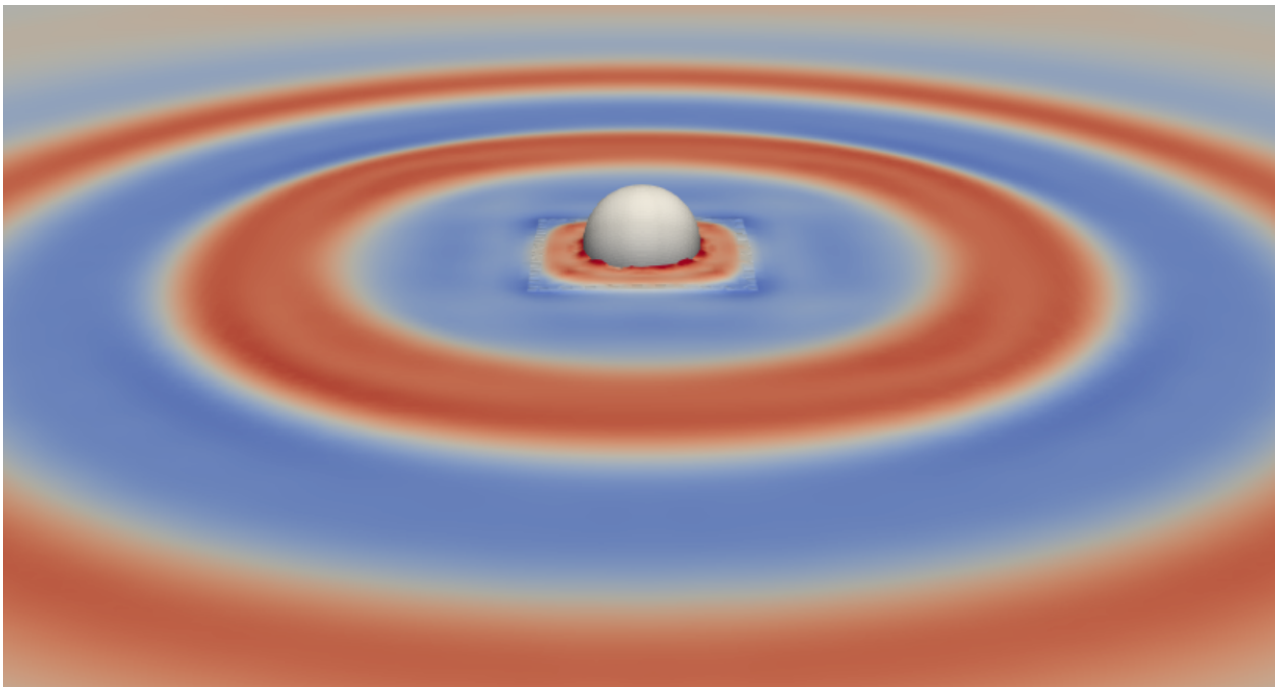
CHALMERS



Motion Decay of a Floating Object

Master's thesis in
Sustainable Energy Systems,
Innovative and Sustainable Chemical Engineering

MIKAEL BENGTSSON
MAX DELVRET

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

MASTER'S THESIS IN
SUSTAINABLE ENERGY SYSTEMS,
INNOVATIVE AND SUSTAINABLE CHEMICAL ENGINEERING

Motion Decay of a Floating Object

MIKAEL BENGTSSON
MAX DELVRET

Department of Mechanics and Maritime Sciences
Division of Marine Technology
CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2021

Motion Decay of a Floating Object
MIKAEL BENGTSSON
MAX DELVRET

Master's thesis 2021:09
Department of Mechanics and Maritime Sciences
Division of Marine Technology
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Laminar simulation of a free decay drop for a 0.15 m radius sphere after 3.2 seconds using an overset mesh technique.

Chalmers Reproservice
Gothenburg, Sweden 2021

Motion Decay of a Floating Object
Master's thesis in
Sustainable Energy Systems,
Innovative and Sustainable Chemical Engineering
MIKAEL BENGTSSON
MAX DELVRET
Department of Mechanics and Maritime Sciences
Division of Marine Technology
Chalmers University of Technology

Abstract

In today's growing demand for green energy the utilisation of ocean waves as an energy source is an attractive possibility. Computational fluid dynamics is an efficient and cheap way to test different design and environment conditions. Two different software are evaluated in this report in order to see possibilities and limitations of the different software. The software used in this project are FINE™/Marine and OpenFOAM.

Guidelines are then presented of how to set up an accurate and efficient simulation of a wave energy converter.

In this project three different meshing methods were tested, to see how efficient and accurate they are. The different meshing techniques were overset, sliding, and a deforming mesh technique, the different methods also required different case setups which is presented in this report.

A 0.15 meter radius sphere is dropped from 0.15 meter above the water surface and its motion is investigated for six seconds. The result for the different meshing techniques is then validated with experimental data. The overset mesh technique was determined to be the most accurate and stable method but it was the most computational heavy method. The overset mesh technique was tested with a k-ω-SST model to determine the effects of the turbulence. It was discovered that the turbulence was not significantly affecting the heave motion of the sphere.

The simulation case was then scaled up and a drop for a five meter radius sphere is simulated with its center of mass five meter above the resting water surface. This was done in order to show that the simulation method could be scaled up. The overset mesh technique was used for this case since it was determined to be the most accurate and stable method. The heave motion of the sphere was investigated for 40 seconds and the result was found to agree with numerical data from previous studies.

A numerical wave tank case with an overset mesh was setup in OpenFOAM in order to show the potential energy production of a wave energy converter. However further studies is needed for incoming waves since there doesn't exist any experimental data to validate with.

Keywords: OpenFOAM, FINE™/Marine, OEC, NWT, VOF, Overset mesh technique, Sliding mesh technique, Deforming mesh technique, RANS

Acknowledgements

This thesis was done at Chalmers University of Technology together with SSPA. We would like to start by expressing our gratitude to our supervisors at SSPA, Alex Abolfazl Shiri and Laura Marimon Giovannetti for guiding us in this project. Also we would like to thank Carl-Erik Janson for giving us the opportunity to work on this thesis. We would also like to thank our examiner Rickard Bensow for rewarding discussions. Furthermore we want to thank Chalmers Centre for Computational Science and Engineering for letting us use the data cluster Hebbe and SSPA for letting us use their computer cluster. Last but not least we want to thank the developers of the open source software OpenFOAM and Numeca for providing us with a licence for FINE™/Marine.

Nomenclature

Abbreviation

CFD	Computational Fluid Dynamics
LOA	Object Length Over All
NWT	Numerical Wave Tank
OEC	Ocean Energy Converters
OpenFOAM	Open-source Field Operation And Manipulation
PIMPLE	Pressure Implicit Method for Pressure Link Equations
RANS	Reynolds Averaged Navier-Stokes
SST	Shear Stress Transport
sixDoF	Six Degrees of Freedom
VOF	Volume Of Fluid

Dimensionless numbers

CFL	Courant-Friedrich-Lewy number
Pe	Peclet number
y^+	Normalized wall distance
μ_*	Friction velocity

Greek letters

α	Colour function
β	Specific model constants
Δ	Difference in quantity
δ	Kronecker delta
ε	Dissipation rate of the turbulent kinetic energy
λ	Wave length
μ	Dynamic viscosity
ν	Kinematic viscosity
ρ	Density
σ	Prandtl-Schmidt number
τ	Wave period
ϕ	Any given property
ω	Specific rate of dissipation of turbulent kinetic energy

Latin letters

A	Surface area
a	Acceleration
B	Specific model constants
C	Model constants
D	Mass transfer coefficient
d	Cell width
E	Energy

e	Unit vectors
F	Force
g	Gravitational constant
H	Wave height
h	Water depth
I	Moment of inertia
J	Translation along the axis
k	Turbulent kinetic energy
L	Length
l	Change in height over one time step
M	Molar mass
m	Mass
n	Normal vector
O	Momentum
P	Pressure
$\tilde{\mathbf{P}}$	Production term
Q	Rotation round corresponding axis
R	Ideal gas constant
r	Radius
S	Wave steepness
T	Temperature
t	Time
U	Mean velocity
u	Velocity
V	Volume
x	Distance
y	Distance from the wall
z	Molar fraction

Subscript

B	Buoyancy
d	Drag
i,j,k	Tensor indices
lam	Indicates laminar/viscous sub layer
n	Species
ref	Reference
T	Turbulent
w	Wave
wall	Wall
x,y,z	Space axis

Contents

1 Introduction

1.1 Background

Electricity production from coal-fired power plants has increased under the 20th century but has started to decrease some percent since 2007 [1]. Coal combustion has a lot of carbon dioxide emissions and if the emission reduction goals from the Paris agreement shall be reached, coal production needs to be reduced [2]. The awareness about today's environmental challenges is increasing around the world every day. Countries put more and more pressure on the companies to minimise their pollution and switch to greener technologies. The transition from fossil fuels to green energy with a constant growing energy demand is a very challenging task. To succeed with this transition new innovative developments and energy sources are needed to meet the growing energy demand. Today's large green energy producers are wind, solar and water power from rivers and lakes.

The ocean is also large potential energy source, today's commercial renewable energy sources wind and solar is mainly restricted to the land areas. Ocean energy converters (OEC) could be used to convert energy from the ocean's waves, currents, and tides. SSPA is one of the companies currently working with OEC and have a lot of research projects in this area [3]. Wave energy is one of the potential OEC that could become a large energy producer. The theory of the technology is that an object floating on the water surface can extract the energy from the heave motion generated by the waves [4]. The technology is still new and wave energy is under development, so more research and testing is needed to get a finished design.

Computational fluid dynamics (CFD) is an effective design approach to rapidly test different design ideas and operation environments. The potential use of CFD to simulate wave energy converters could be important for the technique to become commercial. This is because CFD could help to get a larger understanding of the heave motion of the object and the important forces acting on it. There exist many different numerical modelling techniques to simulate the heave motion generated by waves with different levels of accuracy and computational time. Previous study's where different numerical modelling methods were investigated has been done, showing that different methods give varying results. The reason behind this is that available CFD software are based on different assumptions and differences in the numerical codes. This can result in different results for the same cases, there is therefore a high interest in testing different methods when one is looking at a new problem [5].

1.2 Purpose

The purpose of this project was to accurately simulate the heave motion of a floating object in order to gain more knowledge about the possibilities and limitations with CFD simulations of wave energy converters. Also, guidelines for future simulation projects in this area was developed in this project. There is a high interest of knowing the difference between different available CFD programs when it comes to simulating the motion decay of floating objects, two different CFD programs are evaluated. The two CFD software used in this project were OpenFOAM and FINE^{TM}/Marine. The software were evaluated with respect to both accuracy and computational cost. When the simulations were considered accurate, they were evaluated further, the effect from turbulence was investigated to determine its effects on the heave motion. Heave motion in incoming waves was also simulated in order to investigate the possibilities to model the energy transfer from the waves to the object in both programs.

1.3 Delimitations

Some delimitations were implemented in this project to narrow down the scope of the project. It was an important aspect of the project to compare different numerical codes to gain more knowledge of how CFD could be useful for designing wave energy converters. This project was limited to two CFD software OpenFOAM and FINE^{TM}/Marine. OpenFOAM is a free open source CFD software which makes it easily available, and FINE^{TM}/Marine is commercial CFD software dedicated to marine applications.

A wave energy converter would need some parts connected to the floating object in order to be able to produce energy, this was not considered in the simulations since the final energy production design was outside the scope of this project.

The simulations consider only the heave motion of the sphere, since this movement was the most relevant for energy production. This was achieved by prohibiting motions in all other directions.

This project only considers a sphere as a design for a wave energy converter, as the focus of this project was mainly the simulation strategy and not the final design of the wave energy converter.

2 Theory

In this section the relevant theory for this project is presented. The chapter explains some of the features in both FINE™/Marine and OpenFOAM, as well as general CFD theory which is important for the project.

2.1 Governing equations

There are three governing equations to consider when working with fluid flow, the equation of continuity, equation of motion and the energy equation. For incompressible flows the energy equation doesn't need to be solved since there is no link between the energy equation and the equations of continuity and momentum for incompressible flows. The continuity and momentum equation for an incompressible Newtonian fluid is

$$\frac{\partial U_i}{\partial x_i} = 0 \tag{2.1}$$

and

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \nu \frac{\partial}{\partial x_j}\left(\frac{\partial U_j}{\partial x_i} + \frac{\partial U_i}{\partial x_j}\right) + g_i \tag{2.2}$$

respectively [6].

The equation of state can then be used to link pressure, density and temperature. For incompressible flows the ideal gas law

$$\rho = \frac{P}{RT\sum_n \frac{z_n}{M_n}}, \tag{2.3}$$

can be used if the pressure variation is low in the system, where z is the molar fraction [6].

2.2 Forces

To be able to describe a solid sphere's interaction with a water surface there are many forces to consider. Buoyancy is one of the important forces, this force arises for example when a body is submerged in a fluid. This force acts in the opposite direction of gravity, the buoyancy force is calculated by multiplying the fluid density with gravity and the volume of the submerged body according to

$$F_B = \rho g V. \tag{2.4}$$

When the body moves it's affected by frictional forces both from the air above the water surface and from the water directly interacting with the body, these forces are acting in the opposite direction of the movement of the body. If the Reynolds number of the sphere is small the frictional forces can be expressed with

$$F_d = 6\pi\mu r U. \tag{2.5}$$

If the Reynolds number is high, a correction factor is needed in equation 2.5 to be able to describe the viscous forces on the sphere accurately. In a CFD simulation with a high mesh resolution these forces described above are not modelled using these equations. The forces are calculated by integrating over the pressure to get the pressure forces and the viscous shear stresses to get the shear forces. This is done in each cell closest to the sphere and then calculating the force by multiplying the pressure with the area of the sphere within that particular cell. The forces calculated from each individual cell is summed up and a net force can be obtained. This is done in each time step during the simulation in order to describe the forces acting on the sphere as accurately as possible.

The motion of the body is determined by the forces acting on it, in order to calculate the total force F and momentum O the translational and angular acceleration is needed. The total force and momentum can then be calculated by the two equations,

$$\sum F = m\ a \qquad \sum O = I\ a \tag{2.6}$$

where I is moment of inertia [6].

2.3 Volume of Fluid

Volume of fluid (VOF) is a method for solving the interface between two fluids. It's achieved by the use of a colour function α to indicate the fraction of a certain fluid in each cell. Where $\alpha = 0$ indicates only fluid one and $\alpha = 1$ indicates fluid two. This means that if α is $0 < \alpha < 1$ then there is an interface present in the cell. This implies the transport properties in the governing equations can be written as [7]

$$\begin{aligned} \rho &= \langle \rho_1 \alpha + \rho_2 (1 - \alpha) \rangle \\ \mu &= \langle \mu_1 \alpha + \mu_2 (1 - \alpha) \rangle. \end{aligned} \tag{2.7}$$

2.4 Courant number

The courant number (CFL) can be used to get an appropriate time step for a simulation. For an explicit method the time step should be lower than the time it takes for the properties in the simulation to travel from one cell to a neighbouring cell. When using an implicit solver a larger time step can be used because the solution is iterated with the condition of several cells forward included. The CFL number can be calculated with

$$\Delta t = CFL \frac{\Delta x}{U}. \tag{2.8}$$

In general for an explicit solver the CFL number should be kept below one to ensure that the properties in the simulation only travels one cell per iteration. For an implicit solver the CFL number can be higher than one and the simulation will still be stable [6]. Many CFD software have the option to specify an adjustable time step in the simulation. A limit of a maximum value for the CFL number is specified in the beginning of the simulation, the software then adjust the time step for each iteration in the simulation to always keep the CFL number lower then the specified limit [8].

2.5 Wave theory

Sea waves behaves very different depending on a number of factors, the height of the wave, H, and wave period, τ, are two important parameters. These two parameters along with the gravity constant describe the steepness of the wave, the steepness of regular waves are given as

$$S = \frac{H}{g\tau^2}. \tag{2.9}$$

The water depth h is also an important parameter which affects the shape and behaviour of the wave [9]. These parameters described above decides which wave theory that is most suitable for steady regular waves which is shown in Figure 2.1.

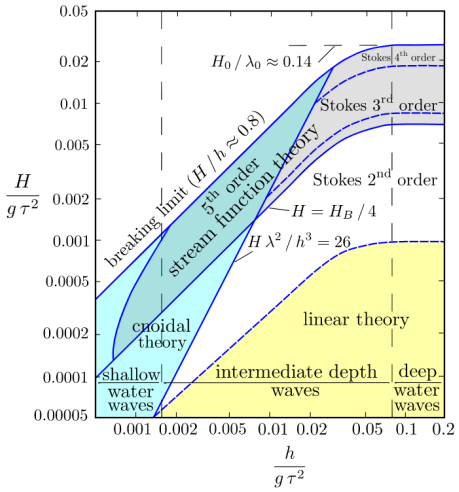


Figure 2.1: *Chart of different wave theories and where they are valid, adopted from [10] [11].*

Stokes 1^{st} order waves follow linear theory shown in the yellow area in Figure 2.1. The linear theory is used to describe smaller steady regular waves. Larger waves that is near the breaking limit is described by higher order Stokes theories. A linear wave theory describes symmetrical water waves that move along the water line. In higher order Stokes-theories the waves have peaks above the water line and troughs below the water line, this generates a more physical realistic wave. The wavelength of the wave can be calculated according to

$$\lambda = \tau U_w, \tag{2.10}$$

where U_w is the speed of the wave.

In reality the sea state on the ocean can seldomly be described with steady regular waves. The waves on the ocean is better represented with a large variety of regular waves [9].

When it comes to a domain size for wave simulations there are some general recommendations. These domain size recommendations are shown in Figure 2.2 in the x,z direction.

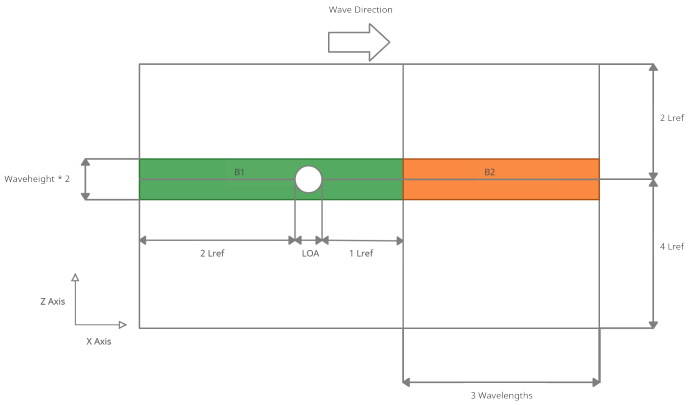


Figure 2.2: *Recommended wave domain in the x and z direction.*

In the figure L_{ref} is defined according to

$$L_{ref} = Max(\lambda, LOA), \tag{2.11}$$

and LOA is the length of the vessel/object.

In Figure 2.3 the general recommendations for the x,y direction of the domain is shown [12].

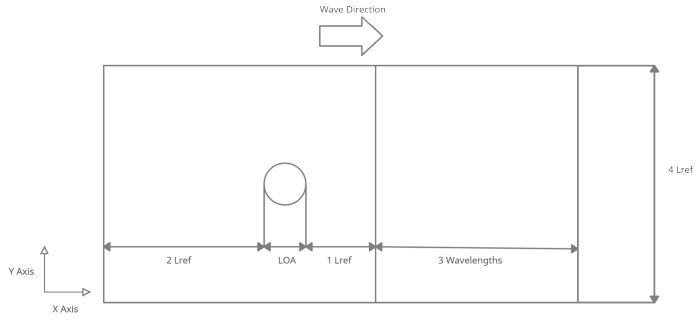


Figure 2.3: *Recommended wave domain in the x and y direction.*

The different zones given in Figure 2.2 also have some recommendations when it comes to the cell size. These recommendations are shown in Table 2.1, where d_x, d_y, d_z is the cell widths in x, y and z direction in the mesh. The non refined area is the entire area outside the zones B1 and B2 in Figure 2.2.

Table 2.1: Recommended cell size for the different wave domain regions.

Region	d_x	d_y	d_z
Non refined area	d_{ref}	d_{ref}	d_{ref}
B1	$\lambda/80$	$\lambda/20$	H/13
B2	d_{ref}	d_{ref}	H/13

In Table 2.1 d_{ref} is defined as [12]

$$d_{ref} = \frac{LOA * 2^8}{1000}. \tag{2.12}$$

2.6 Numerical models

There are three different numerical models which are commonly used to describe the heave motion of a wave energy converter, these numerical models are linear, weakly nonlinear, fully nonlinear. The methods are used to calculate the hydrodynamic forces which are affecting the floating object. The hydrodynamic forces are calculated with the first summation in equation 2.6.

The linear model uses a linear wave theory and assume that the amplitude of the radiating waves and the motion of the body are small in comparison with the wavelength of the wave. This is the simplest and quickest numerical model to use.

The weakly nonlinear model is similar to the linear model but the nonlinear effect of the buoyancy force is considered. These effects arise because of the instantaneous surface elevation during the motion body.

In the fully nonlinear models the water interface is tracked over time. The hydrodynamic forces are then calculated by integrating the total pressure and the viscous shear stresses over the body surface. The numerical models in CFD codes are fully nonlinear, where the interface can be tracked with for example the VOF model [13].

2.7 RANS

Reynolds have introduced statistical averaging methods for turbulent flows, these statistical methods are used to obtain mean values of the properties in the flow over time. This simplifies the information in the flow which also simplifies the description of the flow. With Reynolds averaged Navier-Stokes (RANS) the flow is described by the mean velocity of the flow and the turbulent properties. The Navier-Stokes equations for incompressible flows are given as

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_j} + \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j}. \tag{2.13}$$

The Navier-Stokes equations can be rearranged and by applying time averaging over the equations to obtain the RANS equation

$$\frac{\partial\langle U_i\rangle}{\partial t} + \langle U_j\rangle\frac{\partial\langle U_i\rangle}{\partial x_j} = -\frac{1}{\rho}\frac{\partial}{\partial x_j}\bigg(\langle P\rangle\delta_{ij} + \mu\big(\frac{\partial\langle U_i\rangle}{\partial x_j} + \frac{\partial\langle U_j\rangle}{\partial x_i}\big) - \rho\langle u_i u_j\rangle\bigg). \tag{2.14}$$

The last term in equation 2.14 represents the Reynolds stresses [6].

2.7.1 The k-ε model

The k-ε model is a two-equation RANS model, it is very popular due to its accurate and robust description of the energy dissipation rate, ε, and that ε occurs directly in the transport equation for the turbulent kinetic energy k. The modeled equation for k is given as

$$\frac{\partial k}{\partial t} + \langle U_j\rangle + \frac{\partial k}{\partial x_j} = \nu_T\bigg(\big(\frac{\partial\langle U_i\rangle}{\partial x_j} + \frac{\partial\langle U_j\rangle}{\partial x_i}\big)\frac{\partial\langle U_i\rangle}{\partial x_j}\bigg) - \varepsilon + \frac{\partial}{\partial x_j}\bigg(\big(\nu + \frac{\nu_T}{\sigma_k}\big)\frac{\partial k}{\partial x_j}\bigg). \tag{2.15}$$

ε is a measurement of the dissipation rate of the turbulent kinetic energy. The equation for ε is given as

$$\frac{\partial\varepsilon}{\partial t} + \langle U_j\rangle\frac{\partial\varepsilon}{\partial x_j} = C_{\varepsilon l} + \nu_T\frac{\varepsilon}{k}\bigg(\big(\frac{\partial\langle U_i\rangle}{\partial x_j} + \frac{\partial\langle U_j\rangle}{\partial x_i}\big)\frac{\partial\langle U_i\rangle}{\partial x_j}\bigg) - C_{\varepsilon 2}\frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j}\bigg(\big(\nu + \frac{\nu_T}{\sigma_\varepsilon}\big)\frac{\partial\varepsilon}{\partial x_j}\bigg), \tag{2.16}$$

where C are empirical closure constants. The production of the turbulent kinetic energy is given as

$$-\langle u_i u_j\rangle\frac{\partial\langle U_i\rangle}{\partial x_j} = \nu_T\bigg(\frac{\partial\langle U_i\rangle}{\partial x_j} + \frac{\partial\langle U_j\rangle}{\partial x_i}\bigg)\frac{\partial\langle U_i\rangle}{\partial x_j} - \frac{2}{3}k\frac{\partial\langle U_i\rangle}{\partial x_i}. \tag{2.17}$$

The k-ε model is very robust and accurate for a wide variation of turbulent flows. The model cannot resolve the flow close to the walls and therefore requires wall functions (see section 2.7.4) or low Reynolds number functions to model the boundary layer [6].

2.7.2 The k-ω model

k-ω is another two-equation RANS model. The quantity ω is defined as ε/k, which is the inverse of the time scale for which dissipation occurs. The modeled equation for k are

$$\frac{\partial k}{\partial t} + \langle U_j\rangle + \frac{\partial k}{\partial x_j} = \nu_T\bigg(\big(\frac{\partial\langle U_i\rangle}{\partial x_j} + \frac{\partial\langle U_j\rangle}{\partial x_i}\big)\frac{\partial\langle U_i\rangle}{\partial x_j}\bigg) - \beta k\omega + \frac{\partial}{\partial x_j}\bigg(\big(\nu + \frac{\nu_T}{\sigma_k}\big)\frac{\partial k}{\partial x_j}\bigg). \tag{2.18}$$

The modeled equation for ω is given as

$$\frac{\partial\omega}{\partial t} + \langle U_j\rangle\frac{\partial\omega}{\partial x_j} = B\frac{\omega}{k}\nu_T\bigg[\bigg(\frac{\partial\langle U_i\rangle}{\partial x_j} + \frac{\partial\langle U_j\rangle}{\partial x_i}\bigg)\frac{\partial\langle U_i\rangle}{\partial x_j}\bigg] - \beta^*\omega^2 + \frac{\partial}{\partial x_j}\bigg[\bigg(\nu + \frac{\nu_T}{\sigma_\omega}\bigg)\frac{\partial\omega}{\partial x_j}\bigg]. \tag{2.19}$$

The k-ω model performs very well in regions with low turbulence where both k and ε approaches zero, this is where the k-ε model has a bad performance. The k-ω model also outperforms the k-ε model in regions close to the walls where the model can describe the viscous sub layers. This eliminates the need for wall functions for the walls which comes to the price of a higher computational time [6].

2.7.3 k-ω SST model

k-ω Shear Stress Transport (SST) model is a hybrid model of the k-ε and k-ω turbulence models. This model combine the accurate and robustness in the near wall region of the k-ω model while avoiding the free stream problems by switching to the k-ε model in this region. This is achieved by combining these models using blending functions. The final modelled equation

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho U_i k)}{\partial x_i} = \tilde{\mathbf{P}}_k - \beta^*\rho k\omega + \frac{\partial}{\partial x_i}\big[(\mu + \sigma_k\mu_f)\frac{\partial k}{\partial x_i}\big] \tag{2.20}$$

describes the turbulence [14].

2.7.4 Wall functions

Many flow situations involves flows which are restricted by walls, such as pipe flows. The viscous forces in the flow changes close to the wall which will develop a boundary layer at the wall. Some turbulence models such as the k-ε model are not valid in the region where the viscous effects are high. The gradients of the flow properties is large in this region which means that a fine grid needs to be implemented in order to resolve the flow in this region, which is computationally expensive. One approach to solve this problem is to not resolve the viscous sub layer but instead use wall functions. Wall functions calculates the mean velocity and the turbulent quantities far from the wall which also means a coarser grid can be used close to the wall [6].

y^+ is a non dimensional number which is a normalized wall distance from the wall. y^+ is calculated with

$$y^+ = \frac{\mu_* y}{\nu}, \tag{2.21}$$

where y is the distance from the wall in meters.

μ_* is the frictional velocity and is calculated according to

$$\mu_* = \sqrt{\frac{\tau_{wall}}{\rho}}, \tag{2.22}$$

where τ_{wall} is the shear stresses. A general guideline is that the wall functions are valid for y^+ values between 30 to 100. Below 30 is the viscous sub layer which cant be resolved with wall functions, instead a finer grid is needed to fully resolve the flow. The k-ω SST model and the k-ω turbulence model can resolve the viscous sub layer with a fine grid but wall functions can also be used if the viscous sub layer is not of interest [6].

2.8 OpenFOAM

OpenFOAM is a free open source CFD toolbox that contains tools for the setup of CFD cases, numerical solvers to solve the cases, and post processing tools for evaluating the results [8]. This leaves a large freedom to the user and the possibility for the user to develop their own solvers or boundary conditions. OpenFOAM does not have a user interface but instead rely on text files for inputs from the users with an important folder structure, an example of this can be seen in Figure 2.4. Here is an overset mesh case where the static mesh is in the background folder and the moving mesh is located in the floating body folder.

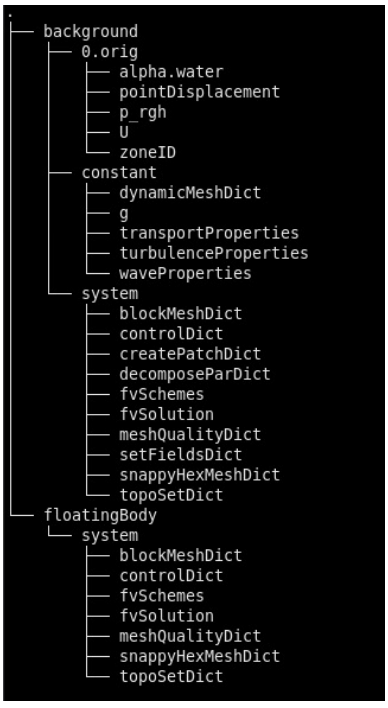


Figure 2.4: *Folder structure in OpenFOAM for the case of an overset mesh.*

The tools used in the OpenFOAM package are *blockMesh*, *snappyHexMesh*, *mergeMeshes*, *topoSet*, *setFields*, *overInterDyMFoam*, *interFoam* and *paraView*.

blockMesh is an mesh generating utility supplied in the OpenFOAM package, it uses the supplied instructions in the blockMeshDict to generate a mesh. This works for generating a simple mesh, for using a more complex mesh it is recommended to use snappyHexMesh in combination with *blockMesh*.

The utility *snappyHexMesh* is a utility that takes the mesh that is generated by the *blockMesh* and inserts geometries. The geometries that can be generated can either be simple like boxes and spheres, or more complex geometries that are user supplied with a stl file (or similar). It also allows the refinement of specific areas. All this give greater freedom when generating the mesh.

The utility *topoSet* allows the assignment of specific topology to be able to perform operations on these specific points.

The utility *setFields* allows the user to give specific cells a certain initial condition. For example this is used to set the initial α value for the Volume of Fluid calculation.

In the OpenFOAM package there are many different solvers, this project focuses on the use of the solvers *overInterDyMFoam* and *interFoam*. This is because these solvers assumes incompressible, isothermal and immiscible fluids and solves the use of multiple fluids by use of the Volume of Fluid method. Both also support of deforming meshes, with the only difference that *overInterDyMFoam* also supports the overset mesh method [15].

2.8.1 dynamicMeshDict

The dynamicMeshDict folder in OpenFOAM is used too specify how the motion of a solid object is solved for. The sixDoFRigidBodyMotion solver in OpenFOAM is specified in the dynamicMeshDict and used to solve the translational and rotational motions of an object, this solver has six degrees of freedom. Here it can be specified if the object is allowed to rotate or move and in which directions. This is very useful if the user wants to limit the motion to one direction.

A scheme is also needed to be specified in order to solve the equation for the motion. There are three available scheme options Newmark, CrankNicolson, symplectic. The Newmark method is a second order implicit method, CrankNicolson is a second order implicit method that is unconditionally stable, sympletic is a second order explicit solver [15].

2.8.2 Numerical schemes

The fvSchemes dictionary in OpenFOAM lets the user specify which numerical schemes that should be used to solve for the different properties in the simulation. There is a wide variation of schemes to chose from in OpenFOAM which gives a large degree of freedom for the user, the schemes are specified in the sub-dictionary fvSchemes. In this chapter some frequently used numerical schemes in OpenFOAM are presented. The schemes used for the different simulation setups in this project are specified in the method section.

Time schemes

Transient simulation problems require discretisation of the governing equations in time. This is done in OpenFOAM with numerical time schemes, the time schemes defines how the properties are integrated in time.

Euler is a first order implicit, bounded spatial discretisation scheme. The Euler scheme has shown to be very stable, quick and very accurate with the condition that the courant number is kept under one to minimize the diffusion [16]. The Euler scheme iterates according to

$$\frac{\partial}{\partial t}\phi = \frac{\phi - \phi^0}{\Delta t} \tag{2.23}$$

where ϕ represent any given property and ϕ^0 is the property in the previous time step.

Crank-Nicolson time scheme is a second order, transient bounded time scheme. This time scheme also requires the specification of a coefficient between 0 and 1, where a 0 would represent pure Euler and 1 pure Crank-Nicolson. A value of 0.9 is recommended to get a good mix between accuracy and robustness. A pure Crank-Nicolson time step discretisation is given as

$$\frac{\partial}{\partial t}\phi = \frac{\phi - \phi^{00}}{2\Delta t}, \tag{2.24}$$

where ϕ^{00} represents the property two time steps back [15].

Gradient scheme

The gradient of a scalar property ϕ can be expressed as

$$\Delta\phi = e_1\frac{\partial}{\partial x_1} + e_2\frac{\partial}{\partial x_2} + e_3\frac{\partial}{\partial x_3}, \tag{2.25}$$

where the e_i vectors are representing the unit vectors for a space in 3D.

Gauss linear is a gradient scheme available in OpenFOAM. It uses Gauss theorem to calculate the cell gradients, Gauss theorem is given as

$$\int_V (\Delta\phi)dV = \oint_A (n\phi)dA, \tag{2.26}$$

where A is the surface area of the cell. Gauss linear is very accurate on hexahedral meshes but can lose accuracy with high skewness of the cells [17].

Divergence schemes

The divergence terms require evaluation of ϕ, the volumetric flux. The treatment of these terms are very challenging and there are therefore a large variety of divergence schemes to choose from. All the divergence schemes are based on Gauss integration, some of the commonly used divergence schemes are specified in Table 2.2. vanLeer is limited scheme which is commonly used for variables that are limited between certain values e.g α which is bounded between zero and one [15].

Table 2.2: Commonly used divergence schemes available in OpenFOAM.

Gauss (interpolationscheme)	Second order
linear	Second order unbounded
linearUpwind	Second order unbounded
LUST	blended, 75 % linear, 25 % linearupwind
upwind	First order bounded
vanLeer	Second order unbounded, limited scheme

Laplacian schemes

The Laplacian schemes are specified in order to determine the Laplacian terms, for example the diffusion term in the momentum equation given in Equation 2.2. The only available scheme for discretisation is the Gauss scheme which also requires specification of an interpolation scheme and an surface normal gradient scheme. The choice of surface normal gradient scheme depends on the mesh, generally the schemes uncorrected and orthogonal is used for meshes with an non-orthogonality of maximum 5°. The corrected scheme is generally used for meshes with an non-orthogonality of maximum 70° [15].

2.8.3 Pressure-velocity coupling

To solve the Naiver-Stokes set of equations pressure and velocity are needed to be solved simultaneous since the velocity is present in both the momentum and continuity equation. A pressure-velocity algorithm is used to solve this problem, Pressure Implicit Method for Pressure Link Equations (PIMPLE) can be used for this purpose in OpenFOAM. PIMPLE starts with the pressure from the previous time step or the initial guess, the pressure is used to solve the momentum equation. The velocity field is then solved for and then a corrected pressure is calculated from the velocity field. The number of inner loops (nCorrectors) can be specified in the PIMPLE algorithm, the algorithm then starts over the iteration from the velocity field. One can also specify (nOuterCorrectors) the number of outer loops, the iteration then iterations from the momentum equation again [9].

2.8.4 Boundary conditions

In this section are different boundary conditions available in OpenFOAM presented, a short physical explanation is given for each boundary condition. The boundary conditions used for the different simulation cases is given in the method section.

zeroGradient

zeroGradient is available for all quantities as a boundary condition where the normal gradient of the defined quantity is constant, which means that the gradient of the quantity is zero at the boundary [15].

fixedFluxPressure

The fixedFluxPressure gradient adjust the pressure gradient so it meets the flux value specified at that boundary by the velocity boundary condition[15].

movingWallVelocity

movingWallVelocity is a boundary condition for the velocity which applies the velocity calculated in the dynamicMeshDict to the specified boundary. This boundary condition is necessary in OpenFOAM when dealing with moving wall boundaries [15].

fixedValue

The fixedValue boundary condition specifies a value at the boundary, and the gradients of the specified quantity is then adjusted during the simulation to meet the specified value at the boundary [15].

pressureInletOutletVelocity

pressureInletOutletVelocity is a velocity boundary condition which specifies the pressure at the boundary. The outflow velocity is given a zeroGradient boundary condition and the inflow velocity is specified from the calculated internal cell value at the boundary in the simulation. This can for example be used when the boundary is assumed to be open to the atmosphere [15].

inletOutlet

The inletOutlet lets the user specify the inflow value at the boundary for the given quantity, the outflow for the quantity is calculated in the simulation [15]. This boundary condition is very useful for simulations when the inflow of a certain quantity is known.

kqRWallfunction

This wall function is calculating the turbulent kinetic energy k at the wall as an boundary condition for the turbulent model [15].

omegaWallfunction

This wall function calculates the turbulent dissipation rate ω at the wall. It take into consideration the y^+ in the calculation so that it varies the equation used depending on which sub-layer the cell is located according to

$$\begin{aligned} \omega &= \frac{6\nu_\omega}{\beta_1 y^2} \quad if \quad y^+ \leq y^+_{lam} \\ \omega &= \frac{\sqrt{k}}{C_\mu \kappa y} \quad if \quad y^+ > y^+_{lam}. \end{aligned} \tag{2.27}$$

This means that y^+ can vary more at the surface and this means that the restraints on the mesh is less strict [15].

2.8.5 Wave creation

To be able to accurately create a numerical wave tank (NWT) there is a need to accurately create and absorb waves, fortunately the interFoam solvers include toolboxes precisely for this purpose. These tools can create waves according to multiple wave models such as Boussinesq and Grimshaw for solitary wave theories, and canodical and Stokes for regular wave theories [18]. It also allows for absorption of waves according to the shallowWaterAbsorption model [15]. These tools allow for many changes to the variables for the wave generation such as wave height, ramp time and wave period, this allows for high flexibility for the use in NWT.

When using these models the only thing that needs to be changed are the patch type for the inlet and outlet to patch instead of wall, to change the velocity boundary condition to *waveVelocity* and for the colour function boundary condition change to *waveAlpha* for the inlet [12].

2.9 FINE™/Marine

FINE™/Marine is a CFD software that is specialized for marine applications. The software is customized for single-fluid and multi-fluid flows around ships and boats. FINE™/Marine can resolve free floating surfaces and calculate motions of ships with a marine dedicated six degree of freedom (sixDoF) motion solver. The graphical user interface is also customized for marine applications with user features dedicated for marine engineering [19].

2.9.1 Numerical schemes

In this section some of the available numerical schemes in FINE™/Marine are presented. The specific numerical schemes used in the different simulation cases are specified in the method section.

AVLSMART

AVLSMART is a higher order bounded scheme which is available for the discretisation of convective fluxes in both the momentum equation and in the turbulence modeling in FINE™/Marine. The AVLSMART scheme uses a third order quadratic upwind (QUICK) as a base scheme. It has been shown that the AVLSMART scheme improves convergence in many cases without losing accuracy [12].

HYBRID

The HYBRID scheme is a combination of central-differencing scheme and first-order upwind scheme. It uses both the numerical accuracy of the central-differencing scheme and the boundedness and transitiveness in the first-order upwind scheme, which scheme it uses depends on the Peclet number. The central-differencing scheme is referred to as second order accurate. The Peclet number is calculated with the ratio between the convective mass transfer and the diffusive mass transfer, the ratio for the Peclet number is

$$Pe = \frac{U\Delta x}{D}, \tag{2.28}$$

where D is the mass transfer coefficient [6].

If the criterion

$$|Pe| > 2 \tag{2.29}$$

is fulfilled the HYBRID scheme uses the first-order upwind scheme, otherwise it uses the central differencing scheme [6].

2.9.2 Body Motion

It is possible to have six different types of motions in FINE™/Marine these are noted (J_x, J_y, J_z) and (Q_x, Q_y, Q_z) where J is the translation and Q is the rotation along the corresponding axis. This means that an object has six degrees of freedom (sixDoF), each of these can then be either fixed, imposed or calculated. If a DoF is fixed this means that there will be no movement in that DoF relatively to the given frame of reference. If the motion is imposed this means that the movement is supplied by the user, this can be done many different ways in FINE™/Marine. Then the DoF can be set to calculated, this means that motion will be solved for in the

solution by the use of Newton's law. This calculation rely on supplied properties of the object, such as weight and inertia [12].

2.9.3 Boundary conditions

In this section different boundary conditions available in FINE™/Marine are presented, a short physical explanation is given to each boundary condition. The specific boundary conditions for the different simulation setups are specified in the method section.

Slip Wall

This boundary condition is for a wall where the velocity can be different from zero and the turbulence due to shear stress is neglected. [12]

No Slip Wall

This boundary condition sets the velocity of the flow to zero relative to the wall at the boundary [12].

Wall with Wall-Function

This boundary condition applies a wallfunction to the solid boundary [12].

Far Field with Constant Values

This boundary condition allows the use of prescribed velocities, mass fraction and turbulence. These can either be a default or entered constant value [12].

Prescribed Pressure

This is a Dirichlet boundary condition, the pressure is given at the start of the simulation. This is recommended at the top and bottom patch for a 3D case with multi-fluid flow [12].

Overset

This boundary condition have no impact on the physics of the flow but it is an indicator to the solver where to perform the interpolation between the meshes [12].

2.9.4 Wave creation

FINE™/Marine also contains tools for the creation of a NWT. This is achieved by changing the boundary conditions and the size of the domain.

To generate waves the boundary condition *Wave Generator* is used, this boundary condition generates waves in by the use of a source term in the Navier-Stokes momentum equation. The waves are created using the Stokes wave theory. The properties that are needed to be specified for the wave generation are the water depth, wave period, the order of the Stokes wave theory and which the direction the wave is traveling in.

When generating waves for testing it is important to dampen the waves as well to make sure that there occur no reflections of the waves that can effect the results. This is achieved by the use of the utility named *Wave Damping*. This inserts a sponge dampening zone, this zone then uses Darcy's law to dampen the velocities in the z-direction [12].

2.10 Overset Mesh Technique

The overset mesh method is available in both FINE™/Marine and OpenFOAM, this method is useful when the simulation involves solid objects with moving boundaries. The overset mesh can handle large motion while keeping a good mesh quality. The method can be implemented to many different cases with moving boundary conditions and still keep a good mesh, although the method comes with a high computational cost.

In the overset mesh method, the idea is to use two or more meshes that overlap. The different meshes are divided into two categories. One background mesh that is static and describes the whole computational domain,

while the others are overset meshes that are connected to an object or boundary that can move. To be able to use the overset mesh method there is a need for a strategy to connect and solve the numerical calculations between the meshes in each iteration. The meshes needs to communicate information between the cells without any numerical errors or instabilities.

Suppose that for a 2D problem there are a background mesh and a foreground mesh with an object in the center. To be able to calculate a solution the cells need to be scanned and divided into categories. In Figure 2.5 this is shown, here the fringe cells can be seen in both the background and the foreground mesh. There are also some cells that are unused, this can be seen as there is no background mesh inside the back fringe cells. The back fringe cells are located where the boundary for the background mesh and the front fringe cells are located at the boundary for the foreground mesh.

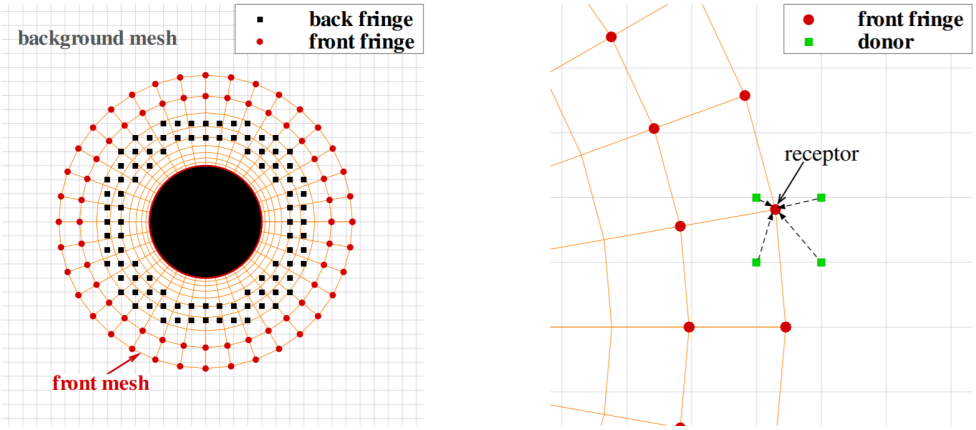


Figure 2.5: *Cell division in overset mesh, retrieved from [20]*

To be able to solve the differential equations the mesh needs to have values of the properties in the fringe cells as these are boundaries. The property value is calculated by interpolation of the donor cells corresponding to the specific fringe cell in the opposite mesh.

In OpenFOAM the interpolation method used is the *cellVolumeWeight* as it is a conservative method with higher accuracy at the cost of higher computational cost [21].

FINE™/Marine uses two different interpolation techniques, the first one is a so called *least squared approach* that is based of linear polynomials. This have a formal second order accuracy, but it suffers from numerical instabilities. To work around this a second distance weighted interpolation is also used. The choice of interpolation method is based on how good the connectivity between the fringe and donor cell are [12].

When it comes to the mesh generation it is important to take into account the mesh size at the overlapping areas, this because if these vary to much in size then the interpolation will not be as good as when they are of similar size [22].

2.11 Deforming Mesh Technique

The deforming mesh method is available in both FINE™/Marine and OpenFOAM, it is an extension of the static mesh where one implements deforming boundaries in the mesh. The deforming mesh has a defined number of cells and the number of cells remain unchanged during the entire simulation. A boundary is specified which has a movement in the simulation, in OpenFOAM an inner and an outer sphere in the mesh is defined. The mesh is then allowed to deform between this outer sphere and inner sphere of the mesh [15]. In FINE™/Marine the software chooses the area where the mesh is allowed to deform which gives less freedom to the user.

In a deforming mesh the mesh deforms simply by tracking a moving boundary in the mesh domain. There are many examples when a moving mesh boundary needs to be implemented, one example is a falling motion of a solid body. The motion of the object implements deformation of the cells depending on the size of the motion, which raises some questions regarding the quality of the simulation. The deformation of the cells needs to be treated in a robust and stable way in order to produce accurate and reliable results. Deformation and

stretching of the cells can also make the simulation unstable. This is because the velocities in the momentum equation becomes relative to the motion of the cells in the mesh. This adds instabilities to the simulation and lower order schemes as first order upwind can then be used to make the simulation stable at the cost of accuracy in the simulation [23].

The deformation of the mesh clearly has an impact on the mesh quality, body fitted grids and complex geometries have cells with large skewness and slope discontinuities. Stretching and deforming of those complex cells can have a large impact on the scheme performance. Explicit second order schemes can lose accuracy when large cell variations occur over a time step. The superior behaviour of the explicit second order schemes compared to the lower order schemes is then lost if large stretching or deformation occurs over one time step in a cell [23].

The treatment of the free stream is also an important aspect to keep in mind when using a deforming mesh. The stretching and deforming of the cells in the mesh can have clear effects on the free stream. It has also been shown that higher order schemes can lose its superior behaviour compared to the lower order schemes when it comes to the preservation of the free stream in deforming meshes [23].

Compared to the overset mesh technique described in section 2.10 and sliding mesh technique described in section 2.12, the deforming mesh technique is superior with respect to computational time. The deforming mesh technique is very simple compared to the other two techniques [24]. However the deforming mesh has some clear application limitations, the movement of the body has to be limited. A too large movement of the body creates too much distortion of the cells in mesh and the simulation then becomes too unstable and produce inaccurate results or doesn't converge [24]. The deforming mesh technique also has limitations when the body is rotating. When the body rotates the mesh follow the rotating object. This can also produce instabilities in the simulation.

2.12 Sliding Mesh Technique

The sliding mesh method is commonly used to treat the motion of solid objects within CFD, the basic idea of a sliding mesh is to create one mesh for the entire computational domain. The computational domain is divided into a static part and a moving part based on frame of reference. A sliding interface separates the static and moving part which moves relative to each other in small time steps with the interface between. Both the mesh and the interface needs to be updated every time step during the simulation. The sliding mesh method doesn't lead to any topological changes of the cells or numerical instabilities during the simulation [25].

In OpenFOAM the sliding mesh technique is easily available in OpenFOAM and can be implemented for most cases. However for bodies with vertical movements the sliding mesh technique can be implemented with some modifications of the deforming mesh technique available in OpenFOAM described in section 2.11. To get a mesh that slides over the interface as described in the sliding mesh technique for FINE™/Marine, a deforming mesh with a very large deformation radius can be implemented. A large domain with large cells towards the boundaries of the domain can be implemented, the cells in the domain of interest in the simulation is set to an appropriate size for the simulation. The two deformation radii in the simulation are then set so just the larger cells towards the boundaries are allowed to deform. The mesh in the area of interest will then be intact and can freely slide over the interface. This technique can than be considered as a sliding mesh technique since the mesh is moving over the interface without deforming.

In FINE™/Marine one mesh is created for the whole domain. The mesh then moves along with the body without any deformation of the cells in the mesh. Adaptive grid refinement is then mandatory to use for this technique to be able to treat that the mesh boundaries slides in and out of the domain. New cells will then be created as mesh moves into the domain [19].

3 Method

The method for the master thesis is described in this chapter. The experimental data that is used for validating some of the simulation cases is presented. Then the computational geometry and mesh in both FINE™/Marine and OpenFOAM are illustrated. The numerical setup and boundary conditions for the simulations are also presented, the method for the incoming waves is shown last.

3.1 Computational time

Many CFD engineering problems are very computationally heavy and can't be run on ordinary laptops or stationary computers because of the time aspect. Therefore it's common to use computer clusters with multiple cores in order to solve the CFD cases. When comparing the time to run different simulations one has to consider the amount of cores the simulations are run with, in general more cores is equal to more computational power. In this project the different simulations are solved on different amount of cores and a good measurement to compare the aspect is therefore needed. To be able to get a comparison of the simulation times in this project the computational time for a simulation is multiplied with the amount of cores in order to get the amount of core hours. It is more then the amount of cores that determines the calculation speed of a computational cluster, but the amount of core hours can give a rough estimation.

3.2 Forces

The forces in the system can be calculated by setting up a force balance and assuming a constant water surface level around the sphere. The position over time for the sphere can be obtained from the simulations. The total force, F_{tot}, on the sphere is then given by multiplying the acceleration of the sphere with the mass. Gravity force, F_g, is constant for the system during the entire simulation, the buoyancy force F_b is estimated by calculating the submerged volume of the sphere from its position and multiplying it with the density of water and the gravitational constant. With the force balance

$$F_{tot} = F_g + F_b + F_{dynamic} \tag{3.1}$$

the dynamic forces, $F_{dynamic}$, can be estimated since all the other forces are known [13].

3.3 Experimental data, 0.15 m radius sphere

In the first part of this project the heave motion for a 0.15 m radius sphere is simulated. The sphere is dropped with its center of mass 0.15 m above a resting water surface and the heave motion is investigated. The result from the simulations are then compared with experimental data to validate the simulations. The data for the experiment is retrieved from [26]. The physical properties of the system is given in Table 3.1.

Table 3.1: Physical properties for the experiment.

Parameter	Value
Diameter of sphere [m]	0.3
Mass of sphere [kg/m^3]	7.056
Acceleration due to gravity [m^2/s]	9.82
Water density [kg/m^3]	998.2
Water dept [m]	0.9
Kinematic viscosity of water [m^2/s]	0.1 10^{-6}
Kinematic viscosity of air [m^2/s]	0.2 10^{-6}
Density air [kg/m^3]	1.2
Surface tension water air [N/m]	0.07
Temperature of water and air [C°]	20

The drop in the experiment is with the spheres center of mass elevated 0.15 meters from the resting water surface, the sphere is then completely out of the water. The heave motion of the sphere is then monitored for six seconds, the data for the position of the sphere center of mass is shown in Figure 3.1.

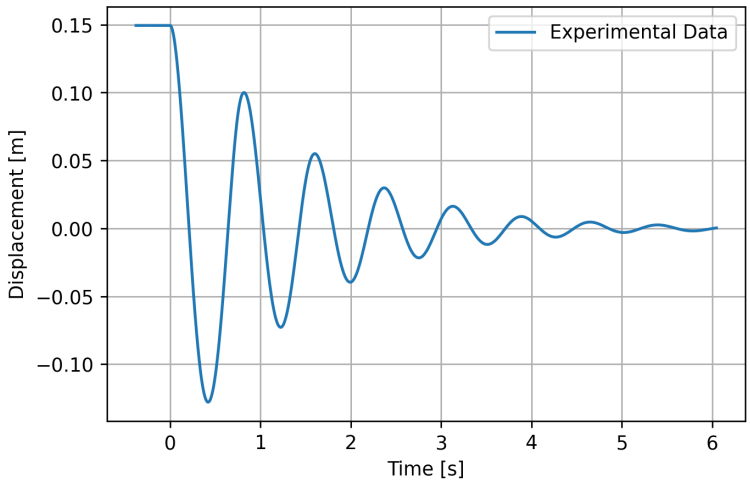


Figure 3.1: *Displacement of the sphere over time for the experiment.*

The data shown in 3.1 is used in order to validate the simulations which have the same physical parameters as the experimental setup.

To be able to evaluate the methods quantitatively a fitted dampening curve is adapted for the experimental data. The dampening curve is describing the decreasing trend of the heave motion for the sphere. The first oscillation is not considered since it has a different behaviour because the sphere starts above the water in the first oscillation. The fitted dampening curve is described according to equation

$$x(t) = x_0 e^{-\theta t}, \tag{3.2}$$

where $x(t)$ is the position in z direction of the spheres center of mass. x_0 is the position of the sphere center at the top of the first oscillation. θ is the dampening coefficient and is calculated with equation

$$\theta\tau = log\Big(\frac{x_0 - x_1}{x_2 - x_3}\Big), \tag{3.3}$$

where τ is the wave period. x_1 is the first trough and x_2 is the peak after that, then it continues with the same pattern. The simulations are then evaluated quantitatively based on how close the dampening coefficient θ is compared to the experimental. The dampening coefficient for the experimental data is 0.774, the fitted dampening curve is shown in Figure 3.2.

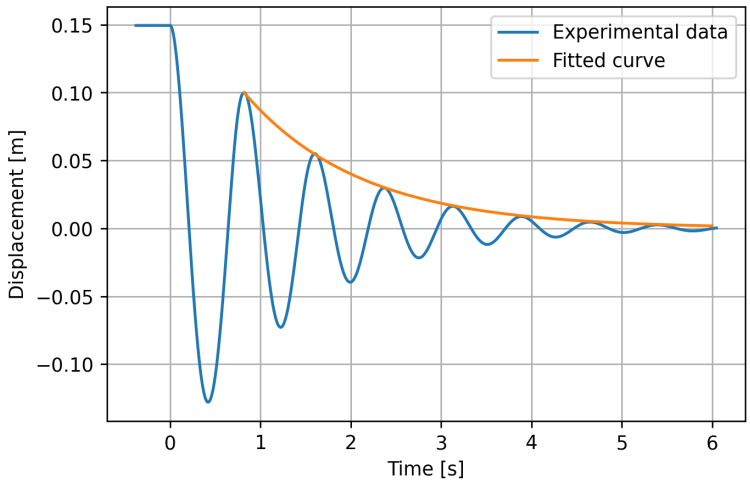


Figure 3.2: *Fitted dampening curve of the heave motion for the experimental data of the 0.15 meter radius sphere.*

3.4 Numerical data for five meter radius sphere

There have been previous studies where the heave motion of a five meter radius sphere has been investigated. In the studies the heave motion of the five meter sphere is investigated with a large variation of numerical models. The environment in the numerical models is the same as given in Table 3.1 but with an infinite water depth, five meter radius sphere and the weight of the sphere is 261.8 tons.

The sphere is dropped from five meters above the resting water surface and the heave motion is tracked by the numerical models for 40 seconds. The results from the different models is given in Figure 3.3.

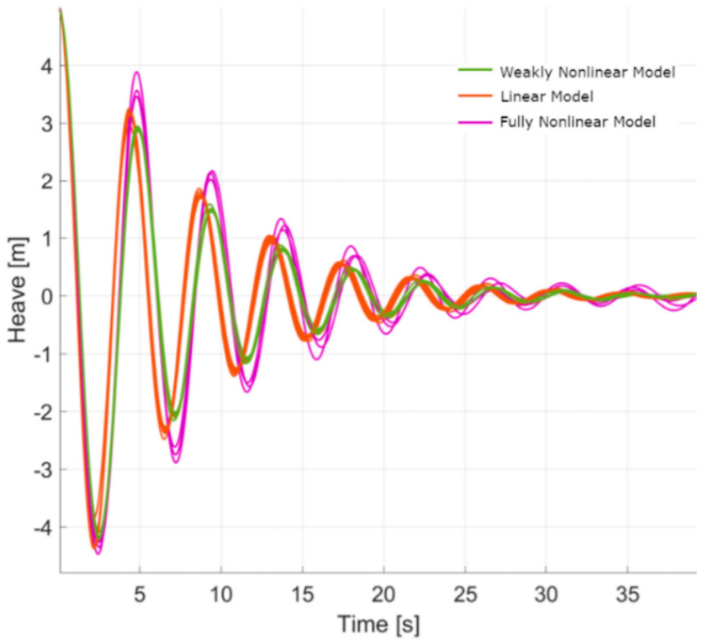


Figure 3.3: *Numerical data for drop of a five meter radius sphere from five meters, retrieved from [13].*

As shown in Figure 3.3 the numerical models show very similar results for the heave motion. The linear, weakly linear and fully nonlinear models has some differences in the results but models with the same linearity follows each other quite well.

Figure 3.4 shows the fitted curve for the dampening motion of the fully nonlinear motions. Only the fully

nonlinear models is used since these are other CFD codes which are most interesting for our case. The curve is calculated with equation 3.3 and 3.2, the dampening coefficient for this curve is 0.1301. The dampening coefficient for this curve is then compared with the simulations in this project.

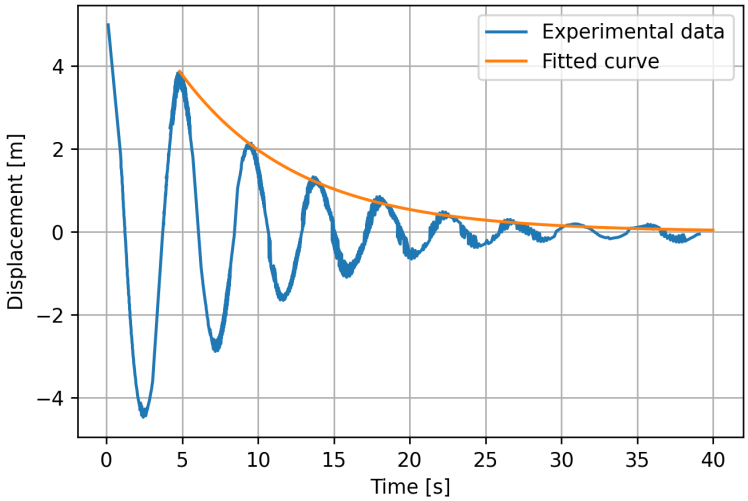


Figure 3.4: *Fitted curve for the dampening of the numerical data for the heave motion of the five meter sphere for the fully nonlinear models.*

3.5 Simulation domains

The size of the simulation domain is very important, the boundaries of the simulations needs to be set at a reasonable distance from the area of interest in the simulations in order to get accurate results. Different meshing techniques also requires different sizes of the domain which are explained later in the report. Simulation of incoming waves also sets different requirements on the domain compared to the free decay cases.

3.5.1 Domain 1, D.1

The choice of domain D.1 is based on both the experimental setup and computational time. Domain D.1 is used for overset mesh simulations in both FINE™/Marine and OpenFOAM. The water dept is chosen to be 0.9 m to represent the water dept in the pool in the experiment. The sphere is placed in the middle of the domain and the walls are placed 3.6 m from the sphere, this distance is more then 10 times the diameter of the sphere to make sure the walls are not affecting the heave motion of the sphere. This also ensures that the water volume in the simulation is large enough so the submerged volume of the sphere during its heave motion doesn't affect the water surface level. In Figure 3.5 is the simulation domain is shown in 2D in the x, z direction. The line below the sphere represents the water interface.

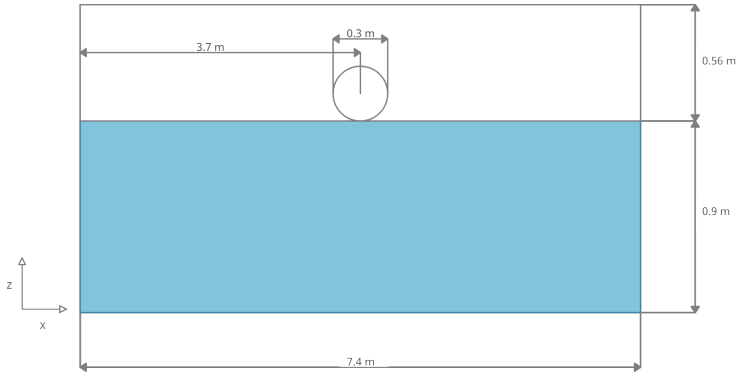


Figure 3.5: *2D figure of simulation domain D.1 in x and z direction.*

Figure 3.6 shows a 2D representation of the simulation domain in y, z direction. In this simulation domain the x and y direction has the same length.

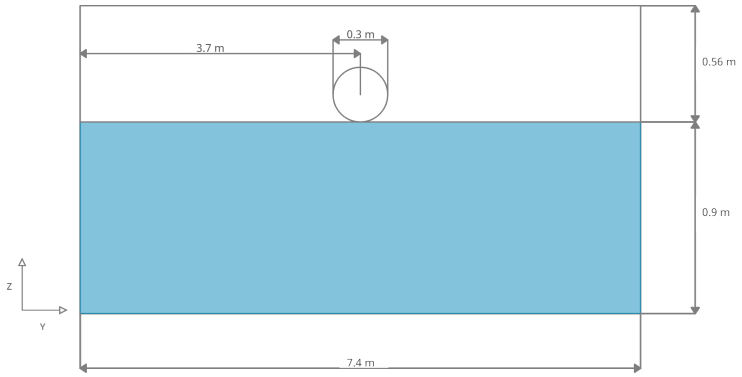


Figure 3.6: *2D figure of simulation domain D.1 in y and z direction.*

To summarize the representation in Figure 3.5 and 3.6 the dimension is presented in Table 3.2

Table 3.2: Dimensions of domain D.1

Water dept [m]	0.9
Distance above water [m]	0.53
x [m]	7.4
y [m]	7.4
radius sphere [m]	0.15

3.5.2 Domain 2, D2

The deforming mesh requires a larger domain in order to be able to handle the deformation in a accurate and stable way. This is because each cell deforms less in a larger domain. The dimensions of domain D.2 is given in Table 3.3, the distance above the water surface is prolonged. This is so a larger deformation radius can be defined in OpenFOAM and also a larger deformation can occur in FINE™/Marine. The sphere is placed 3.6 m from the walls in both x and y direction.

Table 3.3: Dimensions of domain D.2.

Water dept [m]	0.9
Distance above water [m]	1.1
x [m]	7.4
y [m]	7.4
radius sphere [m]	0.15

3.5.3 Domain 3, D.3

Domain D.3 is used for the sliding mesh simulation in OpenFOAM which is described in section 2.12. This domain requires quite large dimensions compared to the other simulation domains with the 0.15 m radius sphere. This is because the inner and outer deformation radius specified in the deforming mesh technique needs to be set far from the area of interest in the simulation. In Table 3.4 the dimensions for domain D.3 is given. The sphere is with its center 6.5 m from the walls in both x and y direction.

Table 3.4: Dimensions of domain D.3.

Water dept [m]	7.1
Distance above water [m]	6.57
x [m]	13
y [m]	13
radius sphere [m]	0.15

3.5.4 Domain 4, D.4

In domain D.4 the sphere is scaled up from a radius of 0.15 m to 5 meter. This simulation then requires a larger domain, since the sphere is scaled up 33.3 times and an overset mesh is used for this simulation as for domain D.1. Domain D.1 is therefore scaled up 33.3 times so the walls have a good distance from the sphere and also get a reasonable depth of the pool. The dimensions for domain D.4 is given in Table 3.5, the sphere is placed 123 m from the walls both in x and y direction.

Table 3.5: Dimensions of domain D.4

Water dept [m]	30
Distance above water [m]	18.75
x [m]	246
y [m]	246
radius sphere [m]	5

3.5.5 Domain 5, D.5

Domain D.5 is used for incoming wave simulations in OpenFOAM. There are some recommendations when it comes to the dimensions of the domain in wave simulations which are described in section 2.5. In these simulations the wave have a wavelength of 1.55 m, this generates the dimensions given in Table 3.6. The waves travel in the positive x direction in the simulation. The sphere is placed 3.25 m from the inlet in x direction and 1.7 m from the outlet. In y direction the sphere is placed 3.1 m from both the walls.

Table 3.6: Dimensions of domain D.5.

Water dept [m]	6.2
Distance above water [m]	3.1
x [m]	4.95
y [m]	6.2
radius sphere [m]	0.15

3.6 Mesh generation

In this section the meshes used in the simulations for both simulation software are presented. The design of the different meshes are motivated and shown. Since different meshing techniques are used in this project a new mesh has to be created for each meshing technique. The techniques are very different from each other, and each meshing technique therefore has different criteria that needs to be fulfilled.

3.6.1 Mesh generation, overset mesh M.1

The overset mesh has both a static and moving mesh, which accurately can resolve the moving waves that occurs during the heave motion on the water surface. The moving mesh around the sphere can resolve the flow in the vicinity of the sphere accurately, it also communicates with the static mesh in a stable manner.

The size ratio between the moving and static mesh is important for the accuracy of the simulation. A similar size ratio of the cells between the meshes is necessary so the information can be exchanged correctly between the cells of the two meshes. The cell size around the sphere needs to be a bit smaller to ensure good resolution of the flow close to the surface of the sphere. This leads to that some cells in the moving mesh is smaller than in the static mesh, the volume ratio of the cells between the moving mesh and the static mesh was limited to 10 to avoid numerical instabilities when information is exchanged between the two meshes.

For the static mesh there is needed refinements along the water surface and in the middle of the domain where the heave motion of the sphere occurs. The heave motion of the sphere creates propagating waves along the water surface. To ensure accurate resolution of the propagating waves and sharpness of the water air interface the refinement is done along the whole water surface in the simulation. Since the sphere has a large heave motion during the simulation the moving mesh have a large movement during the simulation. The refinements in the static mesh needs to cover the whole domain where the moving mesh is moving during the simulation to ensure a good size ratio of the cells along the whole movement. The final mesh is shown in Figure 3.7, where the red area in the figure represents the moving mesh. The amount of cells for M.1 is 573 119.

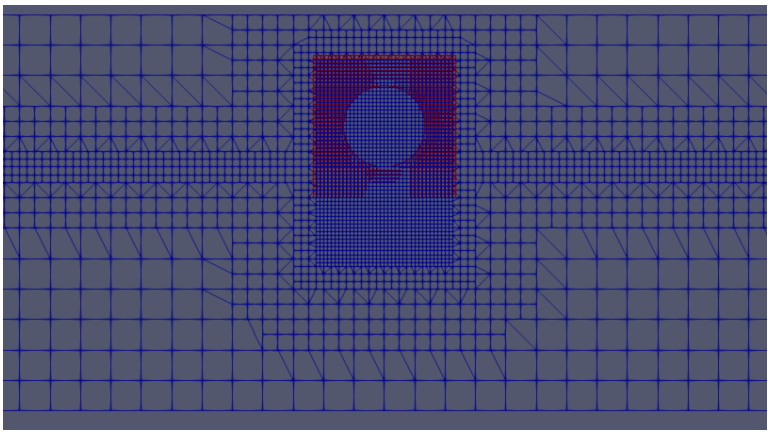


Figure 3.7: *Mesh M.1.*

The overset mesh technique works very similar in OpenFOAM and FINE™/Marine, the same meshing technique as described above can therefore be applied in both software.

3.6.2 Mesh generation, deforming mesh M.2

When creating a deforming mesh the movement of the object and the deformation of the mesh in the simulation needs to be considered. As described before, refinements around the sphere are important for accurate resolution of the flow around the sphere. There is also a need for refinements along the water surface in order to have a sharp water-air interface. However, since the mesh around the sphere is deformed and the cells moves along with the heave motion of the sphere, the refinements along the water-air interface needs to be wider in order to keep a sharp interface even when the cells move in the mesh. If the refinement along the water-air interface isn't increased the non refined cells are used to resolve water-air interface. The final mesh is shown in Figure 3.8, the number of cells in the mesh is 573 440.

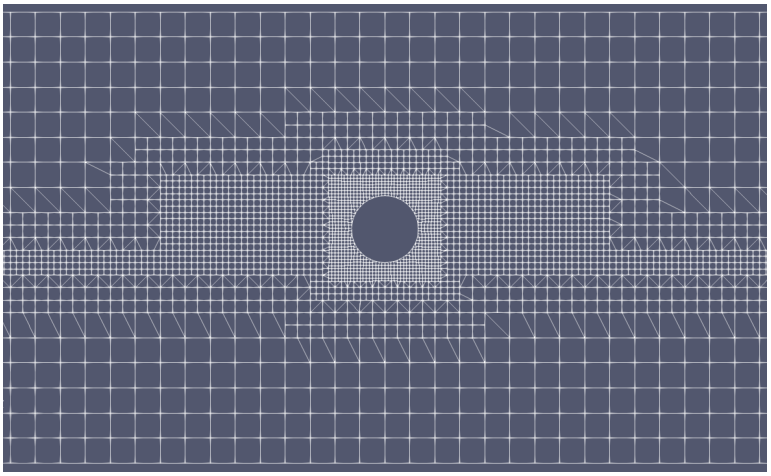


Figure 3.8: *Mesh M.2.*

3.6.3 Mesh generation, sliding mesh OpenFOAM M.3

The sliding mesh in OpenFOAM requires a large mesh because only the coarse cells towards the outside of the domain shall deform, this technique is described in section 2.12. The idea behind this mesh is to refine the area of interest in the simulation and only let the coarse cells in the outside of the mesh deform. By doing this the mesh area close to the sphere is intact. The final mesh can be seen in Figure 3.9, the refined box is five meters in the x, y, z direction and the sphere is placed in the middle. The water air interface is refined all the way out to the walls in order to keep a sharp interface in the whole simulation domain. The final cell count in M.3 is 729 000 cells.

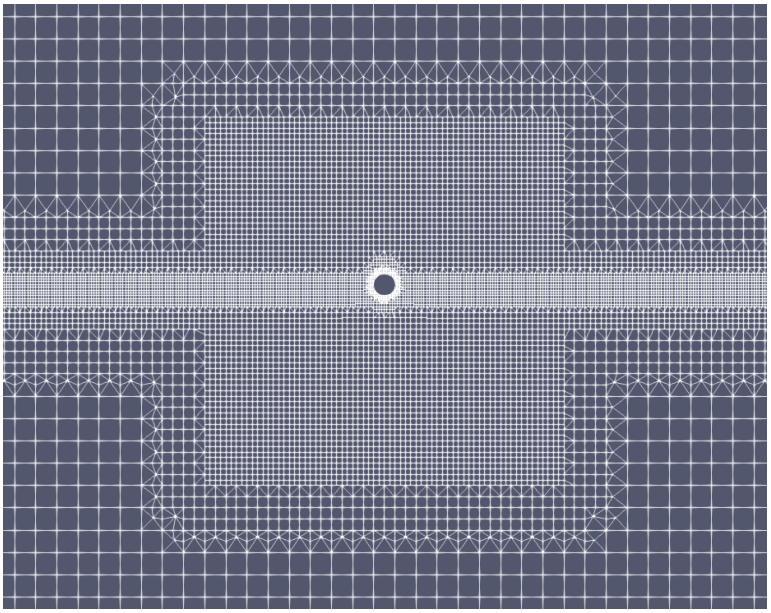


Figure 3.9: *Mesh M.3.*

The refinement at the water surface in Figure 3.9 is 0.5 m wide in the z direction. This is because the mesh moves and the refinement needs to cover the water surface during the whole heave motion of the sphere.

3.6.4 Mesh generation, sliding mesh FINE™/Marine M.4

The sliding mesh used in FINE™/Marine uses domain D.1. It is important to consider the movement of the sliding mesh during the simulation. The mesh moves along with the sphere, the refinement along the water surface therefore need to be wide to cover the interface during the whole simulation. The final mesh is shown in Figure 3.10, the final amount of cells in M.4 is 421 326.

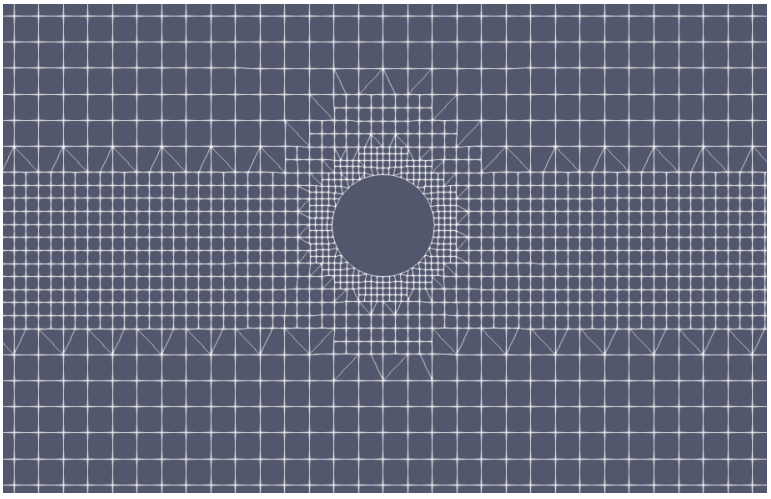


Figure 3.10: *Mesh M.4.*

3.6.5 Mesh generation, large sphere M.5

In mesh M.5 the mesh is scaled up because the spheres radius is changed from 0.15 m to five meter, an overset mesh method is used in both FINE™/Marine and OpenFOAM. The mesh M.5 is similar to mesh M.1 since the simulation case is the same but with a larger sphere, all the refinements are made on the same places as in M.1. The cell size is scaled up 33.3 times in all direction since the sphere is scaled up 33.3 times. The final mesh is

shown in Figure 3.11, the number of cells is 576 087 which is very similar to M.1 which has 573 119 cells.

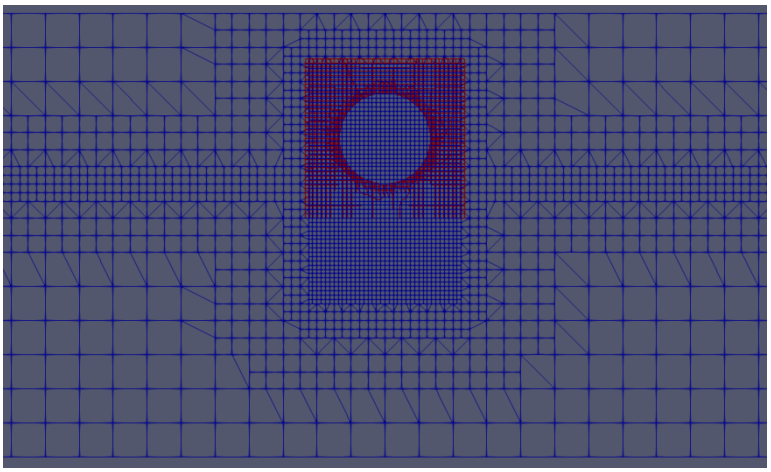


Figure 3.11: *Mesh M.5.*

3.6.6 Mesh generation, incoming waves M.6

In the mesh generation for the incoming wave case domain D.5 is used to generate a mesh that fulfill the cell size recommendations described in section 2.5. Here an overset mesh method is used, so an additional mesh around the sphere is created. The foreground mesh have the same size as the static mesh so that acceptable levels of interpolation are achieved.

The static mesh is also modified so that it have an constant size of cells in the space that the moving mesh occupies when the sphere is allowed to have its expected heave motions. The final mesh is shown in 3.12 where the waves enter from the left side, the number of cells is 2 142 335.

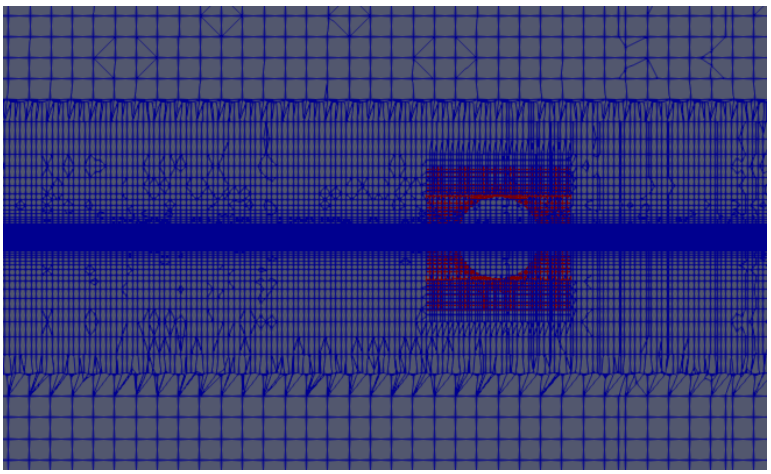


Figure 3.12: *Mesh M.6.*

3.6.7 Summation of mesh generation

To be able to get an overview of the different meshes used in this project, a summation is given in Table 3.7. Both the mesh method and the amount of cells is given in the table.

Table 3.7: Summation mesh generation.

Mesh	Mesh method	Number of cells
M.1	Overset	573 119
M.2	Deforming	573 440
M.3	Sliding	729 000
M.4	Sliding	421 326
M.5	Overset	573 119
M.6	Overset	2 142 335

3.7 Simulation setups

In this section the different simulation setups are presented, boundary and initial conditions are shown for each simulation setup. The simulation case is then given a reference number and letter to easily be able to refer to them later in the report.

All the simulations are modelled as isothermal and incompressible, this is because both the density and temperature are assumed to be constant in the system. The transport model is chosen as Newtonian for both the water and for the air in the simulations since the temperature is assumed to be constant and the viscosity is assumed to be constant as well. The experimental data given in section 3.3 are used as an validation of many of the simulations and all the simulations have the same physical properties as in the experiment given in Table 3.1. The time step in the simulations is 0.001 s. The velocities in the simulations are not expected to become very large, the time step is therefore considered low enough to produce a low CFL number and capture all the important forces in the flow that affect the simulation.

3.7.1 Simulation setups OpenFOAM

Each simulation case in OpenFOAM is given the reference combination, an O then a number. The different domains and meshes used in all the simulations are specified in this chapter. Physical models, boundary conditions, and numerical models are also presented. The sixDoFRigidBodyMotion solver is used to solve the motion in all case setup together with the CrankNicolson motion solver.

OpenFOAM simulation setup one, O.1

Simulation O.1 uses domain D.1 and mesh M.1, an overset mesh technique is used for the simulation. The sphere in the simulation is dropped with its center of mass 0.15 m above the resting water surface as in the experiment. The boundary conditions for the simulation is shown in Table 3.8.

Table 3.8: Boundary conditions for simulation O.1.

	stationaryWalls	atmosphere	floatingObject
zoneID	zeroGradient	zeroGradient	zeroGradient
pointDisplacement	fixedValue	fixedValue	calculated
p_rgh	fixedFluxPressure	totalPressure	zeroGradient
alpha.water	zeroGradient	inletOutlet	zeroGradient
U	fixedValue	pressureInletOutletVelocity	movingWallVelocity

The simulation is chosen to be laminar because the drop height of the sphere is low so it has a relative low speed. The numerical schemes and solution methods are very important for the accuracy of the simulations. In OpenFOAM this is specified in the files fvSchemes and fvSoloution, however these text files becomes very large and the user needs to specify a lot of information to get an accurate solution. Therefore these text files is shown in appendix, the fvSchemes file used in simulation case O.1 is represented in Appendix A.1 and the fvSoloution file can be seen in A.2.

OpenFOAM simulation setup two, O.2

In this simulation setup three cases are presented, all the three cases uses domain D.1. The setup for these two cases are exactly the same as in O.1 but with two new meshes. One 50 % coarser and one 50 % finer mesh are made. This is to test that mesh M.1 used in simulation O.1 is robust and stable. Since an overset mesh method is used there are two meshes to consider, one static and one moving. When M.1 is made coarser and finer the moving and static mesh is changed so the same size ratio always is kept between the two meshes in the overset mesh method. The amount of cells in each mesh can be seen in Table 3.9.

Table 3.9: Number of cells in each mesh for the mesh study in OpenFOAM

	Number of cells
Coarse	286 543
Medium (M.1)	573 119
Fine	930 596

OpenFOAM simulation setup three, O.3

Simulation O.3 uses domain D.1 and mesh M.1, the simulation case uses an overset mesh technique. In order to be able to model the turbulence around the sphere, the cells are refined close to the sphere. The simulation setup is very similar to O.1 with the sphere placed 0.15 m above the water surface but in this simulation case the effect of turbulence is investigated. The k-ω SST model is used for the turbulence model because it can resolve a potential boundary layer around the sphere and the model can handle potential turbulence in the free stream. The introduction of the k-ω SST model requires some extra boundary conditions, the boundary conditions for simulation O.3 is given in Table 3.10.

Table 3.10: Boundary conditions for simulation O.3.

	stationaryWalls	atmosphere	floatingObject
zoneID	zeroGradient	zeroGradient	zeroGradient
pointDisplacement	fixedValue	fixedValue	calculated
p_rgh	fixedFluxPressure	totalPressure	zeroGradient
alpha.water	zeroGradient	inletOutlet	zeroGradient
U	fixedValue	pressureInletOutletVelocity	movingWallVelocity
k	kqRWallFunction	inletOutlet	kqRWallFunction
nut	nutkWallFunction	calculated	nutkWallFunction
omega	omegaWallFunction	inletOutlet	omegaWallFunction

The turbulence modeling requires some modification of the fvSchemes folder given in Appendix A.1.Divergence schemes needs to be specified for both ω and k, linearupwind is chosen for both properties since it is a second order scheme.

OpenFOAM simulation setup four, O.4

Simulation O.4 uses domain D.2 and mesh M.2, the mesh technique for this case is a deforming mesh. Just like simulation setup O.1 this simulation is meant to reproduce the result from the experiment which can be seen in Section 3.3. The sphere is placed 0.15 m above from the water surface, the boundary conditions in simulation O.4 is the same as in Table 3.8. The numerical schemes/solvers are given in appendix Section A.3 and A.4 respectively. A deforming mesh case requires that an inner and an outer deformation radius is specified with its centre defined in the centre of the sphere. The inner radius is defined as 0.25 m which means the cells around the sphere are left intact and not deformed. The outer deformation radius is set to 0.8 m.

OpenFOAM simulation setup five, O.5

Simulation setup 5 uses domain D.3 and mesh M.3, the case uses a sliding mesh method. Setup O.5 uses the boundary conditions given in Table 3.8 and the fvSchemes and fvSoloution are given in Appendix A.3, A.4 respectively. This method also requires that an inner and an outer deformation circle is defined in OpenFOAM

so the mesh can move freely over the interface which is explained in Section 2.12. The inner radius is specified as three meter and the outer radius is given as 6.2 m, this leaves the cells unchanged in a three meter radius circle with its centre defined in the centre of the sphere. The sphere is initially placed 0.15 m over the resting water surface. The flow model in the simulation is set to laminar.

OpenFOAM simulation setup six, O.6

In simulation setup O.6 is the simulation scaled up since a five meter radius sphere is used, the mesh method is an overset mesh. Mesh M.5 and domain D.4 are used for simulation setup O.6. The sphere is initially placed five meter above the resting water surface, the density of the sphere is the same as the 0.15 m radius sphere and the new mass is 261.8 tons. There is no experimental data to validate this simulation with but there are a lot of numerical data which is presented in Section 3.4. Even if the simulation is scaled up the same boundary conditions used in simulation O.1 given in Table 3.8 can be used for this simulation. The fvSchemes and fvSoloution is given in Appendix A.1, A.2 respectively. A laminar flow model is also used for this simulation since the turbulence is expected to have little impact on the simulation.

OpenFOAM simulation setup seven, O.7

The five meter radius sphere is also used in simulation O.7, mesh M.5 and domain D.4 are used in this simulation. In order to model the turbulence around the sphere accurately, the cells are refined close to the sphere. The sphere is dropped with its center of mass five meter above the resting water surface and its heave motion is then investigated in 40 seconds. The turbulence model k-ω SST is chosen for this simulation in order to be able to investigate the effects of turbulence on the simulation. There is a large weight and size difference between the 0.15 meter radius sphere and the five meter sphere which means that the flow around the sphere could behave very different. The turbulent case O.7 is then compared with simulation O.6 to see the impact of the turbulence on the sphere.

OpenFOAM simulation setup eight, O.8

In simulation setup O.8 a NWT is created, waves are moving from the negative x direction to the positive x direction in the domain. The 0.15 meter sphere is used in this simulation and the sphere is initially placed at the water surface half submerged at its resting position, a laminar flow model is used for this simulation. In this simulation setup domain D.5 and mesh M.6 are used to meet the requirements for wave simulations specified in the theory section 2.5. The different Stoke wave theories are also described in this section, to get a realistic regular steady wave Stokes 2^{nd} wave theory is used. The different wave theories are shown in Figure 2.1 and to get a wave in Stokes 2^{nd} theory the parameters given in Table 3.11 is used.

Table 3.11: Wave parameters for simulation O.8.

	Unit
H [m]	0.02
τ [S]	1
λ [m]	1.55

A wave simulation requires some different boundary conditions than the previous case setups. The boundary conditions for simulation O.8 is given in Table 3.12.

Table 3.12: Boundary conditions for simulation O.8.

	bottom	sidewall	inlet
zoneID	zeroGradient	zeroGradient	zeroGradient
pointDisplacement	fixedValue	fixedValue	fixedValue
p_rgh	fixedFluxPressure	fixedFluxPressure	fixedFluxPressure
alpha.water	zeroGradient	zeroGradient	waveAlpha
U	fixedValue	fixedValue	waveVelocity
	outlet	atmosphere	floatingObject
zoneID	zeroGradient	zeroGradient	zeroGradient
pointDisplacement	fixedValue	fixedValue	calculated
p_rgh	fixedFluxPressure	totalPressure	zeroGradient
alpha.water	zeroGradient	inletOutlet	zeroGradient
U	waveVelocity	pressureInletOutletVelocity	movingWallVelocity

The ramp time in the simulation is set to two seconds. This means that fully developed waves do not enter the simulation until after two seconds.

To calculate the possible energy output from the sphere it is assumed that all potential energy that the sphere loses as it moves in the negative z direction can be ideally extracted. This is calculated by checking the change in position for every time step and when the sphere moves in negative z direction. The change in the potential energy is calculated according to

$$\Delta E = m\ g\ l, \tag{3.4}$$

where ΔE is the change in potential energy, m is the mass of the sphere, g is the gravitational constant and l is the change in z position. These are then accumulative added to show how much energy is produced over time.

Summation of the cases in OpenFOAM

The different OpenFOAM setups are summarized in Table 3.13. This gives an overview of the most important information for the cases so they can easily be comparable, where M.1* is the different meshes in the mesh study.

Table 3.13: Summation of the cases in OpenFOAM.

Case	Sphere radius [m]	Turbulence treatment	Mesh	Mesh method
O.1	0.15	Laminar	M.1	Overset Mesh
O.2	0.15	Laminar	M.1*	Overset Mesh
O.3	0.15	k-ω SST	M.1	Overset Mesh
O.4	0.15	Laminar	M.2	Deforming Mesh
O.5	0.15	Laminar	M.3	Sliding Mesh
O.6	5	Laminar	M.5	Overset Mesh
O.7	5	k-ω SST	M.5	Overset Mesh
O.8	0.15	Laminar	M.6	Overset Mesh

3.7.2 Simulation setups FINE™/Marine

In this section the simulations setups in FINE™/Marine are presented. The three different meshing methods overset, sliding and deforming are shown. The case for the large sphere is then presented, and last the case for incoming waves. Each case is referenced to with an F then a number. All the simulations in FINE™/Marine use the same discretization schemes for Turbulence, Momentum and Multifluid, these are given in Table 3.14

Table 3.14: Discretization schemes used in FINE™/Marine.

	Discretization scheme
Turbulence	AVLSMART
Momentum	HYBRID
Multifluid	AVLSMART

The simulation setups in FINE™/Marine is very similar to each other and therefore the same boundary conditions can be used for all the simulations, these are given in Table 3.15.

Table 3.15: Boundary conditions used in FINE™/Marine.

	Boundary condition
Sphere	No slip
Athmosphere	Prescribed pressure (updated hydrostatic pressure)
Bottom	Prescribed pressure (updated hydrostatic pressure)
Walls	Far field

FINE™/Marine simulation setup one, F.1

Case F.1 uses an overset mesh method, domain D.1 and mesh M.1 are used in this simulation. The sphere in this simulation is placed 0.15 meters above the resting water surface. A laminar flow model is chosen for this simulation.

FINE™/Marine simulation setup two, F.2

An mesh study for the overset mesh method is done in FINE™/Marine. The simulation setup is the same as in F.1, mesh M.1 is made 50 % coarser and 50 % finer. Since there are two meshes in M.1, one moving mesh and one static mesh. Its important that the ratio between the two meshes is not changed when changing the amount of cells, so the results are comparable. The number of cells for each mesh is seen in Table 3.16.

Table 3.16: Number of cells in each mesh for the mesh study in FINE™/Marine.

	Number of cells
Course	292 526
Medium (M.1)	555 616
Fine	902 206

FINE™/Marine simulation setup three, F.3

Simulation setup F.3 is using the turbulence model k-ω SST which is implemented in order to see the effect of the turbulence on the simulation. An overset mesh method is used for this simulation, domain D.1 and mesh M.1 is implemented for case F.3. In order to model the turbulence flow around the sphere, the cells are refined close to the sphere. The sphere is initially placed with its center of mass 0.15 meter above the water in order to investigate its heave motion.

FINE™/Marine simulation setup four, F.4

In simulation setup F.4 mesh M.2 is used with domain D.2, simulation F.4 uses a deforming mesh technique. The sphere is initially placed 0.15 meters above the water surface. This simulation is using a laminar flow model.

FINE™/Marine simulation setup five, F.5

Case F.5 is using domain D.1 and mesh M.4, the sliding mesh technique is used for this case. The 0.15 meters radius sphere is initially placed 0.15 meters above the water surface. The flow model in the simulation is chosen as laminar.

FINE™/Marine simulation setup six, F.6

The sphere is scaled up from 0.15 meters to five meters in simulation F.6. This simulation requires a larger domain so domain D.4 and mesh M.5 is used for this case. The sphere is initially placed five meters above the water surface before it's dropped. The density of the sphere is the same as the density of the 0.15 m radius sphere 499.11 kg/m^3, so the weight of the five meter radius sphere is 261.8 tons. The simulation is compared with the numerical data presented in Section 3.4. The case uses a laminar flow model because the effects from the turbulence is expected to be small.

FINE™/Marine simulation setup six, F.7

The five meter sphere is also used in simulation F.7 together with the turbulence model k-ω SST. In simulation F.7 mesh M.5 and domain D.4 is used, the sphere is dropped with its center of mass five meter above the water surface and its heave motion is then investigated for 40 seconds. In order to model the turbulence around the sphere accurately, the cells are refined close to the sphere. There is a huge scale difference between the 0.15 meters sphere and the five meters sphere which means that the flow around the sphere could be very different between the two cases. The turbulent case F.7 is compared with the laminar case F.6 to be able to investigate the effect of turbulence.

Summation of the cases in FINE™/Marine

The different FINE™/Marine setups are summarized in Table 3.17. This gives an overview of the most important information for the cases so they can easily be comparable.

Table 3.17: Summation of the cases in FINE™/Marine.

Case	Sphere radius [m]	Turbulence treatment	Mesh	Mesh method
F.1	0.15	Laminar	M.1	Overset Mesh
F.2	0.15	Laminar	M.1	Overset Mesh
F.3	0.15	k-ω SST	M.1	Overset Mesh
F.4	0.15	Laminar	M.2	Deforming Mesh
F.5	0.15	Laminar	M.4	Sliding Mesh
F.6	5	Laminar	M.5	Overset Mesh
F.7	5	k-ω SST	M.5	Overset Mesh

4 Results

In this chapter the results from this project are presented, the simulation results from both CFD software are compared against the experimental data. The results from the heave motion of the sphere during incoming waves are also analysed and presented in this section.

4.1 Comparison of the mesh methods

In Figure 4.1 the heave motion over six seconds with the deforming mesh technique is shown. Here the x axis represents the time in seconds and the y axis shows the location of the center of mass. The experimental data is also shown in the figure in order to get a good comparison. In Figure 4.1 it is clear that both F.4 and O.4 differ from the experimental data, O.4 is following the experimental data quite well in the beginning and then the heave motion is damped too quickly. F.4 is also following the experimental data in the beginning of the heave motion but then the dampening is underestimated towards the end.

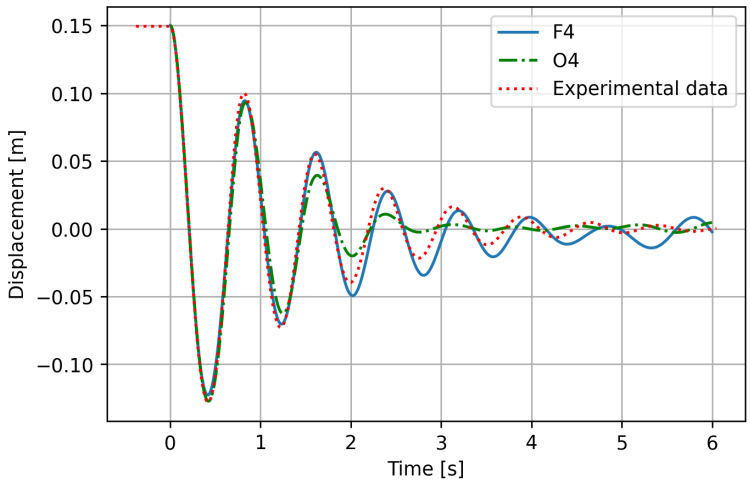


Figure 4.1: *Comparing displacement of the sphere between simulation O.4, F.4 and the experiment.*

Figure 4.2 shows the result for the heave motion of the 0.15 mesh sphere with the sliding mesh technique. The figure shows that simulation O.5 is quite accurate. O.5 follows the experimental data from the beginning of the heave motion to the end, it is some millimeters off in the beginning but the difference between the experimental data and simulation O.5 becomes smaller towards the end of the simulation. Simulation F.5 follows the experimental data quite well up to around two seconds then it is a bit different from the experimental result but the difference is small.

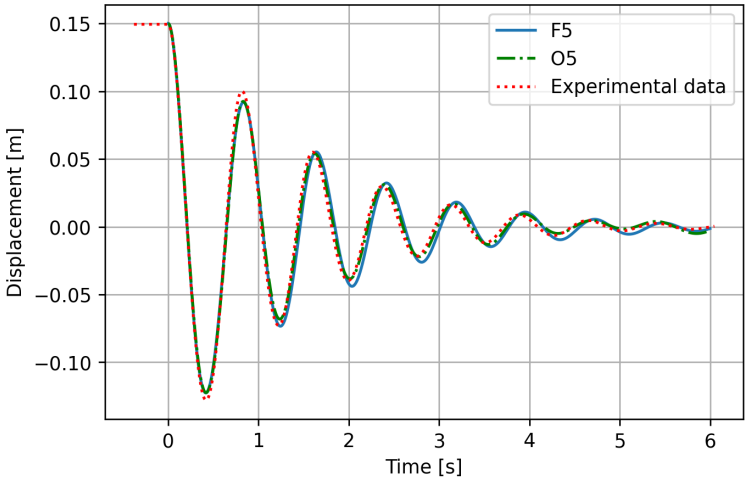


Figure 4.2: *Comparing displacement of the sphere between simulation O.5, F.5 and the experiment.*

The result of the heave motion for the overset mesh method is shown in Figure 4.3. The result for both simulation F.1 and O.1 is quite accurate, both simulations follow the experimental results very well from the beginning of the simulation to the end. There is a slight difference towards the end of the heave motion between simulation O.1 and F.1.

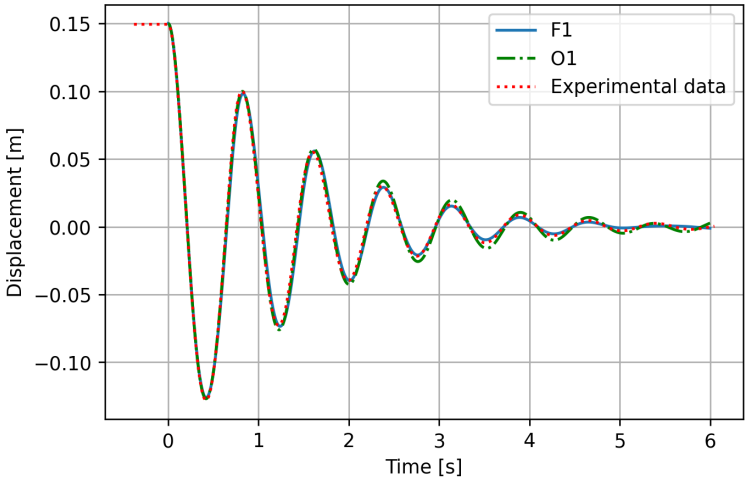


Figure 4.3: *Comparing simulation O.1, F.1 and experimental data.*

In Table 4.1 the dampening coefficients and the difference from the experimental dampening coefficient are shown. Both FINE™/Marine and OpenFOAM show similar results for the different mesh techniques. Deforming mesh has a large difference, sliding mesh has a smaller difference and overset mesh is most accurate.

Table 4.1: Evaluation of the dampening coefficient for the different mesh techniques.

	θ	difference	difference in %
Experiment	0.774	-	-
F4	0.5570	0.2170	28.0
O4	1.2649	0.4909	63.4
F5	0.6466	0.1274	16.5
O5	0.7021	0.0719	9.28
F1	0.7525	0.0215	2.78
O1	0.7277	0.0463	6.00

In Table 4.2 the computational time for the different simulations are given as well as the number of cores that's assigned to the simulation. O.1 and F.1 are the most time consuming simulations, the overset mesh method is very computational heavy but it also generates accurate results. The sliding mesh technique is the second most time consuming technique and deforming mesh technique is the least computational heavy method. Since the simulations have been made on different computers one should compare the core hours and not the computational time.

Table 4.2: Number of cores and computational times for the different meshing methods.

	Computational time [h]	Number of cores	Core hours [h]
F4	28.69	4	114.7
O4	1.92	20	38.41
F5	6.0	32	192.08
O5	6.71	20	134.2
F1	5.79	32	185.1
O1	13.2	20	263.0

Figure 4.4 shows alpha.water with the cells in the mesh as a background at 0.3 seconds in simulation O.4. The cell deformation can be seen can be seen in the figure. It can also be seen that the water interface becomes smeared out were the deformation of the cells occurs.

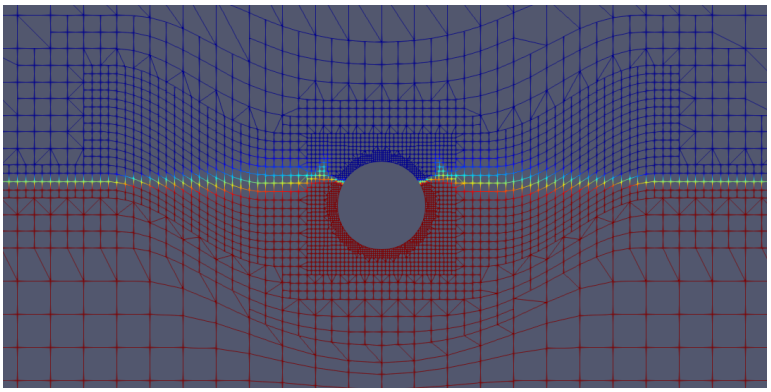


Figure 4.4: *The colour function shown with the cells in the mesh at 0.3 seconds for simulation O.4*

4.2 Overset mesh study

To test the stability and robustness of the mesh in simulation O.1 a mesh study was performed. Three different meshes were used for the simulation, one coarse, medium, and one fine mesh. The three meshes are presented in Section 3.7.1, the result for the simulations in OpenFOAM is presented in Figure 4.5. The figure shows a very similar result for the fine and the medium mesh. The coarse mesh however deviates a bit from the other meshes and the experimental data towards the end of the heave motion. This shows that the medium mesh has an appropriate amount of cells since if it's coarser the result is less accurate and if it's finer the result is not

affected noticeably.

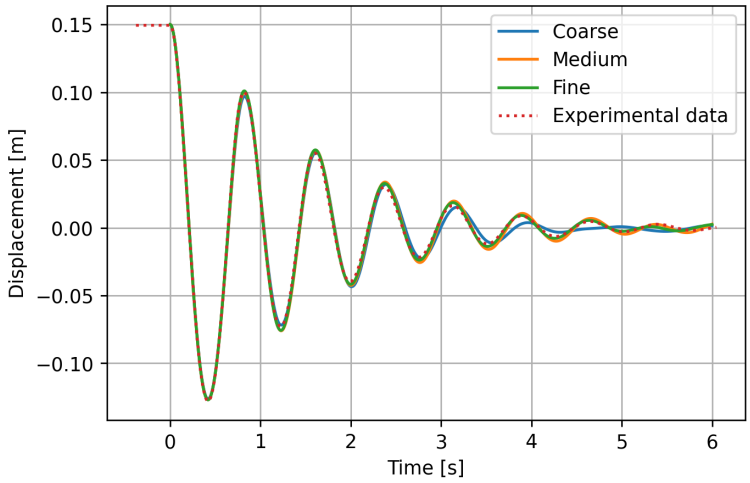


Figure 4.5: *Comparing displacement of the sphere between the fine, medium and course overset mesh in OpenFOAM.*

A mesh study is also performed in FINE™/Marine in order to test that the mesh are stable and robust in FINE™/Marine. The three meshes used in this are presented in Section 3.7.2 and the result are presented in Figure 4.6. A similar trend can be observed in FINE™/Marine as in OpenFOAM which can be seen in the figure. The fine and the medium mesh shows very similar results while the coarser mesh deviates a little, however the deviation for the coarser mesh is very small.

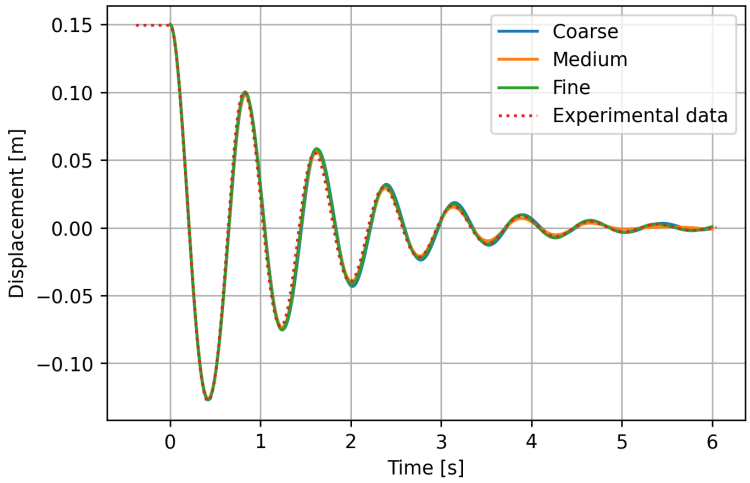


Figure 4.6: *Comparing displacement of the sphere between the fine, medium and course overset mesh in FINE™/Marine.*

In Table 4.3 the computational time is given for the overset mesh study, simulations in both FINE™/Marine and OpenFOAM, the number of cores each simulation utilise is also presented. The amount of cells in the mesh are also shown since this has a big impact on the computational time. The result is as expected, the finer mesh has the longest computational time compared with the medium and coarse mesh. The simulation time is a bit lower in FINE™/Marine than in OpenFOAM.

Table 4.3: Number of cores and computational times for the overset mesh study.

	Computational time [h]	Number of cores	Core hours [h]	cell count
OpenFOAM				
Fine	26.8	20	536.9	930 596
Medium	13.2	20	263.0	573 119
Coarse	8.56	20	171.3	286 543
FINE™/Marine				
Fine	13.7	32	438.4	930 596
Medium	5.79	32	185.1	573 119
Coarse	32.94	32	131.7	286 543

Table 4.4 shows the dampening coefficient for the different meshes and how they differ from the experimental dampening coefficient. In OpenFOAM the dampening coefficient becomes slightly better for the finer meshes. In FINE™/Marine the coefficient is closest to the experiment for the medium mesh then the difference becomes slightly larger for the finer mesh. However the difference is very small.

Table 4.4: Evaluation of the dampening coefficient for the mesh study.

	θ	difference	difference in %
Experiment	0.774	-	-
OpenFOAM			
Fine	0.7436	0.0304	3.93
Medium	0.7277	0.0463	6.00
Coarse	0.6720	0.1020	13.7
FINE™/Marine			
Fine	0.7228	0.0512	6.61
Medium	0.7525	0.0215	2.78
Coarse	0.6903	0.837	10.8

4.3 Overset mesh turbulence modeling

To investigate the effects of turbulence the result from both the laminar and turbulent simulations are presented here. The k-ω SST model has been used in the turbulent simulations, in Figure 4.7 the result from OpenFOAM is shown. Simulation O.3 is plotted along with the laminar simulation O.1 and the experimental data in order to be able to see the difference between the laminar and turbulent simulation. In the figure it can be seen that the result from the turbulent simulation doesn't deviate much from the laminar simulation or the experimental data. A small difference can be seen between two and four seconds, however the difference is small.

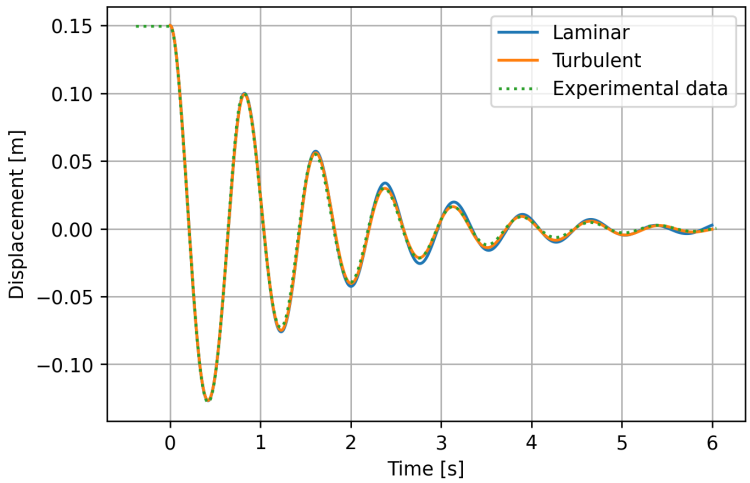


Figure 4.7: *Comparing displacement of the sphere between simulation O.1, O.3 and the experiment.*

A turbulent simulation was also set up in FINE™/Marine for the 0.15 meter sphere. In Figure 4.8 the result for the turbulent simulation F.3 is shown with the laminar simulation F.1 and the experimental data. In the figure it can be seen that the difference between the turbulent simulation F.3 and the laminar simulation F.1 is very small. There is only a small deviation in the smaller heave motions towards the end of the simulation.

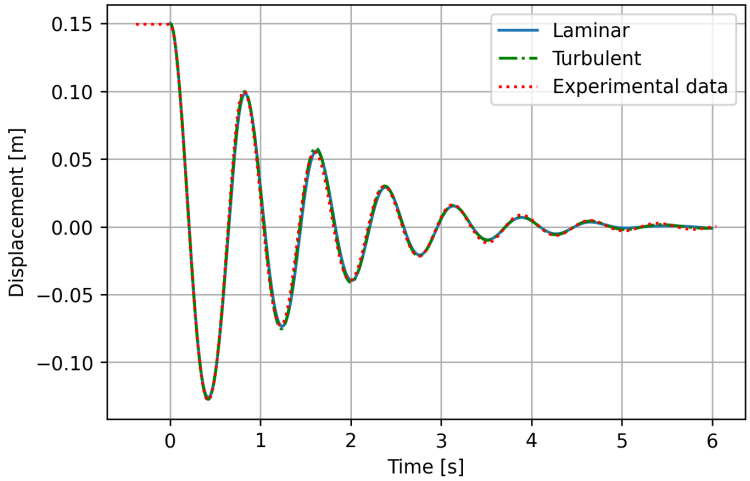


Figure 4.8: *Comparing displacement of the sphere between simulation F.1, F.3 and the experiment.*

The dampening coefficients for simulation F.3 and O.3 is shown in Table 4.5. It can be seen that the difference is very small for both the turbulent simulation in OpenFOAM and FINE™/Marine.

Table 4.5: Evaluation of the dampening coefficient for the turbulence simulations.

	θ	difference	difference in %
Experiment	0.774	-	-
F3	0.7297	0.0443	5.72
O3	0.7612	0.0128	1.65

Table 4.6 shows the number of cores assigned to simulation F.3 and O.3. as well as the computational times for these cases.

Table 4.6: Number of cores and computational times for the turbulence simulations, F.3 and O.3.

	Computational time [h]	Number of cores	Core hours [h]
O.3	13.83	20	276.8
F.3	7.72	32	247.2

To be able to further analyse the effect of turbulence on the simulations the force on the sphere is calculated over the entire simulation. The result for the laminar simulation in OpenFOAM O.1 can be seen in Figure 4.9, showing gravity (F_G), buoyancy (F_B), and the dynamic forces. Positive forces are defined in the negative z direction and negative forces are defined in the positive z direction. As shown in Figure 4.9 the gravity and buoyancy forces are the dominating forces in the simulation. The dynamic forces peaks at 0.4 seconds and 0.8 seconds, this is at the lowest and highest heave motion respectively.

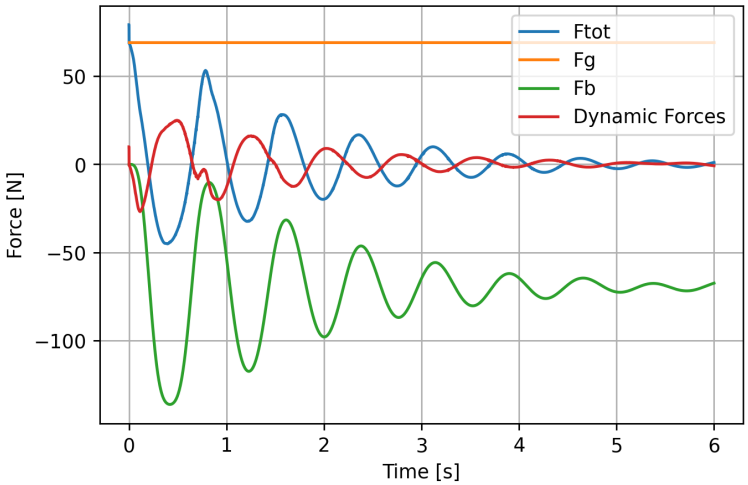


Figure 4.9: *Forces affecting the sphere in the O.1 simulation.*

The same forces are also calculated in simulation O.3 to be able to compare the forces in the laminar and turbulent simulations, the results for O.3 are shown in Figure 4.10. The figure shows that the gravity and buoyancy forces are dominating in the turbulent case as well. The dynamic also peaks at around 0.4 and 0.8 seconds, this is at the minimum and highest oscillation respectively.

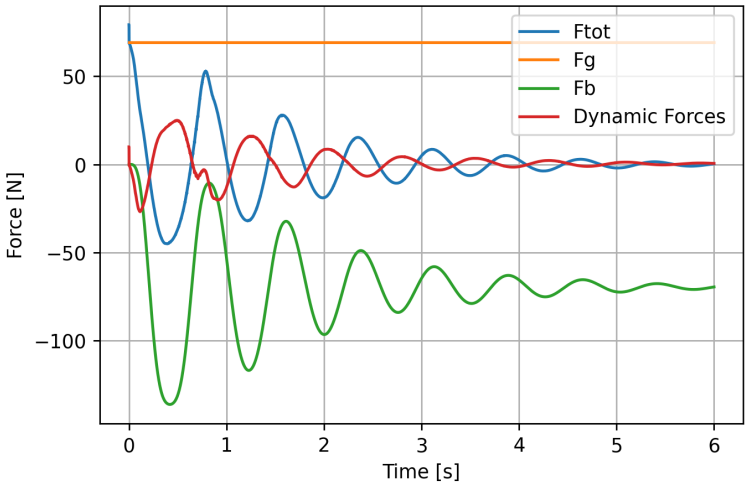
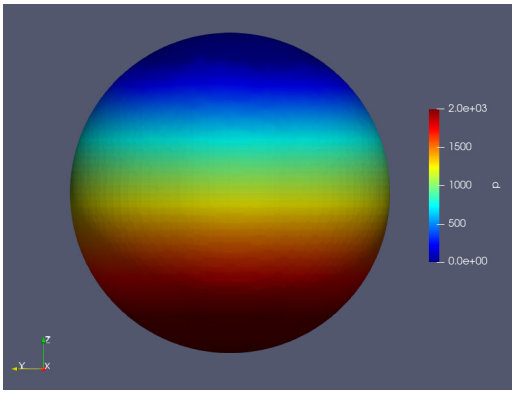


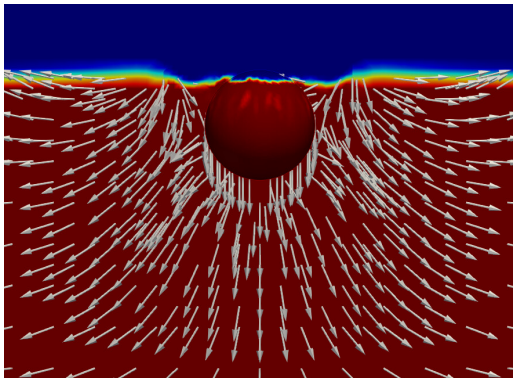
Figure 4.10: *Forces affecting the sphere in the O.3 simulation.*

To be able to analyse the forces both qualitatively and quantitatively contour plots of the pressure on the sphere is included at 0.4 seconds and 0.8 seconds for both simulation O.1 and O.3. The velocity vectors for

the simulations are also plotted together with the colour function in order to see how the water moves in the simulation. In Figure 4.11 this is shown at 0.4 seconds for simulation O.1. It can be seen in the figure that the pressure is the highest at the bottom of the sphere and then decreases towards the top of the sphere. In Figure 4.11b it is shown that the water is moving towards the bottom of the pool right under the sphere. Here the sphere is almost completely submerged in the water.



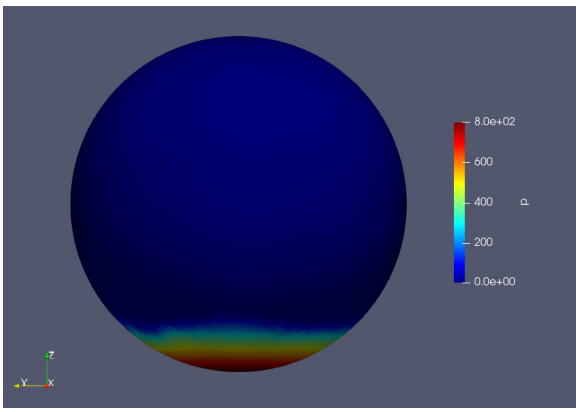
(a) *Contour plot of the pressure on the sphere.*



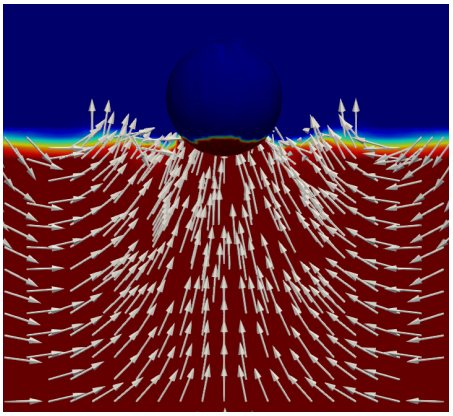
(b) *Contour plot of the colour function together with velocity gradients of the water.*

Figure 4.11: *Contour plot of pressure and contour plot of the colour function with velocity gradients at 0.4 seconds in simulation O.1*

In Figure 4.12 the same contour plots are shown for simulation O.1 at 0.8 seconds. In Figure 4.12a there is only significant pressure at the bottom of the sphere. In Figure 4.12b it is shown that only the bottom of the sphere is in contact with the water. In this figure the water velocity gradients are also shown to point in the direction that the sphere is moving. Water is also flowing from the edges towards the middle to take the place of the water that is moving upwards.



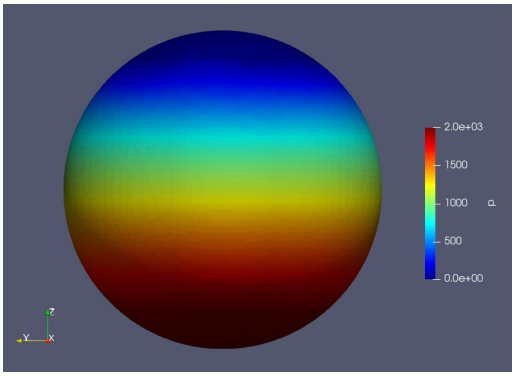
(a) *Contour plot of the pressure on the sphere.*



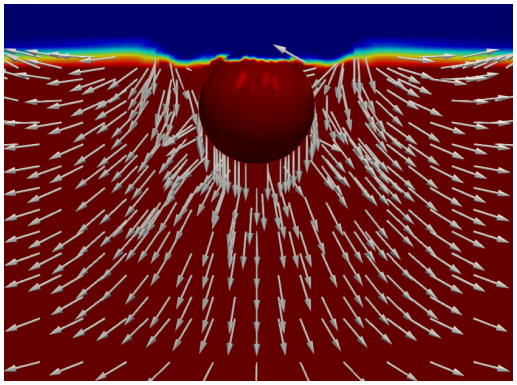
(b) *Contour plot of colour function together with velocity gradients of the water.*

Figure 4.12: *Contour plot of pressure and contour plot of colour function with velocity gradients at 0.8 seconds in simulation O.1.*

To be able to compare the laminar and turbulent cases the same information is displayed for the turbulent case O.3. Figure 4.13 shows the contour plot for the pressure alongside with a contour plot of the colour function with the velocity gradients of the water at 0.4 seconds. Figure 4.13a shows a very similar result as Figure 4.11a. The pressure is at its highest at the bottom of the sphere and then decreases towards the top of the sphere. Both simulations has a pressure of around 2 kPa at the bottom of the sphere. Figure 4.13b also shows a very similar result to the laminar case O.1. The water is moving towards the bottom of the pool right under the sphere and the sphere is almost completely submerged.



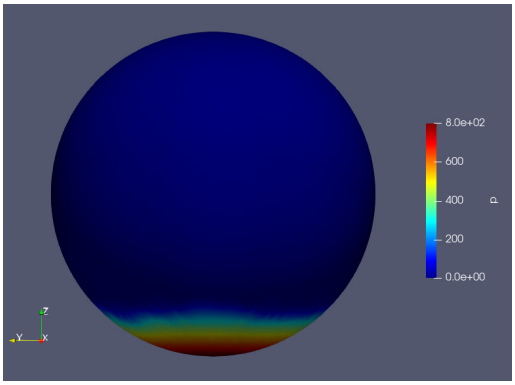
(a) *Contour plot of the pressure on the sphere.*



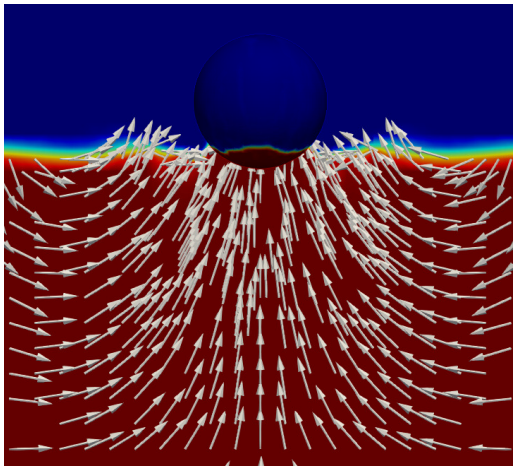
(b) *Contour plot of colour function together with velocity gradients of the water.*

Figure 4.13: *Contour plot of pressure and contour plot of the colour function with velocity gradients at 0.4 seconds in simulation O.3.*

Figure 4.14 shows the same contour plots for simulation O.3 at 0.8 seconds. The result in the Figure has also a very similar result as the laminar case O.1. In Figure 4.13a it is shown that the pressure is at its highest where the sphere touches the water. Figure 4.14b The water moves up from the bottom of the pool towards the sphere which is almost above the water.



(a) *Contour plot of the pressure on the sphere.*



(b) *Contour plot of colour function together with velocity gradients of the water.*

Figure 4.14: *Contour plot of pressure and contour plot of the colour function with velocity gradients at 0.8 seconds in simulation O.3.*

Figure 4.15 shows the average y^+ value in the first grid point towards the sphere over the entire simulation O.3. It can be seen in the figure that y^+ initially starts at almost zero and then goes up to around 65 at 0.8 seconds. y^+ then has a decreasing trend towards the end of the simulation.

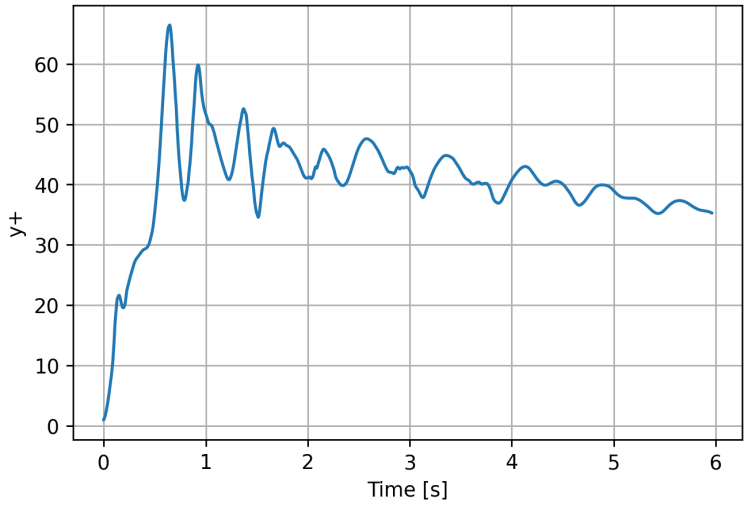


Figure 4.15: y^+ *in the first grid point on the sphere in simulation O.3.*

4.4 Overset mesh, five meter radius sphere results

The heave motion for simulations of the five meter radius sphere is tracked for 40 seconds so the result can be compared with the numerical results described in Section 3.4. This is done in order to show that the model easily can be scaled up and still be accurate and stable. The results from FINE™/Marine and OpenFOAM are plotted alongside with the results from the linear, weakly nonlinear, and fully nonlinear numerical models. The final plot can be seen in Figure 4.16. As shown in the figure, simulations F.6 and O.6 corresponds most with the fully nonlinear models. The fully nonlinear models are CFD simulations made by other groups.

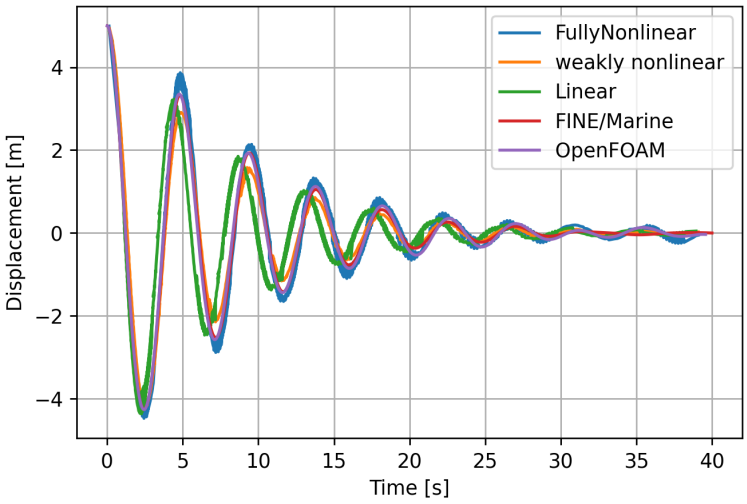


Figure 4.16: *Comparing displacement of the sphere between simulation O.6, F.6 and the numerical models described in section 3.4.*

In Table 4.7 the computational times for both simulation O.6 and F.6 are given.

Table 4.7: Number of cores and computational times for simulation O.6 and F.6.

	Computational time [h]	Number of cores	Core hours [h]
O.6	88.0	20	1760
F.6	28.3	32	905.6

In Figure 4.17 the result is shown for both the turbulent simulations F.7, O.7 and the laminar simulations F.6, O.6. The heave motion of the sphere is shown for 40 seconds. As seen in the figure the result for the turbulent cases and laminar cases are similar.

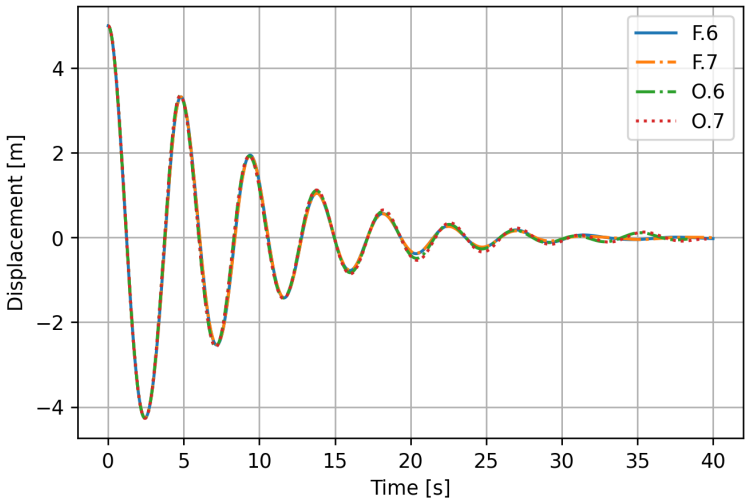


Figure 4.17: *Comparing displacement of the sphere between simulation F.6, O.6, F.7 and O.7.*

Table 4.8 shows the assigned numbers of cores, computational times and amount of core hours for simulation O.7 and F.7.

Table 4.8: Number of cores and computational times for simulation O.7 and F.7.

	Computational time [h]	Number of cores	Core hours [h]
O.7	92.7	20	1854
F.7	29.1	32	932.2

To evaluate the irregularities at the very end for simulation O.6 and O.7, a contour plot is taken of the wave elevation. Figure 4.18 shows the waves for simulation O.6 at 30 seconds into the simulation. This shows the water surface around the sphere and the colour represent the distance from the initial water level. Here waves can be seen moving both towards and away from the sphere.

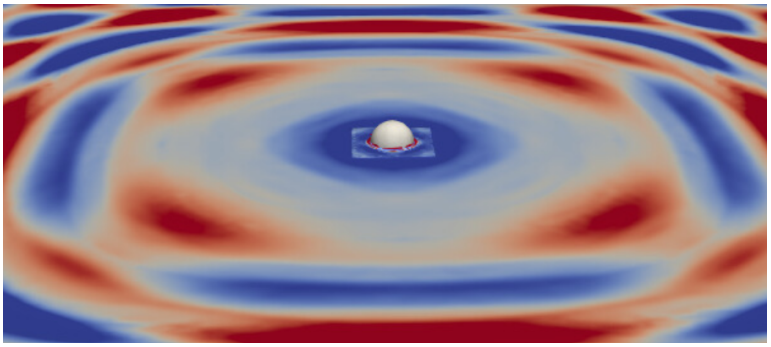


Figure 4.18: *Propagating waves after 30 seconds for simulation O.6.*

Figure 4.19 shows how the average y^+ varies over time for simulation O.7. It can be seen in the figure that y^+ almost starts at zero then increases very quickly up to 80. y^+ then decreases towards the end of the simulation as the movement of the sphere decreases.

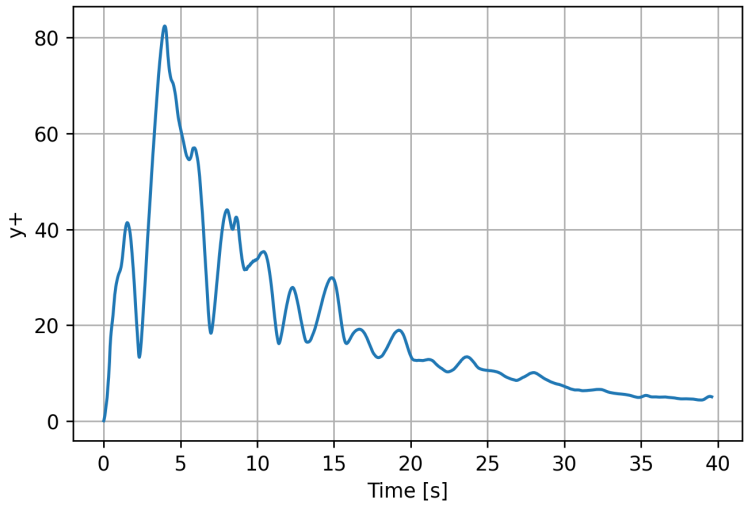
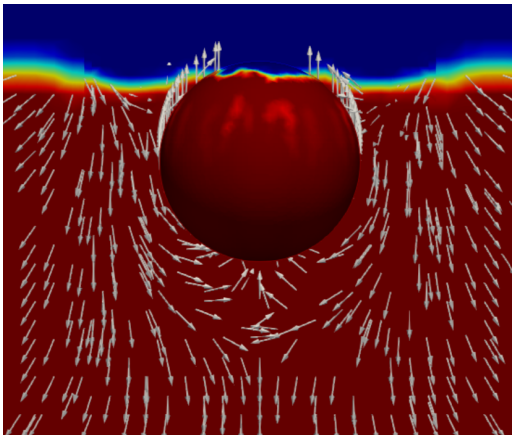
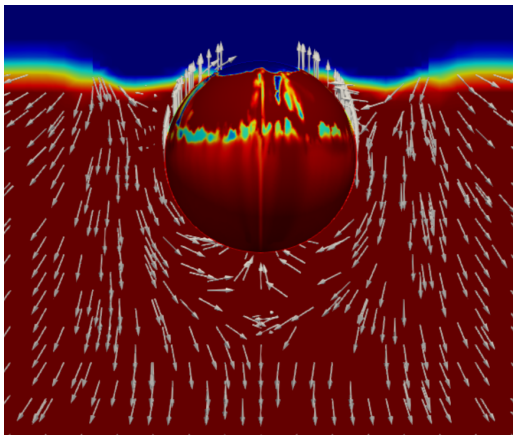


Figure 4.19: y^+ *in the first grid point on the sphere in simulation O.7.*

Figure 4.20 shows the contour plot for the colour function with the velocity gradients at 2.5 seconds for simulation O.6 and O.7. It can be seen that there is some difference between the laminar and turbulent case. In the turbulent case there is some air close to the sphere while the laminar case has almost only water. It can be seen that there is some air and water mixed at the sphere surface in the laminar case.



(a) *Contour plot of the colour function with the velocity gradients of the water.*



(b) *Contour plot of the colour function with the velocity gradients of the water.*

Figure 4.20: *Contour plot of the colour function with the velocity gradients of the water at 2.5 seconds for simulation O.6 and O.7.*

Table 4.9 shows the dampening coefficients and the difference for the heave motion simulations of the five meter sphere. It can be seen in the table that all the simulations has a similar dampening coefficient compared to the numerical data from the fully nonlinear models.

Table 4.9: Evaluation of the dampening coefficients for the five meter sphere.

	θ	difference	difference in %
Numerical	0.1301	-	-
O6	0.1250	0.0051	3.92
F6	0.1235	0.0066	5.07
O7	0.1262	0.0039	3.00
F7	0.1229	0.0072	5.53

4.5 Overset mesh, incoming waves results

Figure 4.21 shows the displacement in meters for the center of the sphere during the 20 seconds long wave simulation. In Figure 4.21 it can be seen that the first wave reaches the sphere after around two seconds. The first wave generates a very small heave motion, this is because the incoming waves is not fully developed until after two seconds because of the ramping time. The wave then has a positive displacement of around 0.01 meters and a negative displacement of around 0.01 meters. This is equal to the wave height which is defined as 0.02 meters.

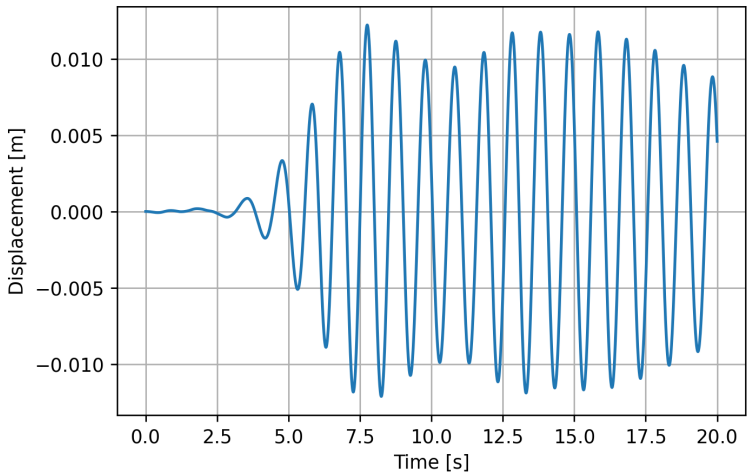


Figure 4.21: *Displacement of the sphere for the incoming wave case, simulation O.8.*

The number of cores and the computational time for simulation O.8 are presented in Table 4.10.

Table 4.10: Number of cores and computational times for simulation O.8

	Computational time [h]	Number of cores	Core hours [h]
O.8	128.6	20	2572

In Figure 4.22 the accumulated potential energy production is shown. The accumulated potential energy is calculated by looking at the potential energy loss of the sphere while it is moving in negative z direction in the simulation. It can be seen in Figure 4.22 that the potential energy production is very low in the beginning of the simulation then increases to a steady rate at around seven seconds. When the sphere is moving in the positive z direction the energy increase is zero because its only considered that the energy can be extracted when the sphere is moving in the negative z direction.

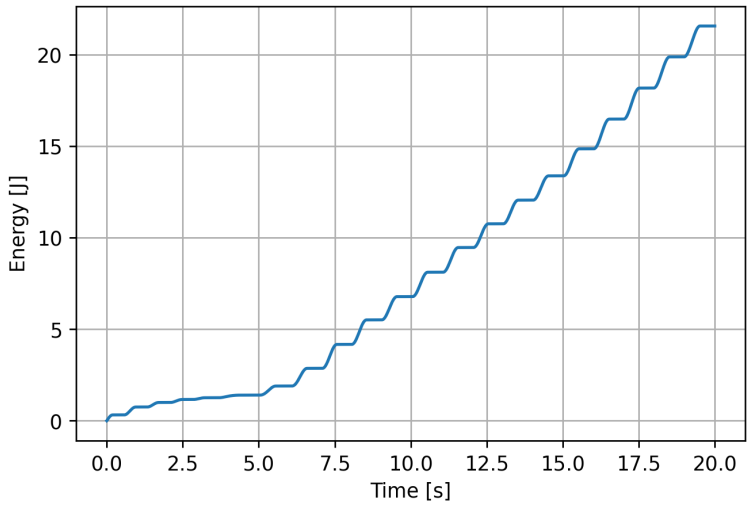


Figure 4.22: *Accumulated potential energy production for simulation O.8.*

5 Discussion

As seen in Section 4.1 the overset mesh method produce the most accurate results compared with the experimental data of the mesh methods in both FINE™/Marine and OpenFOAM. This can be seen in Table 4.1 where the dampening coefficient has the lowest difference for the overset mesh method compared to the experimental. The deforming mesh technique can't describe the dampening of the heave motion correctly towards the end of the simulation in both software. Figure 4.4 shows the colour function with the cells in the mesh in the background. The deformation of the cells along the water interface can clearly be seen in the figure. The water air interface is clearly smeared out where the deformation has occurred. It can be seen in the figure that in the area were the deformation of the mesh occurs the water-air interface becomes larger then a single cell size. A smeared out interface around the sphere leads to that the buoyancy force is calculated incorrectly and the final resting positioning of the sphere can become inaccurate. This could be the reason why the deforming mesh simulation differs from the experimental result in both FINE™/Marine and OpenFOAM towards the end of the simulations which can be seen in Figure 4.1.

The sliding mesh technique simulation was quite accurate in both FINE™/Marine and OpenFOAM as seen in Figure 4.2. The result differs slightly from the experimental data but not significantly. This imposes that the sliding mesh technique can describe heave motion of the falling sphere in an accurate and stable manner. The small differences from the experimental data could maybe be reduced by a further investigation of the cell size in the mesh. A smaller cell size or different approach to the refinements around the sphere could generate more accurate results. This could be investigated in futures studies in order to get more knowledge of the accuracy of the sliding mesh technique for similar projects. Since the mesh move over the water interface the refinement along this region becomes very important, a too large cell size here can smear out the interface. There can also be issues if a larger sphere is used or a larger drop height is implemented because this creates larger ripples and moving waves along the water surface. As written in the theory for the sliding mesh Section 2.12 the mesh technique can have troubles with describing the free stream in the simulation if there is large moving waves in the simulation. However, the mesh technique has a very promising potential but further investigations of the technique are needed in order to determine if it can describe the free stream accurately for example in an incoming wave case.

The overset mesh method has very good prerequisites to produce accurate results. Figure 4.3 shows that the result for the overset mesh is very similar to the experimental data in both OpenFOAM and FINE™/Marine. Neither the static mesh or the moving foreground mesh has any deformation of the cells in the simulation which can lead to numerical errors. The waves created on the water surface by the heave motion of the sphere can be resolved accurately because the static background mesh doesn't move. However it is very important to know the mesh requirements of the method in order for the method to produce accurate results. If the cell size of the background mesh is not matched with the cell size of the moving foreground mesh, some numerical errors can be imposed. This means that it is important to keep in mind the movement of the foreground mesh so the background mesh has the right cell size during the entire motion. The overset mesh has shown to be the most accurate option for describing the heave motion of the 0.15 meter radius sphere and is therefore recommended to use this mesh technique for similar projects.

In Figure 4.5 as well as in Table 4.4 the result for the mesh study in OpenFOAM is shown.The difference for the dampening coefficient and the position becomes smaller for the finer meshes. This shows that the cell size in the mesh is very important for how accurate the simulation is and that the medium mesh is a good and stable mesh. The amount of core hours in the different simulations can be seen in Table 3.9. The result in the table shows clearly that a higher amount of cells leads to a higher computational time. Therefore it is important to make a mesh study to also ensure that not more cells then necessary is used to minimize the computational time. The result for the mesh study in FINE™/Marine is shown in Figure 4.6 as well as in Table 4.4, the coarse, medium, fine mesh has a very similar results. There is a small difference between the coarse and the medium mesh but not much. This means that a little coarser mesh could be used but the result clearly shows that the medium mesh is stable and accurate. In Table 3.16 the core hours for the different meshes is shown, also here it can be seen that there is a high computational difference between the meshes.

When it comes to the computational time, the overset mesh method is clearly the most computational heavy method which can be seen in Table 4.2. This is both due to that the method requires two meshes which leads to a larger amount of cells, the method also requires an interpolation between the meshes during each iteration which also increase computational time. The amount of core hours in FINE™/Marine and OpenFOAM differs a bit for the overset mesh method, FINE™/Marine has a lower amount of core hours. When comparing the computational time for the two programs it has to be considered that the simulations are calculated on different

computational clusters. It is more then the amount of core hours that can be compared when looking at the simulation time but the amount of core hours gives a rough estimation of the simulation time.

The result for the turbulence simulations for the heave motion of the 0.15 meter radius sphere can be seen in Section 4.3. The turbulence simulations are very similar to the laminar simulations in both FINE™/Marine and OpenFOAM, which can be seen if Figure 4.8 and Figure 4.7 are compared. The turbulence around the sphere and in the free stream is not affecting the heave motion significantly since the turbulence simulation and the laminar simulation produce very similar result and are close to the experimental data. The dampening coefficients are also very similar which can be seen if Table 4.1 and Table 4.5 are compared.

The forces in the turbulent simulation O.3 can be seen in Figure 4.10 and the forces for the laminar simulation O.1 is shown in Figure 4.9. The forces are similar in the whole simulation. It can be seen that the size of the buoyancy force is varying over the simulation depending on how much of the sphere is submerged in the water. The gravity force is constant over the whole simulation and towards the end of the simulation the gravity force and the buoyancy force are both around 65 N, which is when the sphere is resting half submerged at the resting water surface. The dynamic forces in the simulation never exceeds 25 N in the simulation. This is smaller compared to both gravity and buoyancy which are the dominating forces in the simulation. Turbulence does not have much affect on the heave motion of the sphere since buoyancy and gravity which is the dominating forces are not affected by the turbulence. The dynamic forces which are affected by the turbulence are not changing the simulation very much because of the small size of the force. This implies that if the boundary layer around the sphere is changed, the heave motion in the simulation is not affected due to that gravity and buoyancy forces still are the dominating forces.

Both the velocity gradients and the pressure around the sphere can be seen for simulation O.1 and O.3 at 0.4, 0.8 seconds respectively in Section 4.3. At 0.4 seconds the sphere is almost fully submerged in the water. The pressure on the sphere is at its highest at the bottom of the sphere but the pressure is significant up to around where the water line touches the top of the sphere. It is clear that there the pressure gets higher on the parts of the sphere that is submerged in the water. A higher pressure on the sphere also means that larger forces are affecting the sphere. If one compares Figure 4.9 and 4.11a it can be seen that the pressure on the bottom of the sphere is about 2 kPa and the separate forces is at its highest on the sphere at 0.4 seconds. At 0.8 seconds the pressure at the bottom of the sphere is 0.8 kPa while the separate forces are lower compared to at 0.4 seconds which can be seen in Figure 4.12a and 4.10. Since higher pressure means that larger forces are acting on the sphere its very logical that the pressure and forces are higher at 0.4 seconds then at 0.8 seconds.

The water flow is also very similar for both the laminar and turbulent simulations which can be seen when comparing the simulations at 0.4 seconds, which is shown in Figure 4.11b and 4.13b. The water is pushed away from the sphere towards the bottom of the pool and towards the edges close to the surface. This is because the sphere takes up space in the water when it is submerged and the water therefore needs to find new space to take up and is pushed away. At 0.8 seconds the water is moving towards the sphere with very similar velocity gradients for both the laminar simulation O.1 and turbulent simulation O.3. The sphere is almost completely out of the water leaving new space for the water to take up and the water is therefore moving towards the sphere.

The y^+ values for simulation O.3 can be seen in Figure 4.15. y^+ varies between 0 to 65 in the simulation. It increases in the beginning of the simulation when the speed of the sphere increases, then decreases over time in the simulation because of the lower and lower velocities. k-ω-SST model can handle to fully resolve the turbulent flow around the sphere for values of y^+ below around 5. For y^+ above 5 the k-ω-SST model uses wall functions in order to model the turbulent values close to the sphere. This means that the viscous sub layer around the sphere is not resolved for most of the simulation. However due to that the turbulence has a very low impact on the simulation it is not affecting the heave motion noticeably that the boundary layer is not fully resolved. If further studies would be made where the free stream is of interest, the grid around the sphere should be refined in order to fully resolve the boundary layer around the sphere.

There is one hint of turbulent behaviour for the water flow if Figures 4.11b and 4.13b are compared. For the turbulent simulation O.3 it can be seen that the water has a bit more splash above the sphere. However this difference is very small and does not affect the heave motion of the sphere noticeably. However if the water flow is the interesting part in the simulation rather then the heave motion the k-ω-SST turbulence model could be worth considering to capture the small turbulence effects in the flow.

The simulation core hours for the laminar simulations F.1, O.1 are a bit lower then for the turbulent simulations F.3, O.3 which can be seen in Table 4.6. The same mesh, domain and setup is used for the laminar and turbulent simulation with only an addition of the turbulence model k-ω-SST. The addition of the turbulence model leads to that the simulation becomes more computational heavy because of the extra

equations that need to be solved for the turbulence. Therefore it is very logical that there is an increase in the amount of core hours for the simulations in both FINE™/Marine and OpenFOAM.

The results for the five meter radius sphere heave motion simulation is presented in Section 4.4. In this section Figure 4.16 presents the results for the heave motion of the sphere in simulation F.6 and O.6 compared to the numerical data described in Section 3.4. The results show that there is a strong correlation between the simulation F.6 and O.6 and the numerical reference data, both for the OpenFOAM and FINE™/Marine case. The dampening coefficients for these simulations also show there is an insignificant difference between the numerical fully nonlinear data and the simulations which can be seen in Table 4.9.

The numerical reference data shown in Figure 3.3 shows the result for the heave motion of the five meter radius sphere. Both linear, weakly nonlinear and fully nonlinear models are used as numerical models. These models have different numerical natures and therefore generates different results. However the different methods corresponds quite well to one another. The most interesting numerical model to look at for this project is the fully nonlinear models as these are results from CFD Simulations. There is some variation in the data for the fully nonlinear models which can be expected due to the size of the sphere and the drop height. This means there are large forces present in the systems and small variations between the simulations can therefore have large impacts on the heave motion and on the system. In Figure 4.16 the laminar simulations in OpenFOAM and FINE™/Marine for the five meter radius sphere are plotted together with the numerical results described in Section 3.4. From the figure it can be seen that the results from simulations O.6 and F.6 correspond well with the numerical data. The simulation result is closest to the fully nonlinear numerical models which is expected since this is CFD codes as well. This can not be seen as a true validation of the CFD simulations since the result is compared with numerical data and not experimental. However it generates credibility to the simulations for the five meter sphere since it generates similar results as other numerical models with a variety of numerical natures.

Simulations O.1 and O.6 have the same case set up but two different domains and size of the spheres. Both simulations are accurate if they are compared with experimental and numerical data respectively. This shows that the simulation method is stable and the geometry easily can be scaled up.

In Figure 4.17 simulation result for the heave motion is shown for both the laminar and the turbulence simulations for the five meter sphere. A k-ω-SST model is used for the turbulent simulations to be able to describe the turbulence effects both close to the walls and in the free stream of the simulation. It can be seen in the figure that the turbulence has little effect on the heave motion. The position of the center of the sphere is very similar for both the turbulent and laminar simulations during the entire simulation. This implies that the turbulence has very little effect on the heave motion for the sphere even when it's scaled up. In Figure 4.9 the forces for simulation O.1 is shown which is a laminar simulation with the 0.15 meter radius sphere. When the sphere is scaled up both the gravity and the buoyancy forces increase on the sphere as well as the dynamic forces. However the gravity and buoyancy forces probably still are the dominating forces in the simulation. The same conclusion as for the small sphere is then very likely, the dynamic forces are small in the system compared to the gravity and buoyancy force which leads to that the small effect of the turbulence.

Figure 4.19 shows how the y^+ value varies over the turbulent simulation O.7. y^+ starts at almost zero in the simulation and then it increases very quickly up to around 80. As the motion of the sphere becomes smaller y^+ becomes smaller in the simulation. This is a very similar behaviour as y^+ in the turbulent simulation for the small sphere O.3.

The computational time for the turbulent simulations of the five meter radius sphere is shown in Table 4.8. There it can be seen that FINE™/Marine has a lower amount of core hours than OpenFOAM. However this difference can depend on many parameters such as how the different computational clusters uses their cores and how much data that is saved in the simulation. If one compare the computational times for the turbulent and laminar simulations which are presented in Table 4.8 and 4.7 respectively. It can be seen that the addition of the k-ω-SST turbulence model leads to a higher amount of core hours in both FINE™/Marine and OpenFOAM. This is expected because the addition of a turbulence model leads to that extra equations need to be solved in each iteration, which of course is more computational heavy then a laminar simulation.

In Figure 4.17 the result between the software can also be compared. It shows almost no variation between the software except after 30 seconds, here the OpenFOAM simulations start to move up and down again. This is unrealistic as something would need to create this motion in the sphere. From analysing Figure 4.18 that shows the O.6 case at 30 seconds. It can be concluded that waves generated by the heave motion of the sphere are shown to have propagated all the way to the end of the domain, and started to move back towards the center. This is probably the cause of the motion at the end of the simulation. To avoid this problem the outer dimensions could either be extended or the boundary conditions could be changed to absorb the wave similarly

to how the O.8 wave simulation absorbs the wave at the end of the domain. This motion at the end of the simulation can't be seen in simulation F.7. This is because the boundary condition far field with constant values in Fine™/Marine is dampening the waves a bit while the boundary condition fixedValue in OpenFOAM doesn't dampen the wave as much, which leads to that they travel back to the sphere.

The heave motion of the sphere for the incoming wave case can be seen in Figure 4.21. The displacement starts at around two seconds but is very small in the beginning. This is because of the specified ramp time of two seconds in OpenFOAM. The wave period of the wave is one second and the wavelength is 1.55 meters which can be seen in Table 3.11. The length from the wave inlet to the sphere is two wavelengths which can be seen in Figure 2.2, this means it takes two seconds for the first wave to reach the sphere. This agrees very well with the result because it can be seen in Figure 4.21 that the sphere starts to move significantly at two seconds. This means that fully developed waves reach the sphere at four seconds but it can be seen that the sphere doesn't reach its maximum heave motion until around seven seconds. The motion of the sphere is also not constant but vary over time even though the waves are regular steady waves. The heave motion of the sphere has a mean value of around 0.02 meters between seven to 20 seconds in the simulation. This is the same value as the height of the waves which means the sphere's center of mass follows the wave quite well. The wave period of the wave is one second and the period of the heave motion of the sphere is also around one second, which can be seen by looking at the time between the peaks in Figure 4.21.

The accumulated potential energy outtake is shown for the heave motion of the sphere in Figure 4.22. The potential energy outtake is very small in the beginning because of the small motion of the sphere. If Figure 4.22 and 4.21 are compared, it is seen that the potential energy outtake increases as the size of the spheres motion increases. The energy outtake from the sphere is quite small, about 22 J over 20 seconds. However the sphere is very small, it has a diameter of 0.3 meters and only weighs seven kilograms. The wave is also rather small it has a wave height of 0.02 meters. A wave energy converter that would be placed in the ocean both need to be larger and placed at a location with higher waves. This would increase the potential energy production significantly since both the mass and the height would increase, the large five meter radius sphere in this project weighs 261.8 tons so there is definitely potential to produce more energy. Of course there would be a some energy losses in a wave energy converter and all the energy can not be converted to electricity which has to be considered. However simulation O.8 shows the potential to use CFD for calculations and designs of wave energy converters. The energy converter could produce energy by an mechanical resistance that converted the mechanical force to electricity. This resistance can easily be implemented in the simulations in order to get more realistic design simulations.

For future studies it would be interesting to set up a similar wave simulation in Fine™/Marine. Due to the time restriction in this project we didn't manage to set up a converging wave simulation in Fine™/Marine.

In this project the final design of the wave energy converter is not considered. The main aim of this project is to accurately describe the heave motion of the floating sphere with CFD simulations and develop guidelines for this. However different designs of the object could be more favorable in terms of heave motion. The energy outtake is probably dependant on the range of motion of the floating object in z direction. A different design of the floating object could be favorable for the heave motion. In this project is also just the motion in z direction considered in the simulations. When encountering incoming waves a sphere could gain spin or motions in the x, y direction depending on if it has any components attach to it. This could lead to unnecessary motion which wouldn't produce any energy. A design that prevents these motions could therefore be intersecting to investigate in future studies.

For future studies it would also be interesting to compare the incoming wave simulation O.8 with some experimental data. This would however require both simulation and experimental studies since experimental data for this case has not been produced yet. This would be valuable information since it would validate that CFD could be used for the purpose of estimating the energy extraction in incoming waves. Stokes 2^{st} wave theory is used in simulation O.8. This is because Stokes 2^{st} is very realistic waves that don't break when its moving. However this is a regular steady waves and to describe the state of the ocean irregular waves has to be used. The height and shape of the wave is extremely important for the motion of the floating object. It is therefore very important that the waves are represented correctly.

6 Conclusion

This project shows that the overset mesh method is an accurate and stable meshing method for CFD modeling of WEC. The model is very computational heavy which makes the choice of both domain size and cell size in the mesh very important. The simulation setups for the overset mesh method described in section 3.7.1 and 3.7.2 has shown to be accurate and stable. These setups would therefore be recommended for CFD modeling of a WEC in OpenFOAM and FINE™/Marine respectively. Sliding mesh technique has a promising potential and the technique has a low computational time but further investigations is needed for this technique.

The project shows that turbulence is not affecting the heave motion for the sphere in this project. However if the flow around the object is of interest a k-ω SST model could be of value, as the turbulence has some small effects on the flow around the sphere.

The setup overset mesh technique is expected to work well for incoming wave simulations in OpenFOAM, but experimental data is needed in order to validate this. Further investigations is needed for incoming wave simulations in both software, however simulation setup 3.7.1 is a good setup to start with in OpenFOAM.

Overall this project shows that there is a promising potential for CFD modeling of WEC but further simulation and experimental studies is needed, especially for incoming wave cases.

References

[1] The World Bank. *Electricity production from coal sources (% of total).* 2015. URL: https://data.worldbank.org/indicator/EG.ELC.COAL.ZS (visited on 01/26/2021).

[2] United Nations Framework Convention on Climate Change. *The Paris Agreement.* URL: https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement (visited on 01/21/2021).

[3] SSPA. *Energy from the ocean-predictable, reliable, renewable.* URL: https://www.sspa.se/what/ocean-energy-energy-from-the-ocean-predictable-reliable-renewable (visited on 01/22/2021).

[4] OpenEI. *Wave energy.* URL: https://openei.org/wiki/Wave_Energy (visited on 01/22/2021).

[5] F. Wendt, K. Nielsen, Y.-H. Yu, H. Bingham, C. Eskilsson, M. Kramer, A. Babarit, T. Bunnik, R. Costello, S. Crowley, B. Gendron, G. Giorgi, S. Giorgi, S. Girardin, D. Greaves, P. Heras, J. Hoffman, H. Islam, K.-R. Jakobsen, C.-E. Janson, J. Jansson, H. Y. Kim, J.-S. Kim, K.-H. Kim, A. Kurniawan, M. Leoni, T. Mathai, B.-W. Nam, S. Park, K. Rajagopalan, E. Ransley, R. Read, J. V. Ringwood, J. M. Rodrigues, B. Rosenthal, A. Roy, K. Ruehl, P. Schofield, W. Sheng, A. Shiri, S. Thomas, I. Touzon, and I. Yasutaka. Ocean Energy Systems Wave Energy Modelling Task: Modelling, Verification and Validation of Wave Energy Converters. *Journal of Marine Science and Engineering* **7**.11 (2019). ISSN: 2077-1312. DOI: 10.3390/jmse7110379. URL: https://www.mdpi.com/2077-1312/7/11/379.

[6] B. Andersson, R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo, and B. van Wachem. *Computational Fluid Dynamics for Engineers.* 14th ed. Cambridge University Press, 2018.

[7] Z. Shen, D. Wan, and P. M. Carrica. Dynamic overset grids in OpenFOAM with application to KCS self-propulsion and maneuvering. *Ocean Engineering* **108** (2015), 287–306. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2015.07.035. URL: https://www.sciencedirect.com/science/article/pii/S0029801815003480.

[8] OpenCFD Ltd. *OpenΔFOAM, The open source CFD toolbox.* URL: https://www.openfoam.com/ (visited on 03/11/2021).

[9] P.-H. Musiedlak. NUMERICAL MODELLING OF RESPONSES OF OFFSHORE WAVE ENERGY CONVERTERS IN EXTREME WAVES (2019). URL: https://pearl.plymouth.ac.uk/handle/10026.1/15115.

[10] B. L. Méhauté. *An Introduction to Hydrodynamics and Water Waves.* New York:Springer-Verlag, 1976.

[11] Kraaiennest. *Water wave theories.* URL: https://commons.wikimedia.org/wiki/File:Water_wave_theories.svg (visited on 03/30/2021).

[12] NUMECA International. *NUMECA Online Documentation Platform.* Version EN202011141652. Brussels, Belgium, 2021.

[13] F. Wendt, K. Nielsen, Y.-H. Yu, H. Bingham, C. Eskilsson, M. Kramer, A. Babarit, T. Bunnik, R. Costello, S. Crowley, B. Gendron, G. Giorgi, S. Giorgi, S. Girardin, D. Greaves, P. Heras, J. Hoffman, H. Islam, K.-R. Jakobsen, C.-E. Janson, J. Jansson, H. Y. Kim, J.-S. Kim, K.-H. Kim, A. Kurniawan, M. Leoni, T. Mathai, B.-W. Nam, S. Park, K. Rajagopalan, E. Ransley, R. Read, J. V. Ringwood, J. M. Rodrigues, B. Rosenthal, A. Roy, K. Ruehl, P. Schofield, W. Sheng, A. Shiri, S. Thomas, I. Touzon, and I. Yasutaka. Ocean Energy Systems Wave Energy Modelling Task: Modelling, Verification and Validation of Wave Energy Converters. *Journal of Marine Science and Engineering* **7**.11 (2019). ISSN: 2077-1312. DOI: 10.3390/jmse7110379. URL: https://www.mdpi.com/2077-1312/7/11/379.

[14] F. R. Menter. Review of the shear-stress transport turbulence model experience from an industrial perspective. *International Journal of Computational Fluid Dynamics* **23**.4 (2009), 305–316. DOI: 10.1080/10618560902773387. eprint: https://doi.org/10.1080/10618560902773387. URL: https://doi.org/10.1080/10618560902773387.

[15] OpenCFD Ltd. *OpenΔFOAM - The Open Source CFD Toolbox - Extended Code Guide.* English. Version v2006. OpenCFD Limited. 2020. URL: https://www.openfoam.com/documentation/guides/latest/doc/.

[16] S. B. Lee. A study on temporal accuracy of OpenFOAM. *International Journal of Naval Architecture and Ocean Engineering* **9**.4 (2017), 429–438. ISSN: 2092-6782. DOI: https://doi.org/10.1016/j.ijnaoe.2016.11.007. URL: https://www.sciencedirect.com/science/article/pii/S2092678216305465.

[17] H. Jasak. *Finite Volume Discretisation in OpenFOAM Best Practice Guidelines.* Chalmers University, Gothenburg, Faculty of Mechanical Engineering and Naval Architecture. 2015. URL: http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2015/HrvojeJasak/DiscretisationBestPractice.pdf (visited on 05/11/2021).

[18] P. Higuera, J. L. Lara, and I. J. Losada. Realistic wave generation and active wave absorption for Navier–Stokes models: Application to OpenFOAM®. *Coastal Engineering* **71** (2013), 102–118. ISSN: 0378-3839. DOI: https://doi.org/10.1016/j.coastaleng.2012.07.002. URL: https://www.sciencedirect.com/science/article/pii/S0378383912001354.

[19] Cadence. *FINE™/MARINE*. URL: https://www.numeca.com/product/fine-marine (visited on 04/01/2021).

[20] H. Chen, L. Qian, Z. Ma, W. Bai, Y. Li, D. Causon, and C. Mingham. Application of an overset mesh based numerical wave tank for modelling realistic free-surface hydrodynamic problems. *Ocean Engineering* **176** (Feb. 2019), 97–117. DOI: 10.1016/j.oceaneng.2019.02.001.

[21] U. Caliskan and S. Miskovic. A chimera approach for MP-PIC simulations of dense particulate flows using large parcel size relative to the computational cell size. *Chemical Engineering Journal Advances* **5** (2021), 100054. ISSN: 2666-8211. DOI: https://doi.org/10.1016/j.ceja.2020.100054. URL: https://www.sciencedirect.com/science/article/pii/S2666821120300545.

[22] G. Houzeaux, B. Eguzkitza, R. Aubry, H. Owen, and M. Vázquez. A Chimera method for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* **75**.3 (2014), 155–183. DOI: https://doi.org/10.1002/fld.3886. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.3886. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3886.

[23] M. R. Visbal and D. V. Gaitonde. On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes. *Journal of Computational Physics* **181**.1 (2002), 155–185. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.2002.7117. URL: https://www.sciencedirect.com/science/article/pii/S0021999102971172.

[24] H. Jasak. "Dynamic Mesh Handling in OpenFOAM". *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. DOI: 10.2514/6.2009-341. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2009-341. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2009-341.

[25] G. Bachler, H. Schiffermüller, and A. Bregant. "A Parallel Fully Implicit Sliding Mesh Method for Industrial CFD Applications". *Parallel Computational Fluid Dynamics 2000*. Ed. by C. Jenssen, H. Andersson, A. Ecer, N. Satofuka, T. Kvamsdal, B. Pettersen, J. Periaux, and P. Fox. Amsterdam: North-Holland, 2001, pp. 501–508. ISBN: 978-0-444-50673-3. DOI: https://doi.org/10.1016/B978-044450673-3/50129-9. URL: https://www.sciencedirect.com/science/article/pii/B9780444506733501299.

[26] M. B. Kramer, J. Andersen, S. Thomas, F. B. Bendixen, H. Bingham, R. Read, N. Holk, E. Ransley, S. Brown, Y.-H. Yu, T. T. Tran, J. Davidson, C. Horvath, C.-E. Janson, K. Nielsen, and C. Eskilsson. Highly Accurate Experimental Heave Decay Tests with a Floating Sphere: A Public Benchmark Dataset for Model Validation of Fluid–Structure Interaction. *Energies* **14**.2 (2021). ISSN: 1996-1073. DOI: 10.3390/en14020269. URL: https://www.mdpi.com/1996-1073/14/2/269.

A Appendix

A.1 fvSchemes overset mesh

The text file fvSchemes for the overset mesh method is shown here. This contains the schemes that the solver uses to calculate the governing equations for the simulation.

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
  default         Euler;
}

gradSchemes
{
    default       Gauss linear;
}

divSchemes
{
  div(rhoPhi,U)     Gauss vanLeerV;
  div(phi,alpha)    Gauss interfaceCompression vanLeer 1;
  div(phirb,alpha)  Gauss linear;
  div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}

oversetInterpolation
{
    method          cellVolumeWeight;
}

oversetInterpolationSuppressed
{
    grad(p_rgh);
    surfaceIntegrate(phiHbyA);
}

fluxRequired
{
```

```
63   default           no;
64     p_rgh;
65     pcorr;
66     alpha.water;
67 }
68
69 wallDist
70 {
71     method meshWave;
72 }
73
74 // ************************************************************************* //
```

Listing A.1: fvSchemes for O1

A.2 fvSolution overset mesh

A fvSolution dictonary is needed to specify the solver used for the schemes in the simulations. Here is the fvSolution dictonary for the overset mesh method specified.

```
1  /*--------------------------------*- C++ -*----------------------------------*\
2  | =========                 |                                                 |
3  | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
4  |  \\    /   O peration     | Version:  v2012                                 |
5  |   \\  /    A nd           | Website:  www.openfoam.com                      |
6  |    \\/     M anipulation  |                                                 |
7  \*---------------------------------------------------------------------------*/
8  FoamFile
9  {
10     version     2.0;
11     format      ascii;
12     class       dictionary;
13     object      fvSolution;
14 }
15 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
16
17 solvers
18 {
19
20     "cellDisplacement.*"
21     {
22         solver          PCG;
23         preconditioner  DIC;
24
25         tolerance       1e-06;
26         relTol          0;
27         maxIter         100;
28     }
29
30     "alpha.water.*"
31     {
32         nAlphaCorr      3;
33         nAlphaSubCycles 2;//2
34         cAlpha          1;
35         icAlpha         1;//1
36
37         MULESCorr       yes;
38         nLimiterIter    5;
39         alphaApplyPrevCorr  no;
40
41         solver          smoothSolver;
42         smoother        symGaussSeidel;
43         tolerance       1e-8;
44         relTol          0;
45     }
46
47     "pcorr.*"
48     {
49     solver          PCG;
```

```
50      preconditioner
51      {
52        preconditioner  GAMG;
53        tolerance       1e-5;
54        relTol          0;
55        smoother        DICGaussSeidel;
56        cacheAgglomeration no;
57      }
58
59      tolerance       1e-05;
60      relTol          0;
61      maxIter         100;
62    }
63
64      p_rgh
65      {
66          solver          PBiCGStab;
67          preconditioner  DILU;
68          tolerance       1e-9;
69          relTol          0.01;
70      }
71
72      p_rghFinal
73      {
74          $p_rgh;
75          relTol          0;
76      }
77
78      "(U).*"
79      {
80          solver          smoothSolver;
81          smoother        symGaussSeidel;
82          tolerance       1e-08;
83          relTol          0;
84      }
85  }
86
87  PIMPLE
88  {
89      momentumPredictor       no;
90      nOuterCorrectors        3;
91      nCorrectors             2;
92      nNonOrthogonalCorrectors  0;
93
94      ddtCorr                     yes;
95      correctPhi                  no;
96
97      moveMeshOuterCcorrectors   no;
98      turbOnFinalIterOnly     no;
99
100     oversetAdjustPhi        no;
101 }
102
103 relaxationFactors
104 {
105     fields
106     {}
107     equations
108     {
109     ".*"      1;
110     }
111 }
112
113 cache
114 {}
115
116 // ************************************************************************* //
```

Listing A.2: fvSolution for O1

A.3 fvSchemes deforming mesh

Deforming mesh requires a fvSchemes dictionary text file, this text file is given here.

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
  default       Euler;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{

  div(rhoPhi,U)   Gauss vanLeerV;
  div(phi,alpha)  Gauss interfaceCompression vanLeer 1;
  div(phirb,alpha) Gauss linear;
  div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear;

}

laplacianSchemes
{
    default         Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}

oversetInterpolation
{
    method          cellVolumeWeight;
}

oversetInterpolationSuppressed
{
    grad(p_rgh);
    surfaceIntegrate(phiHbyA);
}

fluxRequired
{
    default         no;
    p_rgh;
```

```
67      pcorr;
68      alpha.water;
69  }
70
71  wallDist
72  {
73      method meshWave;
74  }
75
76  // ************************************************************************* //
```

Listing A.3: fvSchemes for O4

A.4 fvSolution deforming mesh

Here is the fvSolution dictionary text file for the deforming mesh case given.

```
1   /*--------------------------------*- C++ -*----------------------------------*\
2   | =========                 |                                                 |
3   | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
4   |  \\    /   O peration     | Version:  v2012                                 |
5   |   \\  /    A nd           | Website:  www.openfoam.com                      |
6   |    \\/     M anipulation  |                                                 |
7   \*---------------------------------------------------------------------------*/
8   FoamFile
9   {
10      version     2.0;
11      format      ascii;
12      class       dictionary;
13      object      fvSolution;
14  }
15  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
16
17  solvers
18  {
19
20      "cellDisplacement.*"
21      {
22          solver          PCG;
23          preconditioner  DIC;
24
25          tolerance       1e-06;
26          relTol          0;
27          maxIter         100;
28      }
29
30      "alpha.water.*"
31      {
32          nAlphaCorr      3;
33          nAlphaSubCycles 2;
34          cAlpha          1;
35          icAlpha         1;
36
37          MULESCorr       yes;
38          nLimiterIter    5;
39          alphaApplyPrevCorr  no;
40
41          solver          smoothSolver;
42          smoother        symGaussSeidel;
43          tolerance       1e-8;
44          relTol          0;
45      }
46
47      "pcorr.*"
48
49      {
50      solver          PCG;
51      preconditioner
52      {
```

```
53        preconditioner  GAMG;
54        tolerance       1e-5;
55        relTol          0;
56        smoother        DICGaussSeidel;
57        cacheAgglomeration no;
58    }
59
60    tolerance       1e-05;
61    relTol          0;
62    maxIter         100;
63  }
64
65    p_rgh
66    {
67        solver          PBiCGStab;
68        preconditioner  DIC;
69        tolerance       1e-9;
70        relTol          0.01;
71    }
72
73    p_rghFinal
74    {
75        $p_rgh;
76        relTol          0;
77    }
78
79    "(U).*"
80    {
81        solver          smoothSolver;
82        smoother        symGaussSeidel;
83        tolerance       1e-08;
84        relTol          0;
85    }
86 }
87
88 PIMPLE
89 {
90    momentumPredictor   no;
91    nOuterCorrectors    3;
92    nCorrectors         2;
93    nNonOrthogonalCorrectors 0;
94
95    ddtCorr                     yes;
96    correctPhi                  no;
97
98    moveMeshOuterCcorrectors no;
99    turbOnFinalIterOnly no;
100
101   oversetAdjustPhi    no;
102 }
103
104 relaxationFactors
105 {
106    fields
107    {
108    }
109    equations
110    {
111            ".*" 1;
112    }
113 }
114
115 cache
116 {}
117
118 // ************************************************************************* //
```

Listing A.4: fvSolution for O4