodynamic sound. *AIAA Journal*, 35(4):740–743, 1997.
- [20] C. Greenshields. Openfoam v5 user guide, 2017.
- [21] J. Grue and E. Palm. Wave radiation and wave diffraction from a submerged body in uniform current. *Journal of Fluid Mechanics*, 151(1):257–278, 1985.
- [22] T. Hagstrom. Radiation boundary conditions for the numerical simulation of waves. *Acta numerica*, 8:47–106, 1999.
- [23] M. Haif and L. M. Chaudhry, editors. *Computer Modeling of Free-Surface and Pressurized Flows*. Springer-Science+Business Media, B.V., 2 edition, 1993.
- [24] R. L. Higdon. Initial-boundary value problems for linear hyperbolic system. *SIAM Review*, 28(2):177–217, 1986.
- [25] R. L. Higdon. Radiation boundary conditions for dispersive waves. *SIAM Journal of Numerical Analysis*, 31(1):64–100, 1994.
- [26] C. Hirth and B. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 2005.
- [27] F. Hu. A perfectly matched layer absorbing boundary condition for linearized euler equations with a non-uniform mean-flow. *Journal of Computational Physics*, (208):469–492, 2005.
- [28] F. Q. Hu, X. Li, and D. Lin. Absorbing boundary conditions for nonlinear euler and navier–stokes equations based on the perfectly matched layer technique. *Journal of Computational Physics*, 227(9):4398 – 4424, 2008.

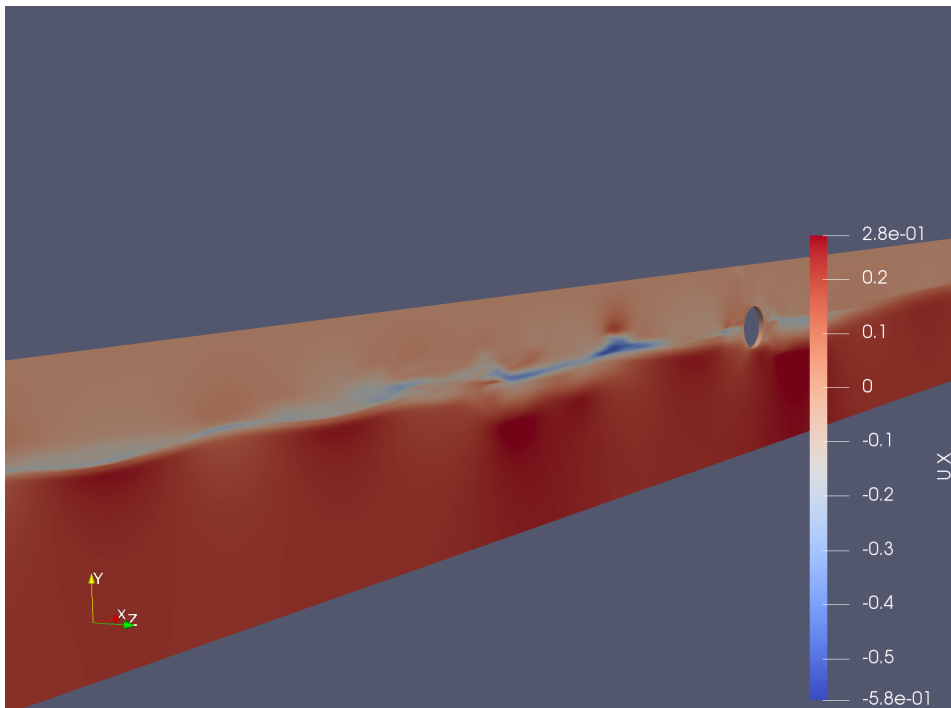
- 
- [29] M. Israeli and S. Orzag. Approximation of radiation boundary conditions. *Journal of Computational Physics*, (41):115–135, 1981.
- [30] ITTC. Recommended procedures and guidelines : Practical guidelines for ship CFD. Technical report, International Towing Tank Conference, 2011.
- [31] H. Jasak. Dynamic mesh handling in openfoam. 01 2009.
- [32] Z. Jasak, H. Tukovic. Automatic mesh motion for the unstructured finite volume method. *Transactions of FAMENA*, 30(2):1–18, 2007.
- [33] J. T. Kirby and T. Chen. Surface waves on vertically sheared flows: Approximate dispersion relations. *Journal of Geophysical Research: Oceans (1978–2012)*, 94(C1):1013–1027, 1 1989.
- [34] L. D. Landau and E. M. Lifshitz. *Fluid Mechanics*. Pergamon New York, 1987.
- [35] Y. Li and S. Ellingsen. Ship waves on uniform shear current at finite depth: wave resistance and critical velocity. *Journal of Fluid Mechanics*, 791:539–567, 2016.
- [36] X. Liu and M. H. García. Three-dimensional numerical model with free water surface and mesh deformation for local sediment scour. *Journal of Waterway, Port, Coastal and Ocean Engineering*, 134(4):203–217, 2008.
- [37] I. Orlanski. A simple boundary condition for unbounded hyperbolic flows. *Journal of computational physics*, 21(3):251–269, 1976.
- [38] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic implicit surfaces*. Springer-Verlaag, 2003.
- [39] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, pages 12–49, 1988.
- [40] S. V. Patanar. *Numerical Heat Transfer and Fluid Fluid Flow*. Hemisphere Publishing Corporation, 1980.
- [41] D. Peregrine. Interaction of water waves and currents. *Advances in Applied Mechanics*, 16:9–117, 1976.
- [42] M. Peric and J. H. Ferziger. *Computational methods for fluid dynamics*. Springer Berlin Heidelberg, 2002.
- [43] R. Perić and M. Abdel-Maksoud. Reliable damping of free surface waves in numerical simulations. *Ship Technology Research*, 63(1), 2016.
- [44] P. Reichl, K. Hourigan, and M. Thompson. Flow past a cylinder close to a free surface. *Journal of Fluid Mechanics*, 533:269–296, 2005.
- [45] H. Riebeek. Ship wave clouds over the north sea, 2011.

- 
- [46] J. Romate. Absorbing boundary conditions for free surface waves. *Journal of Computational Physics*, 99(1):135 – 145, 1992.
- [47] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, 2nd edition, 2003.
- [48] A. A. Saha, S. K. Mitra, M. Tweedie, S. Roy, and J. McLaughlin. Experimental and numerical investigation of capillary flow in su8 and pdms microchannels with integrated pillars. *Microfluid Nanofluid*, 2009.
- [49] B. K. Smeltzer and S. Ellingsen. Surface waves on currents with arbitrary vertical shear. *Physics of Fluids*, 29(4):047102, 2017.
- [50] G. G. Stokes. *On the Theory of Oscillatory Waves*, volume 1 of *Cambridge Library Collection - Mathematics*, page 197–229. Cambridge University Press, 2009.
- [51] M. Tyvand, P.A Lepperød. Oscillatory line source for water waves in shear flow. *Wave Motion*, 51:103–119, 2014.
- [52] B. van Leer. Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme. *Journal of Computational Physics*, 14(4):361 – 370, 1974.
- [53] R. Warming and R. M. Beam. Upwind second-order difference schemes and applications in aerodynamic flows. 14(9):1241–1249, 1976.
- [54] J. Wehausen and E. Laitone. *Fluid Dynamics/Strömungsmechanik*, volume 3 of *Encyclopedia of Physics/Handbuch der Physics*, chapter Surface Waves. Springer, Berlin, Heidelberg, 1960.
- [55] L. Yao, L. Y., and L. Li. A cell-based smoothed radial point interpolation—perfectly matched layer method for two-dimensional acoustic radiation problems. *ASME. J. Pressure Vessel Technol.*, 138(2), 2015.

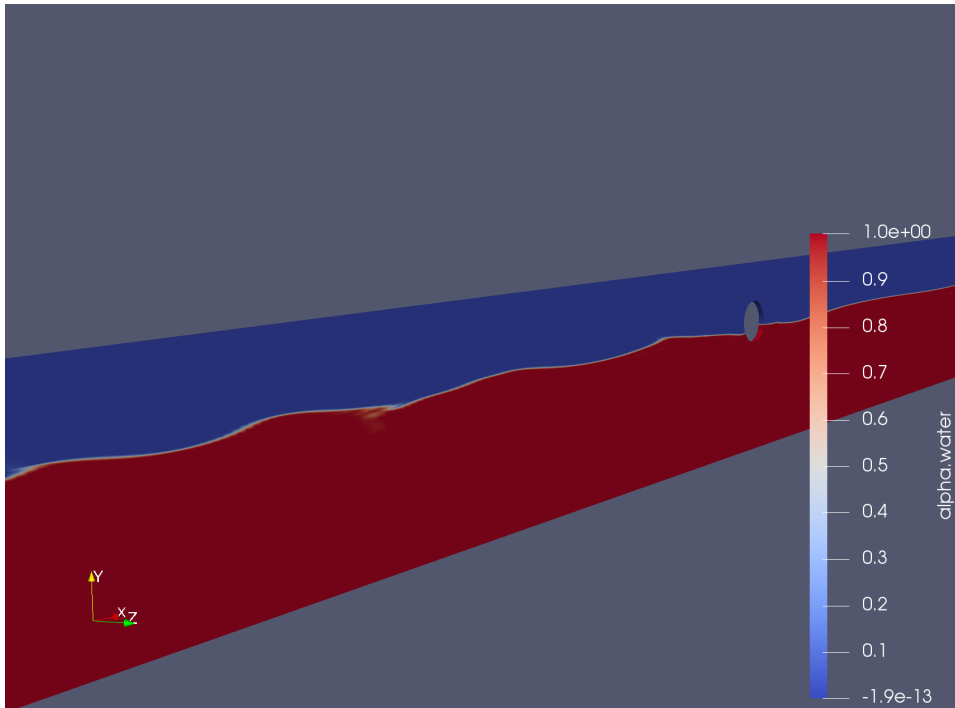
---

# Appendix

## Non-linear effects observed for the uniform flow case



**Figure A1:** Vortices were formed in front of the cylinder and trapped as the wave velocity and background flow were in opposite direction. Sometimes these vortices would collide into the surface causing wave overturning and breaking.

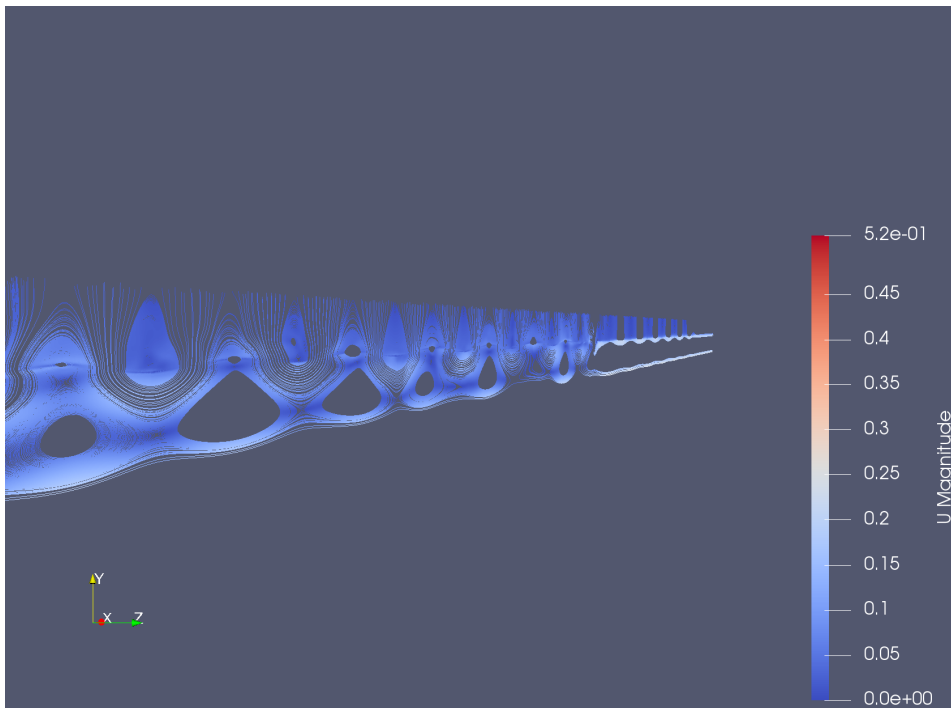


**Figure A2:** An example of vortex hitting the boundary causing wave breaking.

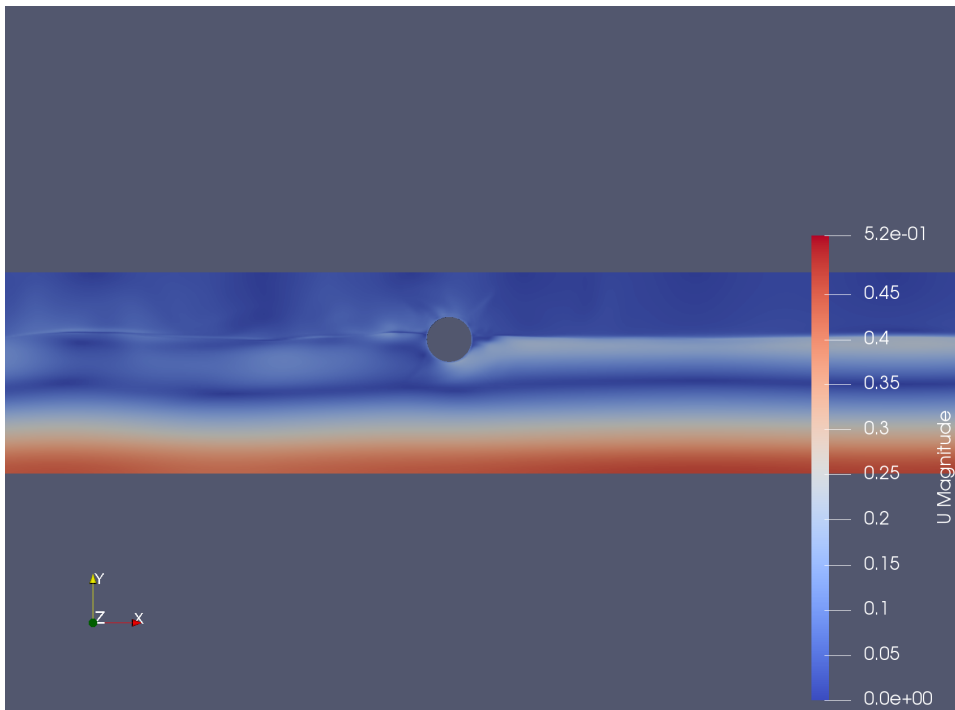


---

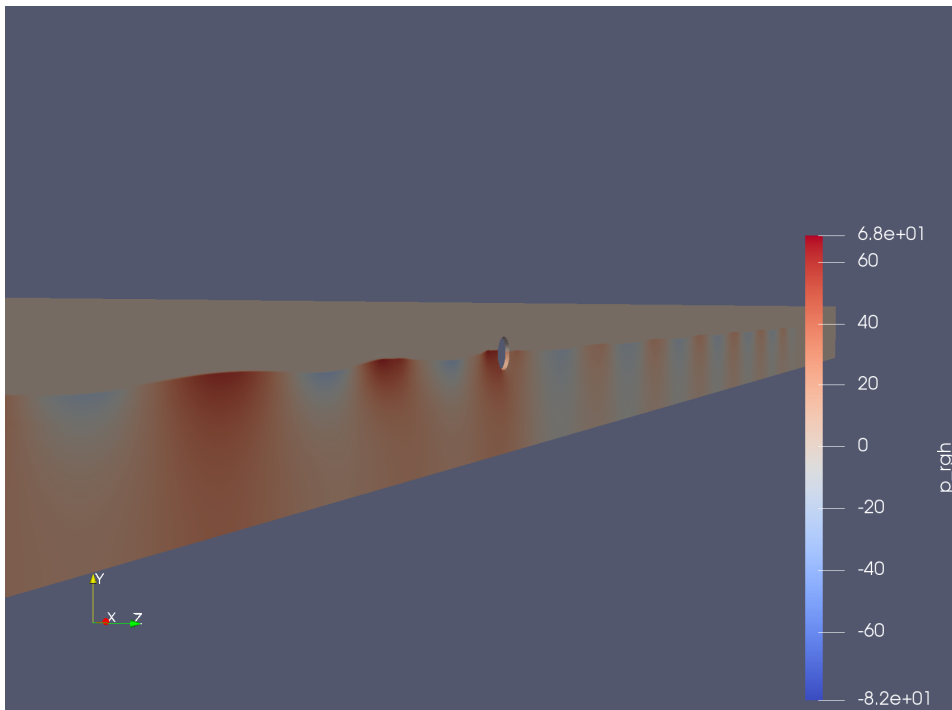
## Linear shear current plots



**Figure A3:** Streamfunction at  $t=23$  for linear shear current



**Figure A4:** The magnitude of the velocity vector  $t=23$  for linear shear current



**Figure A5:** The hydrostatic pressure at  $z=23$  for linear shear current