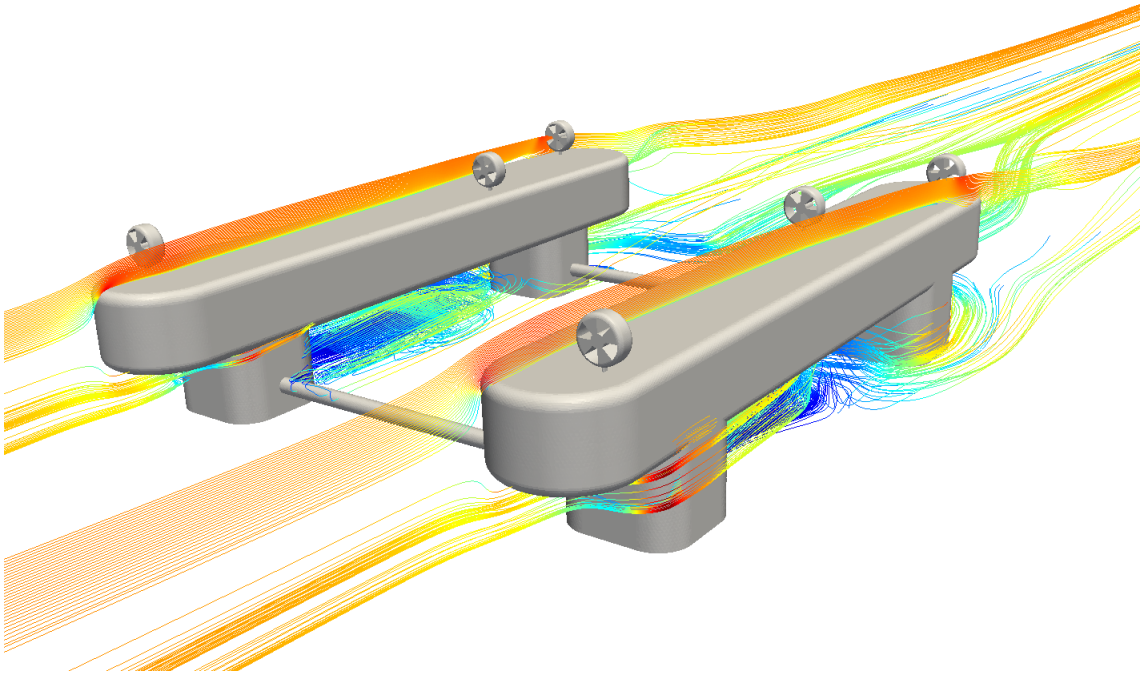
CHALMERS



Simulation of the Flow around a Semi-Submersible using CFD

Master's Thesis in Naval Architecture and Ocean Engineering

JONATHAN ERIKSSON

Department of Shipping and Marine Technology
Division of Marine Technology
Chalmers University of Technology
Gothenburg, Sweden 2017
Master's thesis 2017:X-15/331

Master's thesis 2017:X-15/331

Simulation of the flow around a semi-submersible using CFD

JONATHAN ERIKSSON



Department of Shipping and Marine Technology
Division of Marine Technology
Chalmers University of Technology
Gothenburg, Sweden 2017

Simulation of the flow around a semi-submersible using CFD
JONATHAN ERIKSSON

© JONATHAN ERIKSSON, 2017.

Master's Thesis 2017:X-15/331
Department of Shipping and Marine Technology
Division of Marine Technology
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The hull of a semi-submersible with streamlines colored in terms of velocity.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2017

Simulation of the flow around a semi-submersible using CFD
JONATHAN ERIKSSON
Division of Marine Technology
Department of Shipping and Marine Technology
Chalmers University of Technology

Abstract

The application of Computational Fluid Dynamics (CFD) is almost taken for granted within sectors of engineering dealing with fluid flows. However, its usage within the offshore sector is still limited, primarily due to the large computational times necessary for solving complete sea states. But with increasing computational power and creative new solution methods emerging, the threshold of usability is continuously being lowered. Today the industry relies heavily on scaled model tests; tests which are expensive, time consuming, and may face discrepancies due to scaling effects and unreliable measuring equipment. By utilizing CFD instead, one can analyze the physical properties anywhere within the computational domain, as well as customize and adjust the geometrical and numerical settings at any time. The main objective of this thesis is to investigate the use of CFD simulations for the design of semi-submersible offshore structures.

In this study the submerged section of an offshore structure is the subject of examination. Data available from a previously conducted wind tunnel experiment are used as means of both verification and calibration of the CFD model. Two different CFD software are used in this work; the open-source software OpenFOAM, as well as the commercial software Fluent. Initially a replica of the wind tunnel is modelled and incompressible steady-state Reynolds Averaged Navier-Stokes (RANS) simulations are performed, as it is of interest to find possible instances where steady-state simulations are feasible. Investigations include full geometry and truncated near-field steady-state simulations.

Non-satisfactory results were obtained. The drag force measured in the steady-state simulations reached only a value of 60% of the experimentally measured one. Unsteady simulations, both RANS and LES showed similar results. Reasons for the unsatisfactory results could be insufficient mesh resolution, along with wrong choice of discretization methods and turbulence model.

Keywords: CFD, OpenFOAM, Fluent, semi-submersible, wind tunnel, current loads.

Simulering av flödet omkring en semi-submersible med CFD
JONATHAN ERIKSSON
Avdelningen för Marin Teknik
Institutionen för Sjöfart och Marin Teknik
Chalmers Tekniska Högskola

Sammanfattning

Tillämpning av Computational Fluid Dynamics (CFD) är nästan en självklarhet inom de teknikområden som hanterar olika flöden av vätskor. Dess användning inom offshoresektorn är dock fortfarande begränsad, främst på grund av de långa beräkningstiderna som krävs för att simulera kompletta sjöförhållanden. Men med ökande datorkapacitet och uppkomsten av nya kreativa lösningsmetoder, så ökar användbarheten kontinuerligt. Idag är branschen mycket beroende av fysiska modellförsök; försök som är dyra, tidskrävande och kan ställas inför avvikelser på grund av skaleffekter och opålitlig mätutrustning. Genom att istället använda CFD kan man analysera de fysikaliska egenskaperna var som helst inom beräkningsdomänen, samt anpassa och justera de geometriska och numeriska inställningarna när som helst. Huvudsyftet med detta arbete är att undersöka användningen av CFD-simuleringar för utformningen av offshoreplattformar.

I denna studie är den nedsänkta delen av en offshoreplattform föremål för undersökning. Data från ett tidigare genomfört vindtunnelexperiment används som medel för både verifiering och kalibrering av CFD-modellen. Två olika CFD-programvaror används i detta arbete; OpenFOAM, vilket är baserat på öppen källkod, samt den kommersiella programvaran Fluent. Initialt modelleras en kopia av vindtunneln och inkompressibla steady-state Reynolds Averaged Navier-Stokes (RANS) simuleringar utförs, eftersom det är av intresse att finna instanser där steady-state simuleringar är möjliga. Undersökningarna omfattar steady-state simuleringar för både komplett geometri och trunkerat närområde.

Icke tillfredsställande resultat erhölls. Dragkraften som uppmättes i steady-state simuleringarna uppnådde endast ett värde på 60 % av den experimentellt uppmätta dragkraften. Tidsberoende simuleringar, både RANS och LES, visade på liknande resultat. Orsaker till det otillfredsställande resultatet kan vara otillräcklig upplösning av meshen, tillsammans med felaktiga val av diskretiseringsmetoder och turbulensmodell.

Nyckelord: CFD, OpenFOAM, Fluent, semi-submersible, vindtunnel, strömkrafter.

Nomenclature

Abbreviations

BC	Boundary Condition
CFD	Computational Fluid Dynamics
CV	Control Volume
DNS	Direct Numerical Simulation
DNV	Det Norske Veritas
FVM	Finite Volume Method
GUI	Graphical User Interface
LES	Large Eddy Simulation
NWB	Numerical Wave Basin
OpenFOAM	Open Source Field Operation And Manipulation
PDE	Partial Differential Equation
PIMPLE	Combination of PISO and SIMPLE
PISO	Pressure Implicit with Splitting of Operators
RANS	Reynolds Averaged Navier-Stokes
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
URF	Under-Relaxation Factor
WADAM	Wave Analysis by Diffraction and Morison Theory

Dimensionless quantities

Re	Reynolds number
y^+	Dimensionless wall distance

Greek letters

ν	Kinematic viscosity
μ	Dynamic viscosity

Mathematical Operators

div	Divergence
$grad$	Gradient
∂	Partial derivative

Roman letters

L	Characteristic length
p	Pressure
S_M	Source term
t	Time
$\mathbf{u}$	Velocity vector
u, v, w	Velocity components in Cartesian coordinate system
x, y, z	Cartesian coordinates

Subscripts

∞	Free stream property
w	Wall

Superscripts

$'$	Fluctuation
$\bar{}$	Time-average

Acknowledgements

First of all, I would like to express my gratitude towards my supervisors at GVA, Vahik Khodagolian and Johan Lennblad, who gave me a lot of valuable counseling and made me feel very welcome during my time there.

I especially want to thank my supervisor at the Division of Marine Technology at Chalmers, Claes Eskilsson. Without your patience and guidance this thesis would not have been finished.

Finally I wish to thank my family for their loving support and encouragement. Above all I wish to thank my girlfriend Kajsa for putting up with me during my struggles with this work, and for always believing in me.

Jonathan Eriksson

Contents

Abstract **I**
Sammanfattning . . . III
Nomenclature . . . V
Acknowledgements . . . VII

1 Introduction **1**
1.1 Conventional design approaches . . . 2
1.2 Why use CFD in offshore? . . . 3
1.3 CFD studies for semi-submersibles . . . 4
1.4 Aim and Scope . . . 6

2 Computational Fluid Dynamics **9**
2.1 Navier-Stokes equations . . . 9
2.2 Discretization of modeled equations . . . 11
2.2.0.1 Temporal discretization . . . 11
2.2.0.2 Convective discretization . . . 12
2.2.0.3 Gradient discretization . . . 13
2.3 Turbulence and Its Modeling . . . 13
2.3.1 Turbulent Flow . . . 13
2.3.2 Direct Numerical Simulation (DNS) . . . 14
2.3.3 Large Eddy Simulation (LES) . . . 14
2.3.4 Reynolds Averaged Navier-Stokes (RANS) . . . 14
2.3.5 Turbulence models . . . 16
2.3.5.1 $k-\omega$ SST . . . 16
2.3.5.2 Realizable $k-\epsilon$ . . . 16
2.3.6 Boundary layers . . . 16
2.3.7 Wall functions . . . 17
2.3.8 Drag force coefficient . . . 18
2.4 Software . . . 19
2.4.1 OpenFOAM . . . 19
2.4.1.1 Solvers . . . 19
2.4.2 ANSYS Workbench . . . 20
2.4.3 Other software . . . 21

3 Experimental Setup **23**
3.1 The wind tunnel . . . 23
3.2 Presentation of the cases to investigate . . . 25

4 Simulation of the entire tunnel **27**
4.1 Geometry . . . 27
4.2 Approach . . . 28
4.3 Simulation settings . . . 29
4.4 Results . . . 29

4.4.1 Flow results . . . 29
4.4.2 Drag and force results . . . 31

5 Near-field simulation **35**
5.1 Geometry . . . 35
5.2 Approach . . . 36
5.3 Simulation settings . . . 36
5.4 Results . . . 36

6 Discretization of the convective term **39**
6.1 Geometry . . . 39
6.2 Approach . . . 40
6.3 Simulation settings . . . 40
6.4 Results . . . 41

7 Unsteady simulations **43**
7.1 Geometry . . . 43
7.2 Approach . . . 43
7.3 Simulation settings . . . 43
7.4 Results . . . 43

8 Conclusions and recommendations for future work **47**
8.1 Conclusions . . . 47
8.2 Recommendations for future work . . . 48

Bibliography **49**

Chapter 1

Introduction

Semi-submersible offshore structures are designed to operate in harsh environments which demands for properly dimensioned dynamic positioning and mooring systems. A typical semi-submersible consists of submerged bodies connected by slender walls or columns to the operating decks above the water. Production, drilling and accomodation platforms are all built in this fashion since the motions of these structures are generally smaller than those of a more ship like structure, such as a barge [1]. This makes them especially suitable for tasks which involves very strict requirements on motion. Another benefit of the semi-submersibles are the large operating decks, available as working space and for storage of equipment or supplies. Figure 1.1 shows a rendered picture of the GVA 8000 semi-submersible drilling unit [2].



Figure 1.1: GVA 8000 semi-submersible drilling unit, from [2].

When designing a semi-submersible offshore structure, it is of importance to attain as good dynamic behaviour as possible, in order to prevent drift of the structure due to environmental loads, which consist of wave, wind and current contributions. This can be achieved either by using passive systems, such as moorings, or by active systems, such as dynamic positioning. In any case, an accurate prediction of the environmental loads is vital in both the design and the engineering stage, as well as for the operation of station keeping systems. Recommended practices for how to model, analyze and predict the environmental effects and its loads on a structure are supplied by classification societies, such as those of Det Norske Veritas (DNV) [3, 4].

1.1 Conventional design approaches

Semi-submersible platforms, in contrast to for instance tanker ships, have a complex non-streamlined geometry and show large variations in both shape and dimension among the different platforms. For that reason there is no design method available in which the dimensions of a semi-submersible follow from an optimization technique. This led the designs of the first platforms to be more or less based on assumptions decided at random, complemented with model tests of the original design and an eventual additional model test with an altered design if the results from the first tests were inadequate [1, 5, 6].

Scaled model tests of different sorts have therefore been used since the start to estimate the environmental loads, and have thus played an integral role in the design and development of semi-submersibles to this day. The experimental test method which is acknowledged as the most dependable tool to emulate the realistic and extreme conditions which an offshore platform is likely to experience, is the wave basin test. In a wave basin it is possible to expose the model to all of the environmental loads in order to see how the structure will behave [7]. Figure 1.2 shows a model tested in such a basin [8].

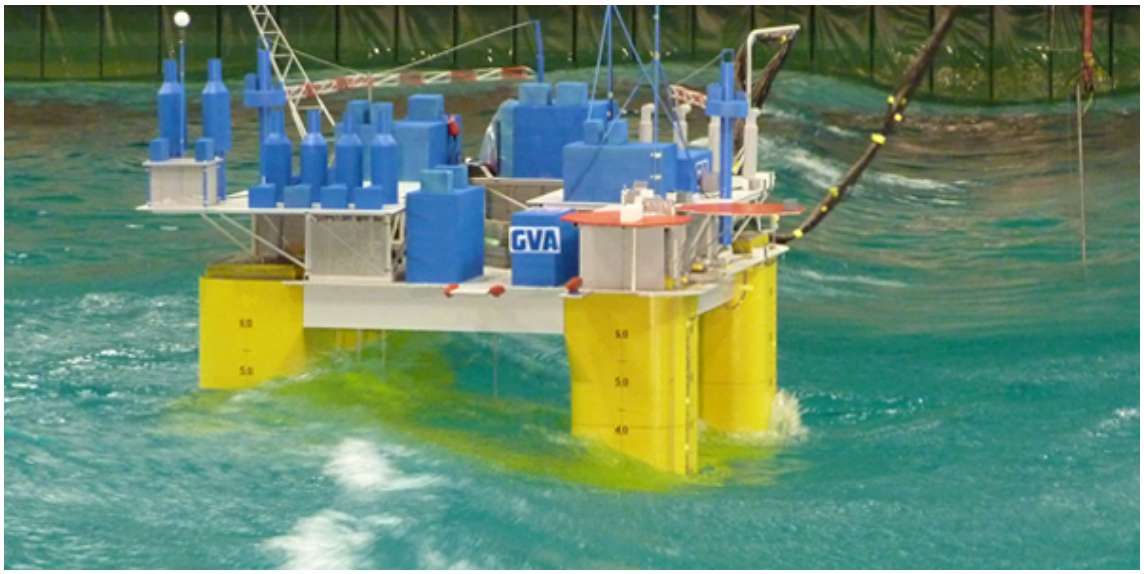


Figure 1.2: An experimental test conducted in a wave basin, from [8].

Another method of model testing relevant for offshore structures is experiments performed in a wind tunnel. By this method it is possible isolate and estimate the contribution from the wind and current induced forces which the structure will be subjected to. A wind tunnel can be especially useful when investigating the environment on the topside of a structure. Since the working decks of a platform are normally compressed with buildings, trusses, cranes and a helideck, it is of interest to study the interaction of these on for instance flammable gas dispersion.

Designing an offshore structure is an iterative process with the aim of optimizing the hull in terms of performance, manufacturability and cost. The sizing of a hull thus requires continuous input from miscellaneous criteria, such as strength, stabilty and global performance. Global performance refers to the structure's motion in water, and it is mainly obtained via a numerical radiation/diffraction analysis tool. These tools, based on the linear and second-order potential theory, are used to assess the loads and motions due to waves. WAMIT [9] is a prime example of such a software, widely used in the industry today. According to DNV [4], semi-submersibles with slender structures, such as braces, may also require a Morison load model, apart from the radiation/diffraction tool. Wave Analysis by Diffraction and Morison Theory [10], or WADAM for short, is a software that employs the potential theory directly from WAMIT, and combines it with Morison theory.

WAMIT and WADAM certainly are useful tools. However, there are physical phenomena, usually related to nonlinear fluid forces and viscous effects, which they have difficulties dealing with. This becomes evident when an unexpected physical phenomenon is obeserved in a model test, one that has not been anticipated analytically. Too large discrepancies between the analytic prediction and the empirical model test may ultimately lead to modifications of the hull design, thus delaying the project and increasing the costs [11].

1.2 Why use CFD in offshore?

As a complement to the traditional design methods of predicting the environmental loads, numerical simulations performed by CFD is available. CFD is a commonly used tool within many fields of engineering dealing with fluid flows, however, it has still not been fully adopted by the offshore industry. This is partly due to the large amount of computational time necessary to simulate a complete sea state, see e.g. Eskilsson et al [12]. There is however a constant development as computational power continuously increases along with the onset of new solution methods.

Numerical CFD models can present more local information compared to the traditional ones, as the forces and properties of the flow can be obtained from any point within the computational domain. Furthermore, they are much easier to modify, whether it may be a geometrical or physical alteration. Another advantage of using CFD in comparison to traditional methods is the ability to carry out simulations without taking any effects of scale into consideration, since the physical consequences are intrinsic in the CFD technique.

Considering that the CFD tool is not influenced by the effects of scale, it offers the opportunity to examine the actual consequence of the scaling effects. There are unfortunately little to no data available to support a full-scale CFD simulation, in contrast to a model-scale simulation. However, by calibrating and verifying CFD models to existing tests, the accumulated knowledge could subsequently be applied on full-scale structures. Establishing a CFD methodology as such offers several benefits over traditional testing. This includes cost, speed and an increased ability to

modify existing models.

The industry is certainly undergoing a paradigm shift towards implementing CFD as a design tool. In a recent study by Kim et al [13], the economic and technical readiness of a CFD-based Numerical Wave Basin (NWB) for semi-submersible design was reviewed. The study showcased the technical readiness of the NWB by simulating fully coupled hull-mooring-riser systems. The name of the NWB-tool is MrNWB (Mooring/Riser Numerical Wave Basin), and it consists of three parts; a mooring/riser model, a wave model and a CFD solver. The method allows for three different computational zones to be formed. The outmost located zone is the Euler Wave Zone, where far-field waves are modelled, and the innermost zone is the CFD zone, which solves for fully non-linear waves within close proximity of the semi-submersible. The Overlay Zone, situated between the two aforementioned, gradually combines the CFD solutions with the Euler solutions. In addition to the wave contribution, the hydrodynamic forces and moments due to mooring are calculated using Morison theory. This information is in turn feeded to the CFD solver in order to update the structure's motion. Results indicate that the cost of the CFD project is about equal to the cost of the reference model test. The most anticipated advantages of using the CFD-based NWB, according to the study, are:

1. Time schedule and cost estimation are reliable and easy to foretell
2. Fast turn-around time for adjustments in design and environmental data
3. Reduced cost and time for the second run of NWB simulation, if additional computation is necessary

1.3 CFD studies for semi-submersibles

Even though the conditions for employing CFD in semi-submersible design already have been met, further studies are still essential in order to supply more realistic field simulations. Until the technique can be used on its own with appropriate precision, a greater insight of the possibilities, limitations and sensitivity of CFD results to different criteria is necessary. This literature study aims to find relevant research done where especially the wind and current loads have been predicted by CFD.

Zhang et al. [14] studied the wind flow and loads on a model-sized topside structure with rather simplified geometrical features. They were using commonly used CFD techniques, such as Reynolds Averaged Navier-Stokes Equations (RANS) and Large Eddy Simulation (LES), with different turbulence models. Focus lied on finding efficient and dependable approaches for the assessment of the wind loads. The results of the simulations, which were performed in FLUENT, were subsequently compared with data measured in a wind tunnel. Results show that LES simulations with a dynamic SGS (sub-grid scale) model are the most comparable with the wind tunnel measurements. It was also found that proper modeling of the initial boundary conditions is crucial, since incorrect velocity profiles and turbulence intensity

profiles resulted in discrepancies of the results. Even though the numerical results are comparable to the experimentally measured ones, the authors state that wind tunnel tests still remain indispensable for investigating wind phenomena on offshore structures, and they address the need for further improvements of CFD applications within the field.

A joint industry project, lead by Vaz et al. [15], conducted a comparative study of different methods to predict the current loads on a semi-submersible. Geometry-wise the structure was simple, just consisting of a single pontoon attached to two square columns with rounded corners. The methods which were compared were: towing tank model testing, semi-empirical predictions and CFD calculations. By quantifying the deviations of the different methods' results to the experimentally measured data, comparisons and conclusions could be drawn. A goal of the study was to establish best practice guidelines for CFD work of this sort. Unsteady RANS was chosen for the CFD calculations and among the participants three different codes were used: CFX, STAR-CCM+ and FreSCo. Each participant was asked to carry out the calculations in steps of three:

1. Blind Computations: computations which were run without prior knowledge of the experimental data.

2. Improved Computations: altered computations with the purpose of achieving better correlation with the experimental data.

3. Additional Computations: supplementary numerical studies (such as turbulence, wall functions and roughness) and physical studies (such as free-surface effects, effect of draft and distance between columns).

Results show that it was difficult to obtain an accurate time history of the drag and lift loads. The Blind Computations yielded results which were more than 20% erroneous for the average drag while the lift was nonexistent in most of the computations. After the Improved Computations were concluded the errors of the best result had been reduced to 8% and 25% for the average drag and lift respectively. The computations overall showed large variations in the loads over time. For that reason the experiments should be performed again, ensuing a more trustworthy statistical analysis. The Additional Computations revealed that incorrect laminar-turbulent transition and numerical errors are of special concern for the results. Additional studies of these issues are said to be necessary before physical aspects, such as scale effects, can be considered.

A study which has investigated the possible scale effects on the current coefficients of a semi-submersible has been conducted by Koop and Bereznitski [16]. However, the main purpose of their paper was to research the applicability, cost and accuracy of CFD computations to attain the current loads for all headings of the structure. This was achieved by first performing a comprehensive verification study using 10 different grids to assess the model-scaled current loads. These results were compared to experimental data measured in both a wind tunnel and an offshore basin.

Thereafter the effect of including a thruster geometry to the structure was looked into. Finally, full scale computations were launched with 5 successively refined grids in order to determine the impact of grid resolution. All of the simulations were steady-state and performed by the MARIN in-house CFD code ReFRESCO. The results from the model scaled simulations indicated that the force coefficients are quite independent on grid setting. When comparing the results to the experimental data it seems that the best agreement was obtained between the CFD and wind tunnel. Furthermore, it appears that the differences between the CFD simulations and the physical experiments are greater than the influence of the implemented thrusters. Both the model scale and full scale results converged with increasing grid refinement. The full scale values are on average roughly 15-20% below the model scale's results, with larger variations for some headings.

Croonenborghs et al. [17] predicted the wind and current loads on the Statoil operated Åsgard B platform through full-scale steady-state RANS simulations. The paper was focused on two objectives: to model the Atmospheric Boundary Layer (ABL) with good accuracy and to attain a high quality mesh for a complex deck structure. In most studies regarding environmental loads, such as those of Zhang et al. and Vaz et al., a simplified geometry of the structure have been used. Simplifying the geometry normally offer several benefits, like: reduced mesh size, improved mesh quality, and better computation times. The downside is that the results become less accurate with increasing level of simplification. To circumvent this, a highly detailed geometry was used; only excluding small, insignificant details from the original CAD model. The results of the full-scale simulations, which were carried out using OpenFOAM, were ultimately compared to wind tunnel measurements of a scaled model. A mesh of good quality was achieved, although no viscous layers were modeled, and the numerical results match all in all quite well with the experimental data. The authors state that further research concerning the influence of unsteady flow and scale effects is required.

In a subsequent study by Croonenborghs et al. [18] on the same platform, four participants each assessed the wind and current loads by different means, with the purpose of examining the variation of the results among the different methods. Two of the participants ran wind tunnel measurements in two different tunnels using the same scale model, another participant performed the CFD computations which were accounted for above, while the fourth participant launched towing tank tests. Some of the results agreed well among the three different approaches which were compared, whereas other results demonstrated notable variations between the different approaches. The variability in their results highlights the significance of using multiple sources for comparison, upon evaluating the environmental loads on a semi-submersible platform.

1.4 Aim and Scope

The purpose of this project is two-fold. The first objective is to validate the use of CFD to replicate and validate wind tunnel measured loads on the hull. Since

these kind of structures do not offer any comparable test results, knowledge must be attained by performing parameter variations for the CFD models. The outcome from this is a detailed report concerning rule-of-thumbs for numerical settings to be used by GVA in future CFD work.

The second purpose is specifically oritented towards investigating a boundary layer reducing fence used in the experimental test, and its influence on the measured forces, especially the side forces.

Existing data from a wind tunnel test will be used to both verify and calibrate the CFD model. There are instances in the process where possibly steady-state simulations can be used, focus will therefore lie on investigating eventual steady-state applicability.

This thesis is limited to study the current forces on the hull of a semi-submersible, since the top side of the structure is much more geometrically complex. Furthermore, the work is restricted to investigate just one condition at even keel and head on wind, due to the limited time-frame of this thesis.

Chapter 2

Computational Fluid Dynamics

Computational Fluid Dynamics or CFD is a branch of fluid mechanics that uses numerical methods to solve and analyze problems that involves fluid flow with the help of computer-based simulations. The core of CFD are numerical algorithms, which are used to solve the governing equations of fluid dynamics: the conservation of mass, momentum, and energy. It is conventional to split the CFD-process into three main elements: pre-processing, solving and post-processing. Pre-processing involves the preparation of the required input for the numerical solution process. The most necessary activities at this stage, according to Versteeg and Malalasekera [19], involve:

- Mathematical expression of physical phenomena
- Definition of geometry
- Generation of the mesh
- Definition of fluid properties
- Establishing initial and boundary conditions

2.1 Navier-Stokes equations

The mathematical expressions of the physical phenomena must naturally be formulated prior to the numerical solution in the subsequent step. For a flow that can be assumed to be incompressible and isothermal, the mathematical expressions which are derived from the governing equations mentioned earlier, become [19].

$$div(\mathbf{u}) = 0 \tag{2.1}$$

$$\frac{\partial u}{\partial t} + div(u\mathbf{u}) = div(\mu\ grad\ u) - \frac{\partial p}{\partial x} + S_{Mx} \tag{2.2}$$

$$\frac{\partial v}{\partial t} + div(v\mathbf{u}) = div(\mu\ grad\ v) - \frac{\partial p}{\partial y} + S_{My} \tag{2.3}$$

$$\frac{\partial w}{\partial t} + div(w\mathbf{u}) = div(\mu\ grad\ w) - \frac{\partial p}{\partial z} + S_{Mz} \tag{2.4}$$

The first equation Eq. (2.1) is known as the continuity equation and describes the divergence of the velocity in each point. Equations Eq. (2.2) to Eq. (2.4) are the the Navier-Stokes equations which represent the conservation of momentum in the x, y and z direction, respectively. Together, these equations describe the movement

of three-dimensional, unsteady fluid flow of an incompressible, Newtonian fluid. For a more in-depth explanation of these equations, see [19].
In the following solving step of the CFD procedure, the partial differential equations (PDEs) are discretized. A prerequisite for this is a well defined geometry of the computational domain, including every geometric feature that influence the flow of the fluid. For cell based CFD methods, the computational domain is subsequently discretized in space by the generation of the mesh. This populates the domain with computational cells, or control volumes (CVs), which essentially serve as finite fluid elements. The cell center of each of these elements contain the pressure and velocity calculated during the upcoming numerical solution process. A definition of the fluid properties as well as specifying the initial and Boundary Conditions (BCs) are also necessary to solve the system at hand. For an incompressible fluid it is enough to specify the viscosity and the density of the fluid. The BCs determine the values of the transported elements at all boundaries of the computational domain. In the computational domain the boundaries consist of an inlet, outlet, surrounding walls and the defined geometry discussed earlier. The BC is constant in contrast to the initial condition, which represent a first guess of the corresponding property values in order to start the iterative algorithm. By setting the initial values as close to the correct values as possible, the user may improve the solution process.

The second element of the CFD-process is the stage of solving the systems of the PDEs stated. There are several techniques available that may be used to solve these, such as Finite Difference, Finite Element Method and Spectral Method. However, the most well-established (and comprehensively validated) CFD-technique is the Finite Volume Method (FVM). The FVM-routine consists of the following steps [19]:

- Discretization of PDEs
- Solution of the resulting linear system of algebraic equations

The intent of the discretization is to reduce one or more PDE to a set of discrete linear equations that in turn can be solved to obtain the value of the dependent variable at each CV centre. This results in a sparse linear system that requires an iterative scheme to solve. The discretization can be divided in two parts according to Jasak [20]: the discretization of the computational domain, which has been performed during the mesh generation, and the discretization of the PDEs. For an unsteady simulation, the equations have to be discretized in time as well as in space. The spatial discretization of the equations is done by integrating the underlying equations for each CV. Discretization of the time starts by dividing the time interval into a finite number of time-steps. The equations are then discretized in time by the integration of the governing equations in time. There are several discretization schemes accessible for the discretization of the governing equations. What separates these schemes are mainly differences in accuracy and boundedness of the solution. When it comes to accuracy, it is preferable for the order of the discretization to meet the order of accuracy of the initial PDEs. The Navier-Stokes equations are of second order since the first term of the right hand sides of Eq. (2.2)

to Eq. (2.4) contain a second derivative of the velocity in space. Thus, it would be advantageous from an accuracy point-of-view to achieve the same order from the discretization. The bottom line of the discretization procedure is a sparse linear system of algebraic equations, which can be expressed as Eq. (2.5) below [19][20].

$$\mathbf{A\Phi} = \mathbf{R} \tag{2.5}$$

where $\mathbf{A}$ is the discretization matrix, $\mathbf{\Phi}$ is the vector containing the variables being solved for and $\mathbf{R}$ contains the source terms of the equations. This system of equations would ideally be solved right away with an appropriate solver in the final step. However, the underlying equations Eq. (2.2) to Eq. (2.4) are non-linear and the velocities and pressure are coupled to one another. In order to sidestep this problem various iterative algorithms have been constructed. These algorithms typically consists of momentum predictors and pressure correctors. A momentum predictor calculates a new velocity field with the momentum equations, based on the pressure of the previous time-step or the initial guess, while a pressure corrector with the help of the continuity equation corrects the pressure field. The most common algorithms for solving fluid flow are the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) [**?**] and the Pressure Implicit with Splitting of Operators (PISO) [**?**] and their derivations. PISO is being used for transient problems and SIMPLE for steady-state. Both algorithms are based on evaluating some initial solutions and then correcting them. SIMPLE only makes 1 correction whereas PISO requires atleast 2. Finally, the resulting systems of algebraic equations have to be solved using an appropriate solver. This can either be done with direct or iterative methods. The direct method provides the solution of the system in a finite number of arithmetic operations while the iterative method improves an initial guess of the solution, until convergence is attained [19][20].

Following the solving of the equations comes the final step of the CFD-process, namely post-processing. This step offers the opportunity to visualize and analyze the numerical results in various ways.

2.2 Discretization of modeled equations

In order to solve the Navier-Stokes equations one must discretize the PDEs. There exist different discretization schemes for different terms of the modeled equations, and they will briefly be presented in this section.

2.2.0.1 Temporal discretization

Discretization of the term $\partial/\partial t$ determines how the algorithm updates the solution in time. The temporal discretization is done through integration over time on the general discretized equation. Let the spatial domain be discretized to a semi-discrete form, Eq. (2.6):

$$\frac{\partial \phi}{\partial t}(x,t) = f(\phi) \tag{2.6}$$

The discretization of Eq. (2.6) can be performed in many ways, the following discretization, Eq. (2.7) is valid for the explicit Euler, implicit Euler and Crank-Nicholson schemes,

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = f(\theta\phi^{n+1} + (1-\theta)\phi^n) \tag{2.7}$$

Here, ϕ indicates an arbitrary property and the time step is defined as $\Delta t = t^{n+1} - t^n$. When $\theta = 0$, the discretization is known as explicit Euler, which is first order accurate. If $\theta = 1$ the discretization is called implicit Euler, also first order accurate. Lastly, for $\theta = 0.5$, a second order accurate scheme, the Crank-Nicholson, is attained.

When running a steady-state simulation, where one assumes that there are no time dependencies affecting the solution, the time derivatives are not accounted for.

2.2.0.2 Convective discretization

In each and every cell center, the value of a transported property, ϕ, is being stored. The discretization of the convective terms controls which value the fluid should be ascribed when it crosses the face of a cell. For the solutions to be physically realistic, a discretization scheme should satisfy the following requirements [19]:

- **Conservativeness.** Flux consistency at the CV faces. In other words the flux of ϕ leaving a CV must be the same as the flux of ϕ entering the adjacent CV through the same face.
- **Boundedness.** The predicted values are limited within certain physically realistic bounds.
- **Transportiveness.** Accounts for the direction in which the relative strengths of convection and diffusion influence the flow.

Centered schemes

The central difference scheme is a commonly used interpolation scheme which could be applied to discretize the convective terms. However, it is not ideal for CFD purposes, since the scheme is not able to recognise the direction of the flow, nor the strength of convection relative to diffusion, thus not satisfying the requirements of boundedness and transportiveness. [19].

Upwind schemes

One way of sidestepping this problem, is by implementing an upwind scheme. The upwind scheme, developed for strong convective flows with suppressed diffusion effects, takes direction of the flow into account, overcoming that inability of the central differencing scheme. This scheme is first order accurate, conservative and bounded. One major shortcoming of the scheme, is the so-called false diffusion which occur when the flow is not aligned with the grid. When this problem arises, the transported

properties tend to become smeared, and the resulting error has a diffusion-like appearance [19].

The spatial accuracy of the upwind scheme can be improved further by including another data point. In other words, interpolation between the nodes is being done in order to increase the accuracy. This method is referred to as the linear upwind scheme and it is second order accurate.

2.2.0.3 Gradient discretization

The gradient of a scalar, ϕ, at a given CV centroid is usually computed with the help of Green-Gauss theorem, which states that the surface integral of a scalar function is equal to the volume integral of the gradient of the scalar function. Thus, the values of ϕ at the surfaces of the CVs needs to be calculated in order to solve the gradient. There are two main strategies to calculate the face values of a scalar:

- **Cell based method.** The face value is computed using the values at its neighboring cells. This method is easy to implement and is the least computationally demanding.
- **Node based method.** The face value is computed using the values at its neighboring nodes. It is more accurate compared to the cell based one, at the expense of being more computationally intensive.

The cell gradient can alternatively be calculated using a least squares method, where the gradients between the cell and its neighbors are calculated.

2.3 Turbulence and Its Modeling

2.3.1 Turbulent Flow

The flow around an object can either be laminar or turbulent. Laminar flow is characterized by a smooth, constant fluid motion, while the turbulent flow on the other hand is characterized by chaotic property changes, such as rapid variation of pressure and flow velocity in space and time. Additional features that defines turbulent flow are three-dimensionality and spawning of unsteady vortices of different sizes, known as eddies. As a result of interaction between eddies, kinetic energy is continuously transferred from larger eddies to smaller ones, in a progress referred to as the energy cascade, and lasts until the energy is finally dissipated in the smallest eddies, the so called Kolmogorov scales. The smaller eddies are influenced by viscous effects, while the larger eddies are predominantly driven by inertia effects. However, both eddies are important for calculations, simply because all properties of the flow are relevant. This is what makes simulations complex and very demanding on processing power, since even the smallest eddies need to be fully resolved and the cell-size of a computational mesh has to be of the same order of scale.

One means of differentiating between laminar and turbulent flow is given by the dimensionless Reynolds number in Eq. (2.8). The quantity is defined as the ratio of inertial forces to viscous forces and consequently quantifies the relative importance of these two types of forces for a certain flow condition. Reynolds [21], described the number as below.

$$Re = \frac{U_\infty L}{\nu} \tag{2.8}$$

where U_∞ is the free stream velocity, L is a characteristic length and ν is the kinematic viscosity. Turbulent flow is generally characterized by high Reynolds numbers.

2.3.2 Direct Numerical Simulation (DNS)

A DNS solves the Navier–Stokes equations numerically without using any turbulence model. This means that the whole spectrum of spatial and temporal scales of the turbulence must be fully resolved, from the smallest Kolmogorov scales, up to the integral scale length. This makes DNS very extensive and thus is the computational cost of DNS very high, even at low Reynolds numbers. Therefore lies its main application in research and academia, while either RANS or LES solutions are preferred for normal engineering applications.

2.3.3 Large Eddy Simulation (LES)

The major difficulty in simulating turbulent flows, comes from the wide range of length and time scales to account for. Instead of resolving the whole range, as DNS does, LES resolves just the large scales of the flow field solution. The motivation behind this, is that the large scales are dependent on geometry, while the smaller, more expensive ones [22], are considered universal. These small scales are filtered out and implicitly modeled using a subgrid-scale turbulence model. Thus, all flow scales larger than the specified filter size will be fully resolved, and the scales smaller than the filter size will be modeled. The success of an LES is dependent on the quality of the subgrid-scale turbulence model and to a large extent on the resolution of the grid. Even though LES requires just a fraction of the computational power of DNS, typically around 1% [23], it is still computationally expensive. The necessary high grid resolution to resolve the scales, in combination with the small time-steps usually required to simulate unsteady flow, lead to long run-times and large volumes of data.

2.3.4 Reynolds Averaged Navier-Stokes (RANS)

In the interest of analyzing the influence of turbulence on the mean flow, which is adequate for many industrial applications, the flow may be separated into a mean and fluctuating part. This process is referred to as Reynolds decomposition and is the first step of the RANS method. The Reynolds decomposition reads

$$\phi = \bar{\phi} + \phi' \tag{2.9}$$

where ϕ is an arbitrary property, decomposed into its mean value $\bar{\phi}$ and its fluctuating part ϕ'. When applying the Reynolds decomposition to Eq. (2.2) to Eq. (2.4) the so-called Reynolds-Averaged Navier-Stokes equations are formed, which read

$$div\ \bar{\mathbf{u}} = 0 \tag{2.10}$$

$$\frac{\partial \rho\bar{u}}{\partial t} + div(\rho\bar{u}\bar{\mathbf{u}}) = \ div(\mu\ grad\,\bar{u}) - \frac{\partial \bar{p}}{\partial x} + \left[\frac{\partial(-\rho\overline{u'^2})}{\partial x} + \frac{\partial(-\rho\overline{u'v'})}{\partial y} + \frac{\partial(-\rho\overline{u'w'})}{\partial z}\right] \tag{2.11}$$

$$\frac{\partial \rho\bar{v}}{\partial t} + div(\rho\bar{v}\bar{\mathbf{u}}) = div(\mu\ grad\,\bar{v}) - \frac{\partial \bar{p}}{\partial x} + \left[\frac{\partial(-\rho\overline{u'v'})}{\partial x} + \frac{\partial(-\rho\overline{v'^2})}{\partial y} + \frac{\partial(-\rho\overline{v'w'})}{\partial z}\right] \tag{2.12}$$

$$\frac{\partial \rho\bar{w}}{\partial t} + div(\rho\bar{w}\bar{\mathbf{u}}) = div(\mu\ grad\ \bar{w}) - \frac{\partial \bar{p}}{\partial x} + \left[\frac{\partial(-\rho\overline{u'w'})}{\partial x} + \frac{\partial(-\rho\overline{v'w'})}{\partial y} + \frac{\partial(-\rho\overline{w'^2})}{\partial z}\right] \tag{2.13}$$

The Reynolds decomposition has introduced new terms on the right hand side of Eq. (2.11) to Eq. (2.13). These terms are unknown and are referred to as the Reynolds stresses. At this stage, there are ten unknowns (three velocity components, pressure, six stresses), but only four equations. This is the so-called closure problem. Thus, in order to close and solve the system of equations, the stresses must be determined. According to Davidson [24], there exist different levels of models able to solve this system of equations. They are listed below in increasing order of complexity, capability to model the turbulence and in demand of processing power.

1. **Algebraic models.** The most simple models use an assumption introduced by Boussinesq [25], in order to calculate the Reynolds stresses. Simply put, a turbulent viscosity is first calculated using an algebraic equation. The turbulent viscosity and the velocity gradients are then related to the Reynolds stresses using Boussinesq assumption, allowing the stresses to be computed. All of the following models are based on Boussinesq assumption.

2. **One-equation models.** One equation turbulence models solve a transport equation of a turbulent property, typically the turbulent kinetic energy k, and a second turbulent quantity, usually a turbulent length scale, is attained via an algebraic expression.

3. **Two-equation models.** These models include, as the name implies, two transport equations that represent the turbulent properties of the flow. This enables a two-equation model to account for history effects, like convection and diffusion of turbulent energy. These models are the most commonly used turbulence models, and models like the $k-\omega$ of Kolmogorov [26], and the $k-\epsilon$ model by Jones and Launder [27] have become industry standard models for many engineering applications.

4. **Reynolds stress models.** These are the most elaborate turbulence models and they usually employ a method called Second Order Closure to solve the system of equations. For more information regarding these turbulence models, see [24].

2.3.5 Turbulence models

When modeling the governing equations with RANS, it is, as stated earlier, a necessity to model the turbulent scales. The models briefly presented here are the ones who are relevant for this project.

2.3.5.1 $k-\omega$ SST

The shear stress transport (SST) formulation is a mix of the $k-\omega$ and $k-\epsilon$ models. In the region near the wall, the $k-\omega$ formulation is used while the $k-\epsilon$ method is applied farther away from the wall, in the fully turbulent regions. The model is widely used and shows good behaviour in adverse pressure gradients and separating flow.

2.3.5.2 Realizable $k-\epsilon$

An improvement over the standard $k-\epsilon$ model, the realizable $k-\epsilon$ model differs from the standard model in two ways: it contains a new formulation for the turbulent viscosity and a new transport equation for the dissipation rate, ϵ. The realizable $k-\epsilon$ model is proficient in capturing the mean flow of complex structures, according to Davis et al [28].

2.3.6 Boundary layers

A boundary layer is a layer of fluid adjacent to the surface of an object past which the fluid flows. If one assumes a no-slip condition when the fluid is in contact with the object's surface, there can be no relative motion between the fluid in contact with the surface and the surface itself. Thus, if the surface has zero velocity, then the fluid in contact with the surface has zero velocity as well. These fluid particles that are stuck to the wall will also slow down neighboring particles due to viscosity and fluid friction, creating a thin layer of fluid between the surface, where the velocity is zero, and the free streaming fluid a little farther away from the surface. This layer in turn may be sub-divided into three different parts, according to Tennekes and Lumley [29]: laminar-, transition- and the turbulent layer, as seen in Figure 2.1.

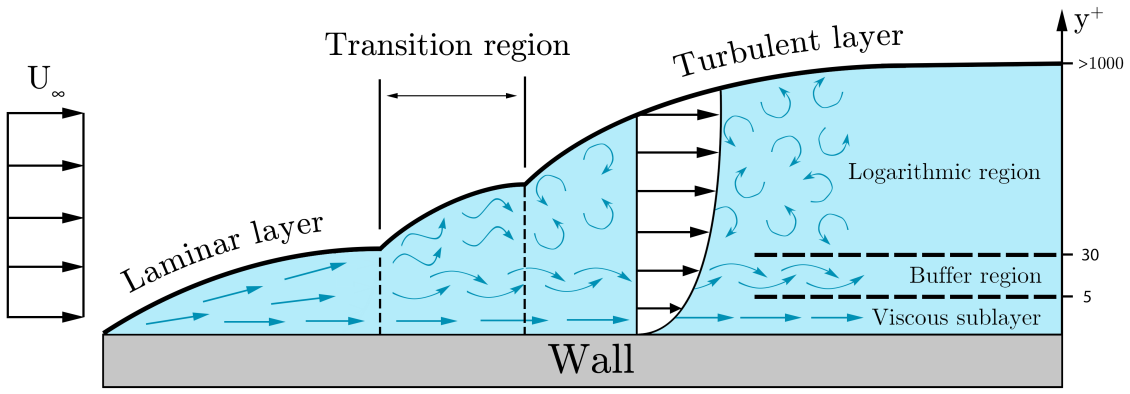


Figure 2.1: The three boundary layer regions. After [30].

The laminar boundary layer forms when the uniform velocity fluid hits the leading edge of the surface. In this region the flow is very smooth and predictable. After some distance downstream, small chaotic oscillations start to develop in the fluid, and the flow begins to transition to turbulence, eventually becoming fully turbulent. The turbulent flow near the wall can be divided up into three zones. For a thin layer just above the wall, the flow velocity is linear with distance from the wall. This layer is referred to as the viscous sublayer, due to that the shear stresses are dominated by viscous shear stresses. Farther outwards from the viscous sublayer, in the buffer region, the viscous shear stresses are gradually being replaced by turbulent shear stresses, until being fully replaced in the logarithmic region. Note that the logarithmic region is considerably larger than the other two regions combined, which is told by the y^+ axis in Figure 2.1, which is not in scale. The quantity y^+ is a dimensionless wall distance, defined as follows.

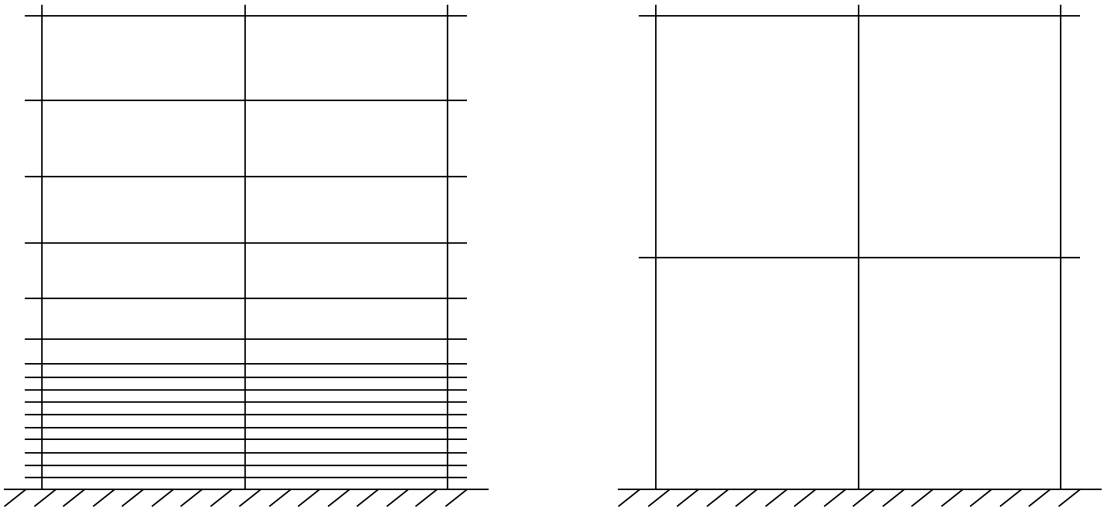
$$y^+ = \frac{u_\tau y}{\nu} \tag{2.14}$$

where $u_\tau = \sqrt{\frac{\tau_w}{\rho}}$ is the shear velocity, ν is the kinematic viscosity and y is the distance to the wall.

2.3.7 Wall functions

In order to model the flow bounded by a wall, a large number of computational cells are usually used to resolve the innermost boundary layers, as seen in Figure 2.2a. This is however very demanding in terms of processing power, since the number of computational cells escalates with increased resolution. Wall integration of turbulence models requires that the first computational cell outside of the wall should be located at a distance around $y^+ = 1$, which is within the viscous sublayer [31].

An alternative to this approach is the implementation of wall functions. This method, proposed by Launder and Spalding [32], is meant to sidestep the excessive grid resolution by allowing the first computational cell to be positioned in the logarithmic region, see Figure 2.2b. A relatively coarse mesh, $y^+ \geq 30$, that do not resolve the innermost boundary layers, is required for this approach to work. Wall functions uses semi-empirical formulations based on von Kármán's [33] law-of-the-wall, which states that the average velocity of a turbulent flow at a certain point is proportional to the logarithm of the distance from that point to the wall. The wall functions can generally be used for high-Reynolds number flows without a significant loss in accuracy, however it might have difficulties depicting complex flow, such as separation and reattachment. Furthermore, geometries with strong curvature can be problematic for the wall functions. Despite these shortcomings, wall functions are widely used for simulations today due to the reduced required processing power, as a result of the decreased number of computational cells.
Lastly, a combination of the two methods mentioned, where the software calculates the y^+ value for a cell and determines whether wall functions should be applied or not. Wall functions are applied if the first cell has an y^+ that exceeds 30, i.e. if the cell is located in the logarithmic region, as seen in Figure 2.1.



(a) Resolved grid near the wall ($y^+ = 1$)

(b) Grid with wall funtions applied ($y^+ = 30$)

Figure 2.2: Different approaches to model the flow near a wall. From [31].

2.3.8 Drag force coefficient

The drag force coefficient is a dimensionless quantity used to quantify the resistance of an object in a fluid environment, such as air or water. It is used in the drag equation, Eq. (2.15), where a lower drag coefficient implies that the object will have less aerodynamic or hydrodynamic resistance. The drag force basically consists of two different contributions, a viscous force due to surface friction and a pressure force due to the shape of the object.

$$C_d = \frac{F}{\frac{1}{2}\rho U_\infty^2 A} \tag{2.15}$$

The drag equation shows how the drag force, F, is divided by the dynamic pressure, $\frac{1}{2}\rho U_\infty^2$, multiplied with the projected area, A. The drag force coefficient formula can use drag forces, F_X, F_Y and F_Z, and the drag areas appropriate to each direction, to form the drag force coefficients C_X, C_Y and C_Z, respectively.

2.4 Software

This section briefly describes the main software packages used during the project.

2.4.1 OpenFOAM

OpenFOAM (Open source Field Operation And Manipulation) [34] is an open source software bundle for CFD. The software bundle is first and foremost a C++ library used to create executables, known as applications. The applications can be divided into two groups: utilities, which have the purpose of executing tasks that concern manipulation of data; and solvers, which are designed to solve a specific problem in continuum mechanics. The OpenFOAM package includes various solvers and utilities covering a large spectrum of problems. One great advantage of OpenFOAM, is the possibilty of incorporating custom-made applications into the existing library of utilities and solvers, without impairing the compatibility with existing ones. Another reason for working with OpenFOAM, is that there are no parallel license costs, since it is open source. This, in combination with the possibility of running all codes in parallel, makes OpenFOAM attractive for academic and industrial purposes alike, since it provides an opportunity to simulate problems with greater complexity more accuractly and quickly, for free.

OpenFOAM is equipped with both pre- and post-processing environments. Pre-processing utilities are tools used to prepare the simulation cases while the post-processing utilities process the results of the simulation cases. The global overview of OpenFOAM is shown in Figure 2.3:

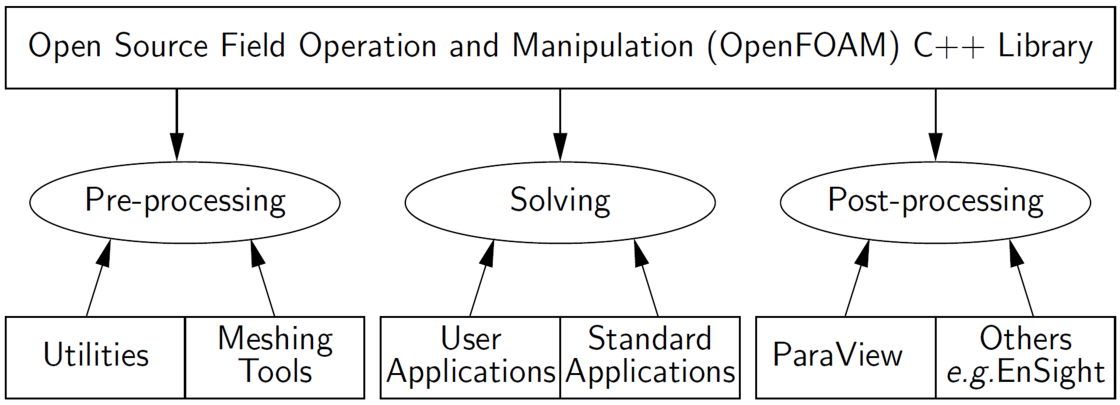


Figure 2.3: Global overview of the OpenFOAM structure, extracted from the OpenFOAM user guide [34].

2.4.1.1 Solvers

Solvers come in great numbers for many different applications. The solvers mentioned here are the ones relevant for this project.

potentialFoam

The potentialFoam solver is a basic solver for potential flow. Its use is mainly oriented to generate a converged initial field for subsequent simulations.

simpleFoam

simpleFoam is solely a RANS solver for incompressible steady-state simulations of turbulent flow. It is, as the name already suggests, using the SIMPLE algorithm for solving the pressure-velocity coupling. This solver needs the specification of a turbulence model since this solver is a RANS solver, see section 2.3.4.

pimpleFoam

The pimpleFoam solver is using the PIMPLE (merged PISO-SIMPLE) algorithm for the pressure-velocity coupling. It is a transient solver capable of handling large timesteps.

For any further information regarding utilities, solvers or OpenFOAM as a whole, please refer to the OpenFOAM user guide [34].

2.4.2 ANSYS Workbench

The ANSYS Workbench platform is a framework which integrates simulation technologies with pre- and post processing applications, with a convenient drag-and-drop GUI. There are many applications hosted by ANSYS Workbench, here however, only the ones of importance of this project are presented briefly.

ANSYS DesignModeler

Every CFD simulation starts off by defining the geometry representing the design. ANSYS DesignModeler allows the user to either build its own geometry from scratch, or import an existing CAD geometry fully parametric. If necessary, it is even possible to adjust and modify imported models, since CAD models are often designed for other purposes than simulation. The software also specifies the computational domain.

ANSYS Meshing

This is a highly automated meshing environment which makes it easy to generate a proficient mesh.

ANSYS Fluent

ANSYS Fluent, henceforth called Fluent, is a commercial CFD code with a large variety of engineering applications, such as multiphase and reacting flow as well as hydro- and aerodynamics. The software is also used by GVA.

2.4.3 Other software

Paraview

ParaView is an open-source, multi-platform visualization and data analysis application. During this project ParaView was used to visualize and analyze both the results of OpenFOAM and Fluent.

MATLAB

MATLAB is a numerical computing environment widely used across the industry and the academia. The software is versatile and capable of matrix manipulations, implementation of algorithms and plotting of functions and data, for instance. Another feature is the possibility to interact with programs written in other languages, such as C, C++, Java, Fortran and Python. In this thesis however, MATLAB was mostly used to assess and visualize the simulation outputs.

Chapter 3

Experimental Setup

In the interest of investigating the influence that wind and current loads have on the design aspects of an offshore platform, a series of wind tunnel testings have been performed. The number of set-ups are many, as testings have been undertaken for both hull and topside; at different draughts, heel angles and different wind- and flow directions. For this thesis just one of the set-ups will be used in the upcoming CFD simulations, namely the one where the current loads on the hull at even keel in head-on wind during operational draught, are calculated.

3.1 The wind tunnel

The reference wind tunnel has a working section of 4.8 metres wide by 2.4 metres high, and the length of the tunnel is 19 metres. During the expermental measurements the operating wind speed was 25.9 m/s. The wind tunnel model hull was built to a scale of 1:200. To compensate the difference in Reynolds number between model and full-scale flow conditions, a selected roughness treatment was applied to the circular columns. In order to minimize the boundary layer effects on the model test, a boundary layer reducing fence was installed upstream of the model. A visualization of this set-up is shown in Figure 3.1 below.

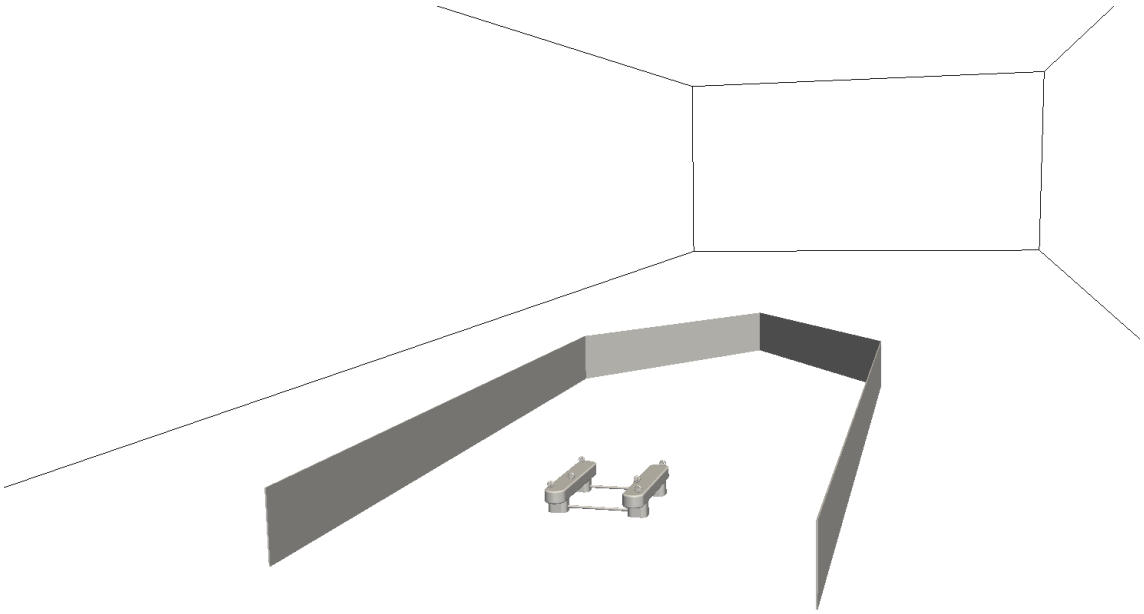


Figure 3.1: The set-up of the wind tunnel with the hull and boundary layer fence.

It is important to consider the effects of the boundary layer, as the reduced local velocites will lead to miscalculated load measurements. The impact of the boundary layer effects can be reduced by introducing an obstacle to the path of the fluid flow, such as the boundary layer fence, in which the fluid will become separated from the surface of the floor, thus breaking the boundary layer and allowing it to reform some distance before the flow reaches the model. In Figures 3.2 and 3.3 one can see schematic sketches of how the boundary layer fence works. In the first figure, the red zone depicts the area where the fluid has been detached from the fence. After some distance the fluid is reattached to the floor's surface, and a boundary layer is being formed once again. The second picture aims to demonstrate how a new boundary layer is being formed, by showing the velocity profile prior to and after the fluid has passed the fence.

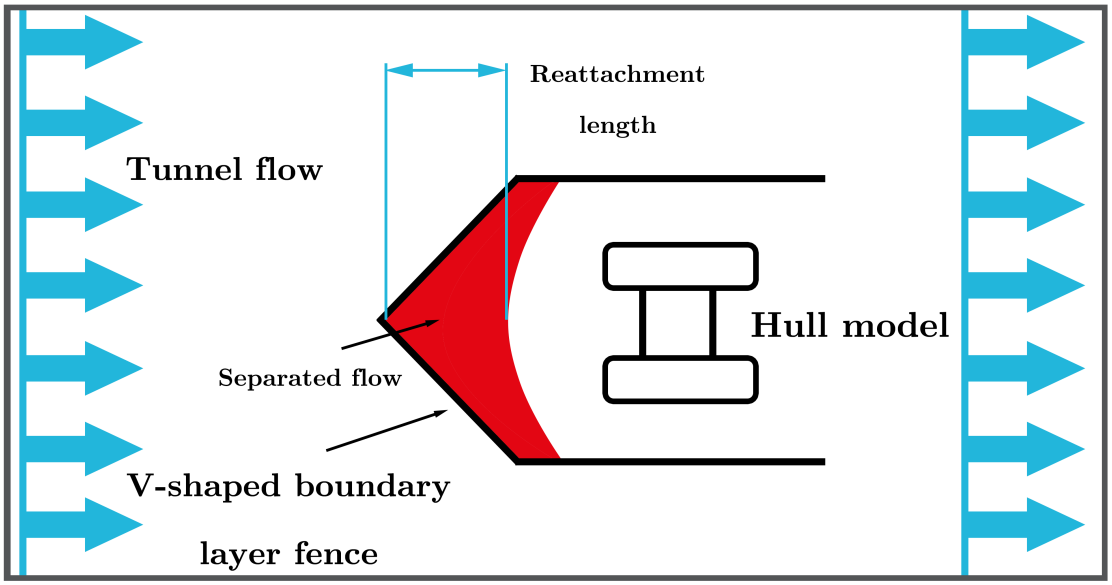


Figure 3.2: The principle of the boundary layer fence, as seen from above.

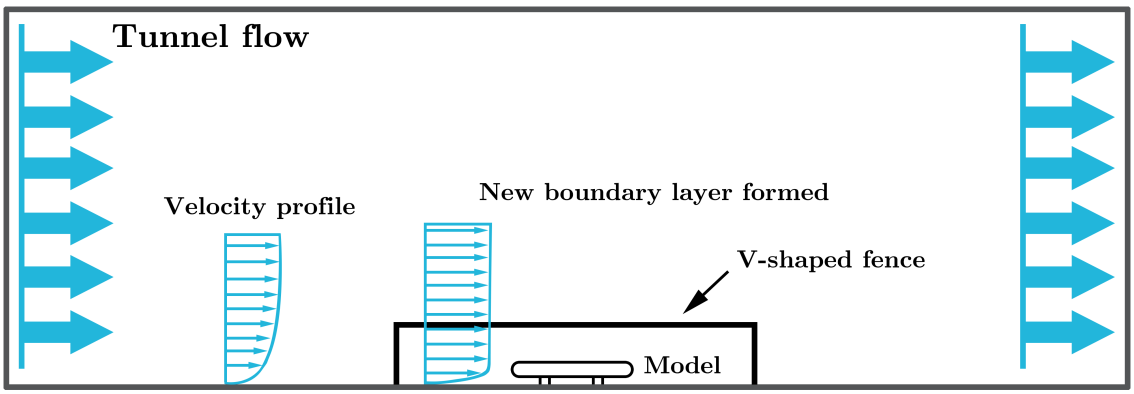


Figure 3.3: The principle of the boundary layer fence, as seen from the side.

Although the boundary layer has been decreased, it will still affect the results to some degree, as it has potential to underestimate the horizontal forces. In order to

compensate for this, an adjustment factor can be applied to the measured results, which has been done in this thesis.

3.2 Presentation of the cases to investigate

The first case will simulate the entire wind tunnel, including the boundary layer fence. Steady-state RANS simulations will be performed in both OpenFOAM and Fluent, using the same geometry.

The second case will run a truncated near-field simulation, without the boundary layer fence. In order to obtain "ideal conditions", a slip condition on the floor will be set. A grid dependency test will also be done. The simulation will run in Fluent and will compare two different turbulence models. Otherwise the simulation settings are the same as those in the previous scenario.

Chapter 4

Simulation of the entire tunnel

This section explains the specific setup for the simulation of the whole wind tunnel.

4.1 Geometry

The computational domain in this case, with symmetry applied, spans 19 m longitudinally and 2.4 m both horizontally and vertically. The boundary layer reducing fence and the model are positioned 10 and 13.664 m downstream, respectively. This setup can be seen in Fig. 4.1 below.

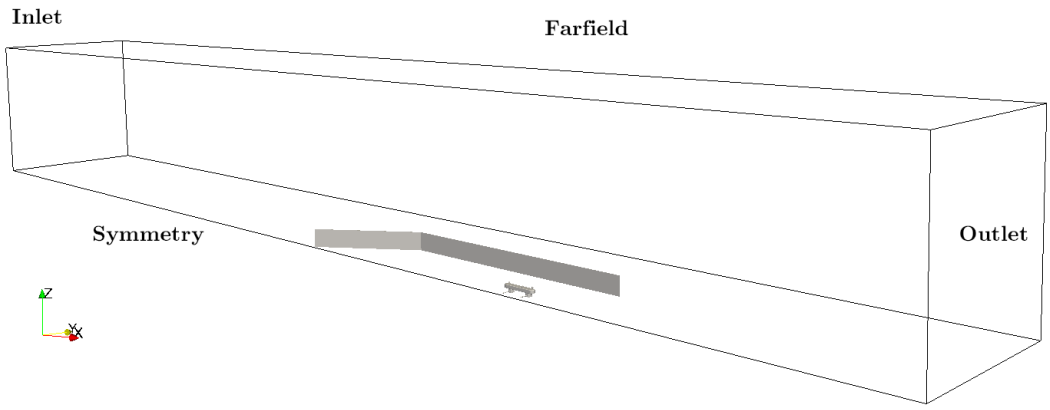


Figure 4.1: Computational domain of the wind tunnel.

The mesh in this case is made in ANSYS Meshing and is composed of $8.2 \cdot 10^6$ cells. Three inflation layers have been applied to the surfaces of the model and the floor and the y^+ values of the mesh can be seen in Table 4.1.

Table 4.1: y^+ values of the surfaces.

Surface	Average y^+	Max y^+
Model	11.11	39.96
Floor	163.65	327.73
Fence	91.68	1020.40
Total	88.40	1020.40

As told by the table, the fence is not very resolved. This is simply because we are interested in the flow after it has passed the fence, not at what is going on with the flow at the surface of the fence.

In the area around the model's body the mesh is much more refined, as can be seen in Figures 4.2 and 4.3. This refined area of the mesh continues behind the model, as

it is of importance to have a refined rear end in order to study the wake flow behind the structure, which will be of significance for the drag prediction.

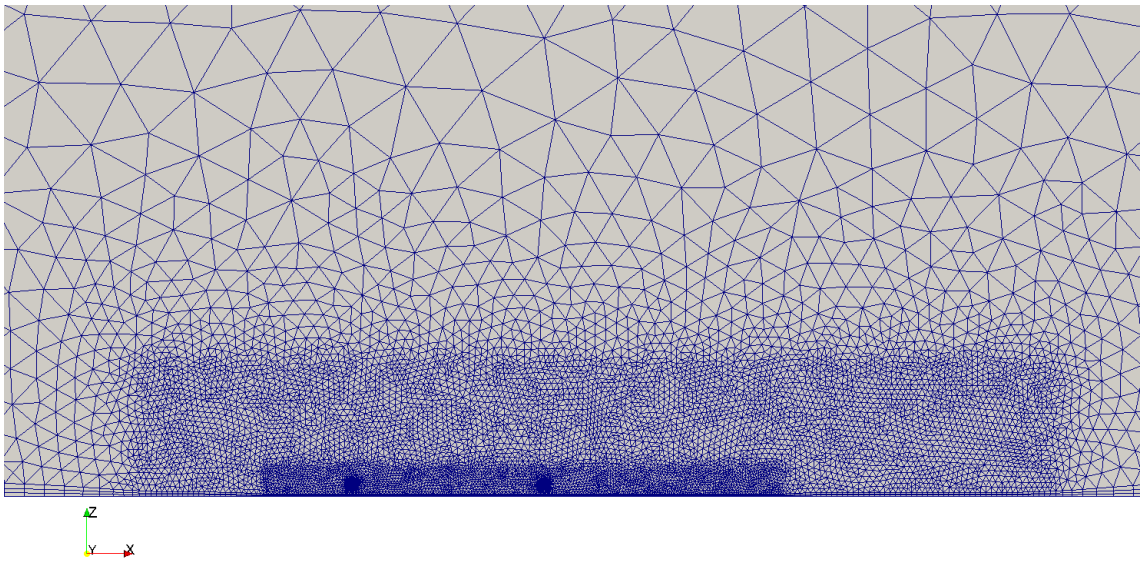


Figure 4.2: View of the mesh around the model, at $y = 0$.

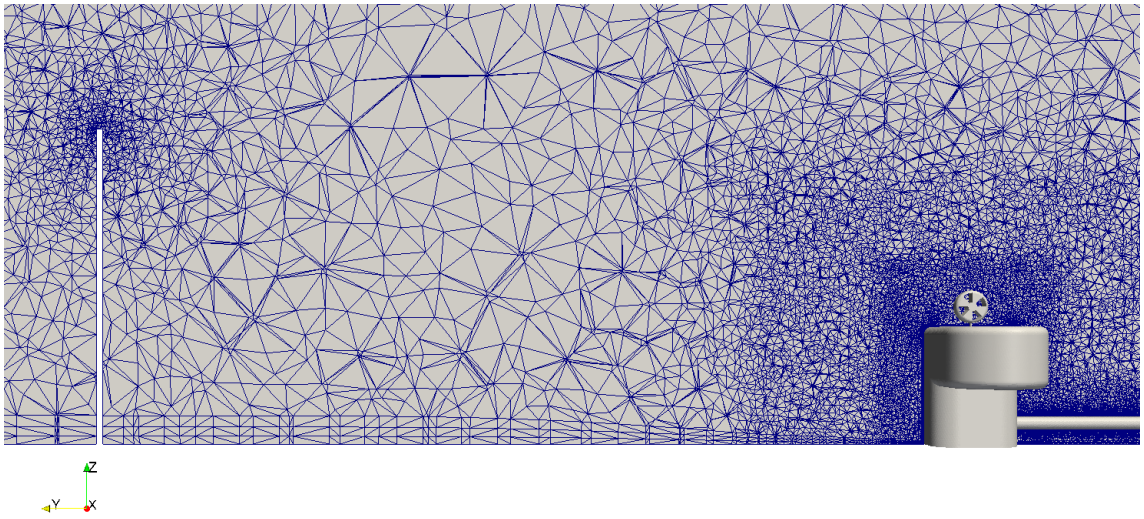


Figure 4.3: A cut through the domain, showing the mesh around the model and the fence.

4.2 Approach

The first measure after the case has been set-up, is to initialize the flow field to provide a suitable set of initial conditions for the RANS solver. In OpenFOAM this is done by the potential flow solver of potentialFoam, while Fluent uses Hybrid initialization to determine the velocity and pressure fields.

When the flow field has been initialized, the RANS simulation in turn can be launched. This case is solved in steady-state and both OpenFOAM and Fluent will be using the SIMPLE algorithm, with $SST\ k-\omega$ as the turbulence model.

4.3 Simulation settings

The BCs are set in the following manner: at the inlet, a fixed velocity in the streamwise direction (x-direction) of 25.9 m/s is defined. The turbulence kinetic energy, k, and the specific dissipation, ω, are set to 0.3 and 1.7 respectively. Physical boundaries, such as the model, fence and ground are all treated as walls. The farfield on the other hand is treated as surfaces with slip condition. At the outlet the pressure is set to zero.

A linear Gauss scheme is used for the discretization of the diffusion terms. As for the convective terms, Fluent is discretizing the momentum and the turbulent properties using a second order upwind scheme, while OpenFOAM is using a first order upwind for these.

The under-relaxation factors (URF) for both simulations were the same, namely

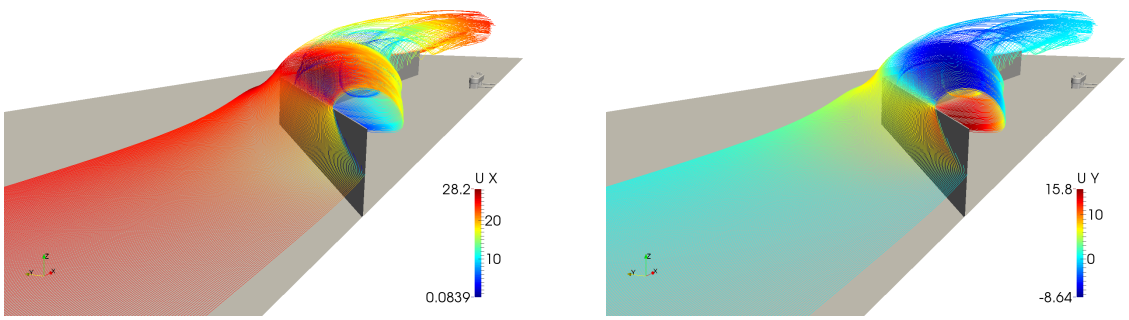
Table 4.2: Under-relaxation factors for the simulation of the entire wind tunnel.

Property	p	U	k	ω
URF	0.3	0.7	0.7	0.7

4.4 Results

4.4.1 Flow results

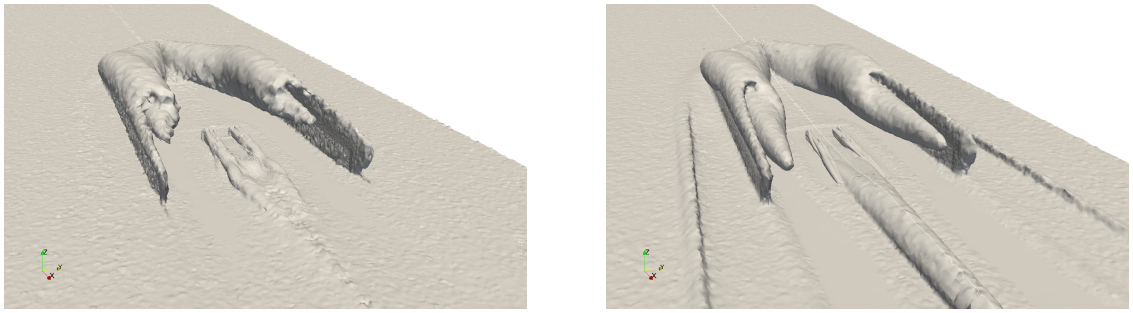
The boundary layer fence gives rise to two large vortices, which are travelling along the walls of the fence, on each side of the model, as seen in Figures 4.4.



(a) Velocity in x-direction **(b)** Velocity in y-direction

Figure 4.4: Streamlines in terms of velocity, Fluent simulation.

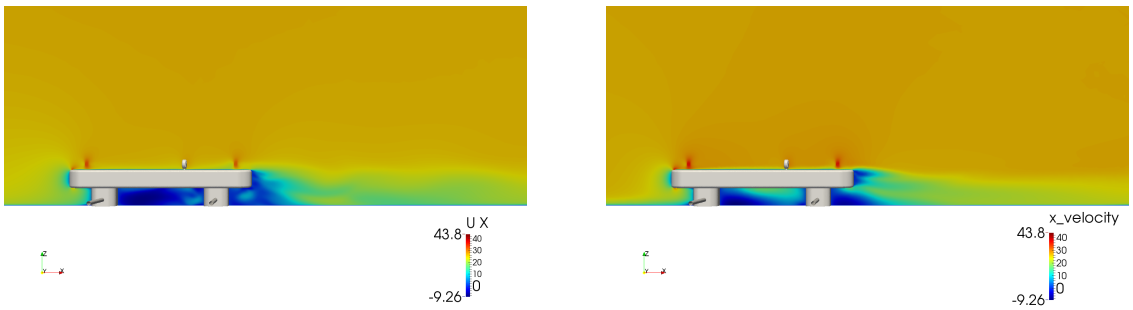
The flow is much more defined in the simulation performed by Fluent, which is only natural since it uses a higher order of discretization for the convective terms compared to OpenFOAM. This can clearly be seen in Figure 4.5, as the vorticity contour of Fluent is much more smooth and detailed.



(a) OpenFOAM **(b)** Fluent

Figure 4.5: Iso-surface of vorticity magnitude.

When examining the velocity, it also points towards a more defined and detailed flow for Fluent. Figure 4.6 shows the velocity in the x-direction and especially the flow between the columns are looks more realistic for Fluent.



(a) OpenFOAM **(b)** Fluent

Figure 4.6: Velocity field in the x-z plane, intersecting the pontoon and columns.

The flow in the y-direction seems to be having a suction effect on the model, see Fig 4.7 and Fig 4.8.

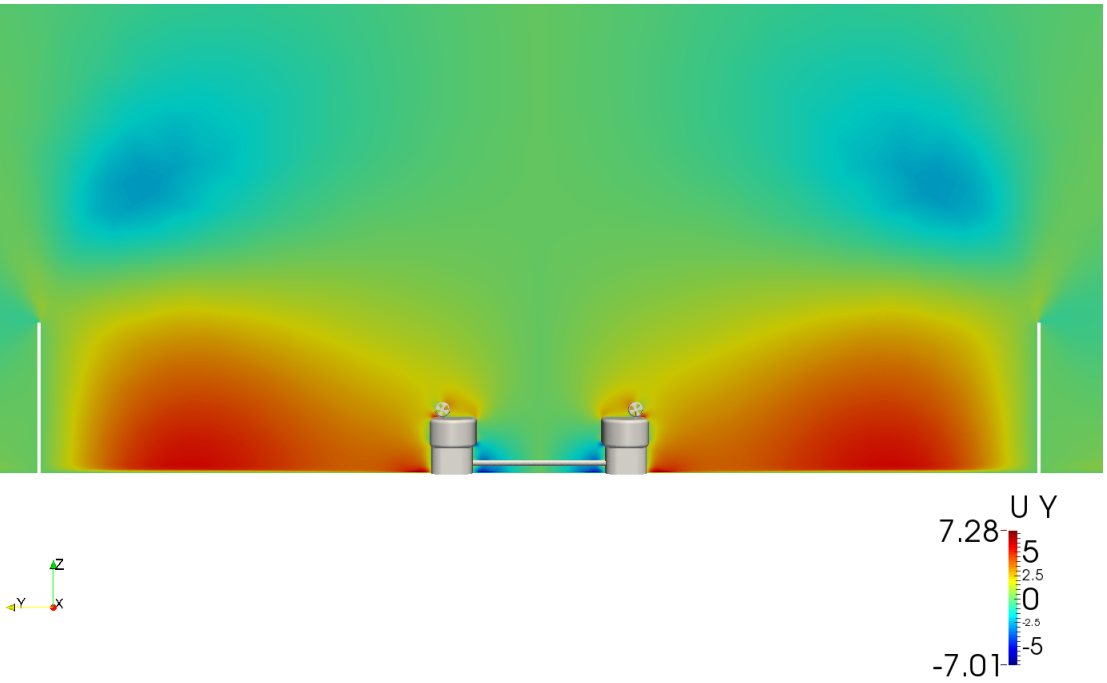


Figure 4.7: The velocity in the y-direction from the Fluent simulation. Note that the right side is mirrored.

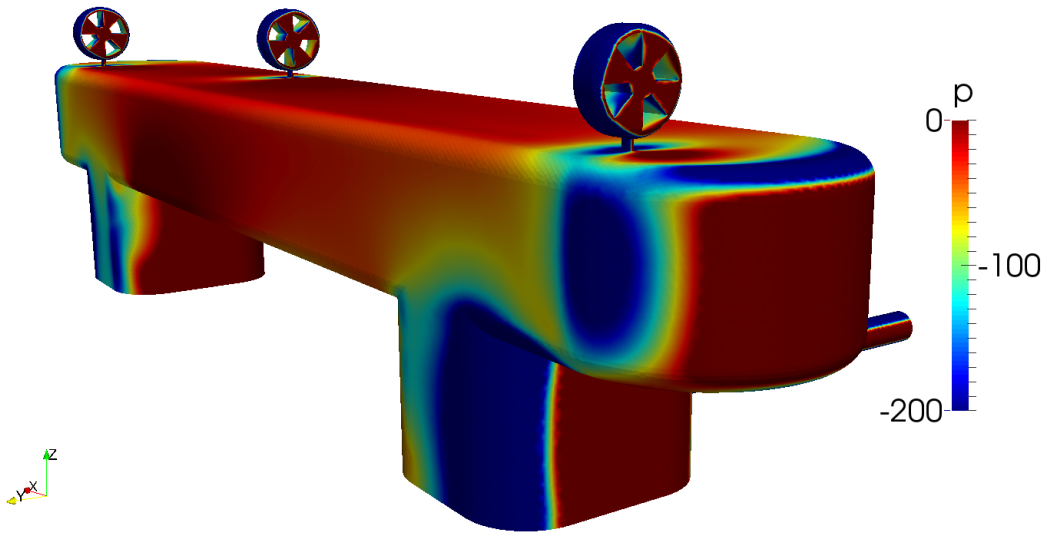


Figure 4.8: The pressure acting on the left side of the model, from the Fluent simulation.

4.4.2 Drag and force results

The measured drag (in the x-direction) can be seen in Figure 4.9. It clearly shows how the drag results of OpenFOAM and Fluent are pretty much equal. However,

they are both far from the experimentally measured value, reaching just roughly 60% of that measured value.

The side force, F_Y, from the simulations measures $F_Y \approx 1.38N$. As the simulations are done with half the model, the result can not be compared with experimental data, since this was measured using the whole model. Instead it will be compared with the result of the following case, where simulations without the fence will be performed.

The flow appears to approach the model at an angle, as seen in Fig 4.10. This will probably affect the measured values, maybe not the forces acting in the x- and y-direction, but most definitely in the z-direction. It is likely that the region where the fluid has been separated from the boundary layer fence should be better resolved in order to counter this.

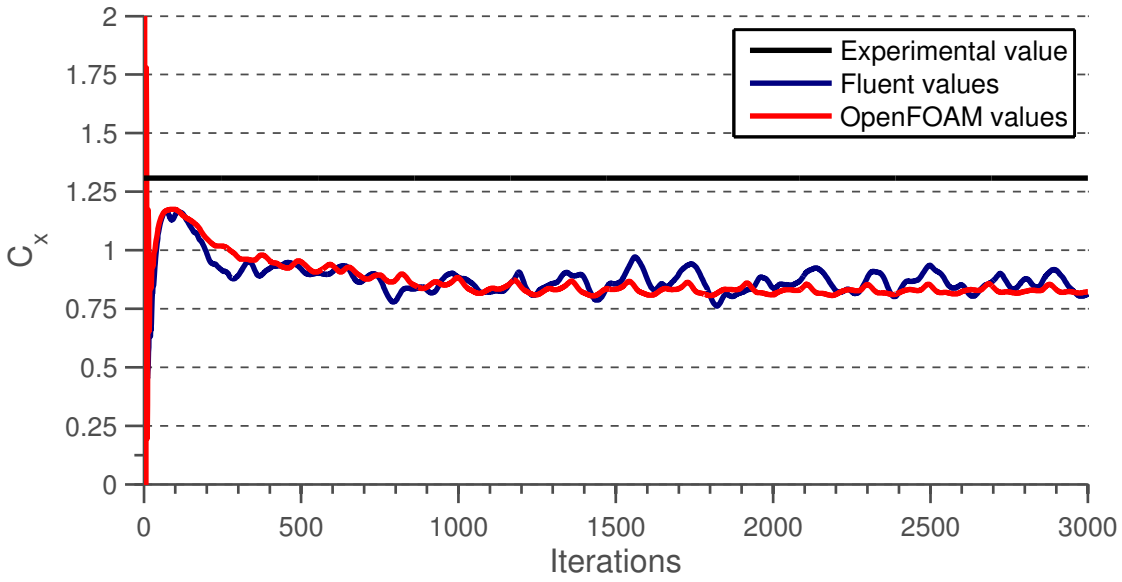


Figure 4.9: Drag values in the x-direction for Case 1; red line: OpenFOAM values; blue line: Fluent values; black line: experimental value

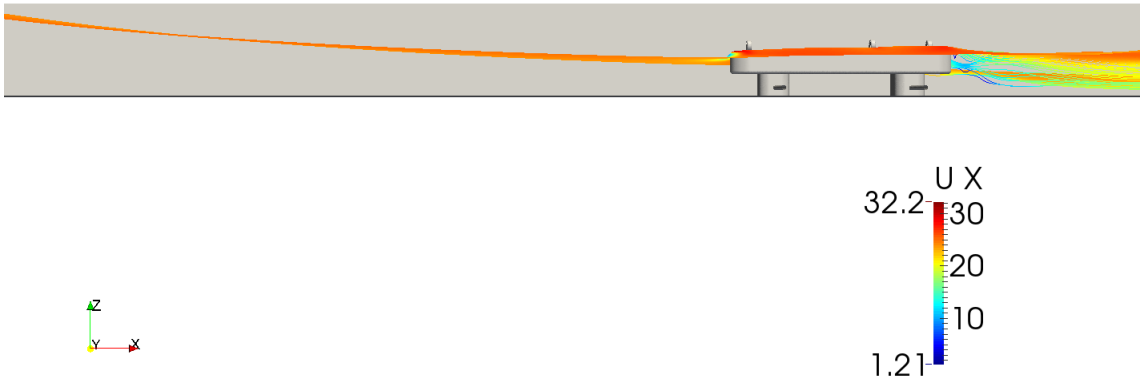


Figure 4.10: Streamlines in terms of x-velocity interacting with the model. From the Fluent simulation.

Chapter 5

Near-field simulation

For this case, the boundary layer fence is excluded in exchange of a smaller domain and assuming slip condition on the floor. In this fashion, one can achieve the condition which the boundary layer fence is aiming to establish, by using fewer computational cells. This also provides a good opportunity to compare the side forces affecting the body, without the influence of the boundary layer fence.

5.1 Geometry

As mentioned, the model is being studied without the boundary layer fence. The computational domain in this case is thus smaller than the previous one, since only the model is being studied, see Fig 5.1. More specifically, the domain is 6.87 m long, 0.836 m wide and 2.3 m tall. The model is located 2.204 m downstream.

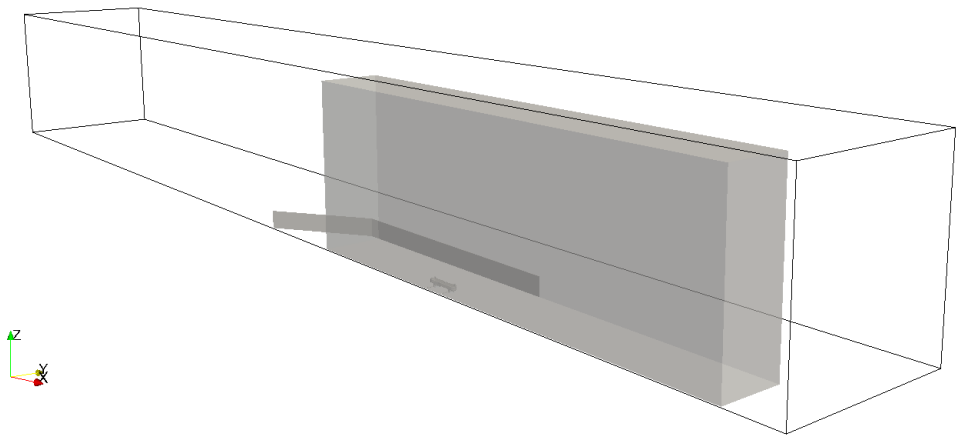


Figure 5.1: The opaque grey box is the new computational domain.

The refinement around the model is the same as in the previous case. A series of meshes have been generated in ANSYS Meshing in order to make a grid dependency test. The difference between the meshes is basically different sizes of the largest cells of the meshes. In Table 5.1 the meshes are listed, showing their number of computational cells and y^+ values. The y^+ values are referring to the values of the model, as the floor is treated with a slip condition.

Table 5.1: Meshes used in the grid dependency test.

Grid	No. cells	Average y^+	Max y^+
Coarse	2M	10.56	70.99
Medium	4M	10.42	78.22
Fine	8M	10.54	70.42
Very Fine	16M	10.52	70.75

5.2 Approach

The purpose of this case is to study the model more thoroughly without the boundary layer fence. influence of mesh density and choice of turbulence model.
Two turbulence models; the $SST\ k-\omega$ and a Realizable $k-\epsilon$ with non-equilibrium wall function treatment, are being used in this grid dependency study along with a laminar solver. The simulations, which are steady-state RANS, will be run by Fluent alone. Just as in the previous case, the SIMPLE algorithm will be used.

5.3 Simulation settings

In this case the floor is treated with a slip condition, as mentioned earlier. This way, the flow will uniformly approach the model, which is ideal, and what the boundary layer fence is trying to reproduce. Other settings are as before.

The solver settings are the same as before, namely a linear Gauss scheme is used to discretize the diffusion terms. As for the convective terms, both the momentum and the turbulent properties are using a second order upwind scheme.

The under-relaxation factors are also the same, as seen below.

Table 5.2: Under relaxation factors for the near-field simulation.

Property	p	U	k	ω
URF	0.3	0.7	0.7	0.7

5.4 Results

The grid dependency study shows that the results are not changing much when increasing in mesh density for instance. There are differences in the results for the different turbulent models, however, both of them are significantly lower than those of the laminar solver. Since the results drop when switching on turbulence, it might be an indication that the problem requires as a transient simulation to be properly solved.

Table 5.3: Grid dependency results in terms of the percentage of x-direction drag force coefficient of the experimental value.

(a) Laminar

Grid	C_x [%]
C	70.46
M	68.91
F	70.94
VF	72.65

(b) $SST\ k-\omega$

Grid	C_x [%]
C	57.67
M	54.98
F	57.42
VF	58.58

(c) Realizable $k-\epsilon$

Grid	C_x [%]
C	46.91
M	44.83
F	43.45
VF	43.79

The results of the $SST\ k-\omega$ simulations yield basically the same value in both this case and the previous case. This is good since it requires a smaller domain and mesh to simulate just the model without the boundary layer fence. The results of the other turbulence model, Realizable $k-\epsilon$, deviates quite alot from those of $SST\ k-\omega$. This indicates that the choice of turbulence model might be crucial.

The measured side force in this simulation, which is without the fence, is $F_Y \approx 0.007N$, which is a lot smaller than the $1.38N$ that was measured in the simulation with the fence. This however then indicates that the fence has a large impact on the flow over the model.

Chapter 6

Discretization of the convective term

During the first case of simulating the entire wind tunnel, OpenFOAM had trouble achieving convergence when applying a second order discretization of the convective term. This case is meant to investigate this further.

6.1 Geometry

The domain is the same as in Case 2, however, it is now generated in OpenFOAM, via blockMesh and snappyHexMesh. In Figures 6.1 and 6.2 the outline of the mesh and its refinement can be seen.

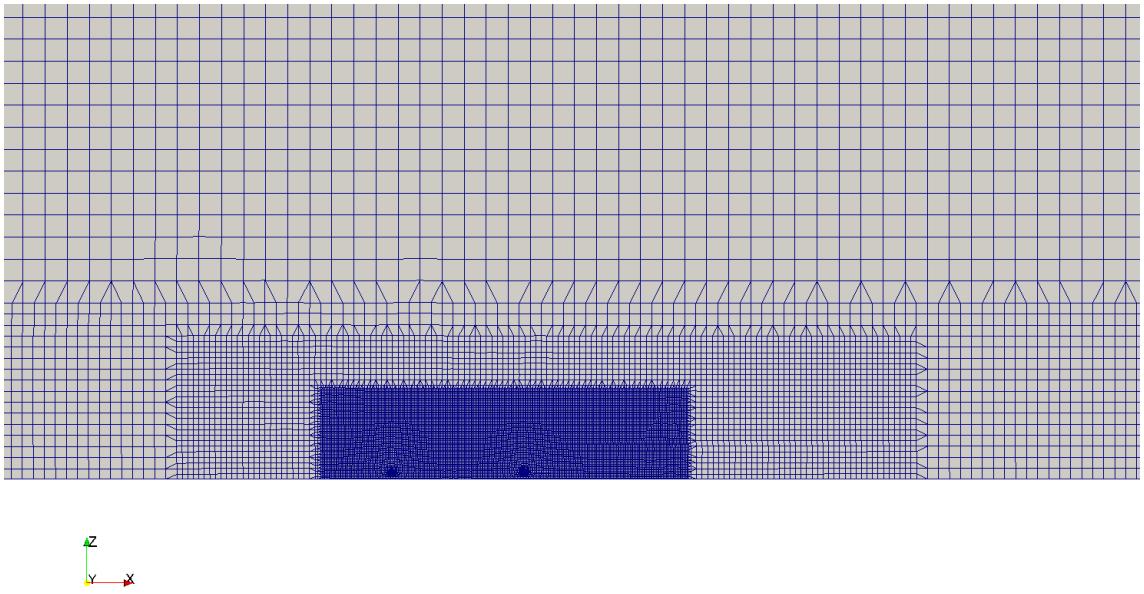


Figure 6.1: View of the mesh around the model, at $y = 0$.

Three meshes have been produced in order to be able to perform any grid dependency studies, as seen in Table 6.1.

Table 6.1: Computational meshes created in OpenFOAM.

Grid	No. cells	Average y^+	Max y^+
Coarse	1.275M	26.72	172.80
Medium	1.7M	26.31	149.19
Fine	2.55M	26.65	156.52

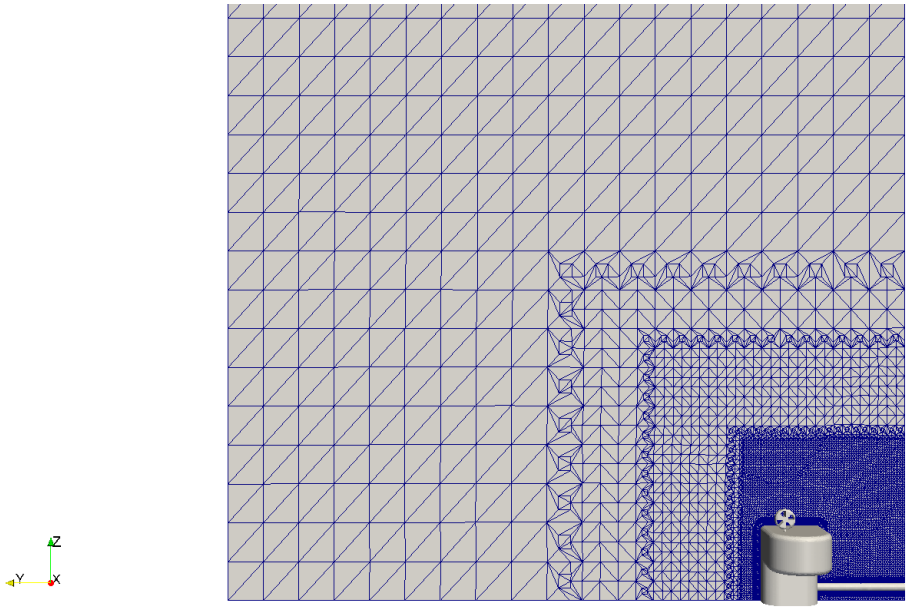


Figure 6.2: A cut through the domain, showing the mesh around the model.

6.2 Approach

Steady-state, $SST\ k-\omega$ RANS simulations will be performed for four different settings. Two simulations will apply first order discretization of the convective terms - where one case will use low URF and the other more normal values. The remaining two simulations use second order discretization, where one just discretizes the momentum as second order and the other discretizes both the momentum and the turbulence properties. The second order simulations will use the converged first order data as initial values.

6.3 Simulation settings

The BCs are identical to the ones of the previous near-field simulation.

The specific settings for the four simulations are as follows,

- **First order with low URF.** The gradients are discretized using a Gauss linear scheme and the convective terms are discretized using the first order upwind scheme. URF as follows

Table 6.2: Under-relaxation factors for the first order with low URF.

Property	p	U	k	ω
URF	0.1	0.3	0.3	0.3

- **First order with regular URF.** Same as above but with the values of URF as in the previous cases.
- **Second order momentum** Here a second order linear upwind is being used for the momentum. URF are also the same as in the previous cases.
- **Second order momentum and turbulence** Linear upwind is applied for turbulence properties as well as the momentum.

6.4 Results

By using first order schemes with low URF, it was possible to achieve a drag coefficient of 88% of the experimental value. However, as soon as a second order scheme is used, the residuals and the drag coefficient is reduced down to a level of about 68%, with poor residual values. As the second order schemes are more accurate, it is very strange that the first order simulation yields better results. The drag coefficient (of the first order simulation) seems to be decreasing with increasing mesh density, as can be seen in Table 6.3, so this may be part of the explanation.

Table 6.3: Grid dependency study of the first order, low URF simulation.

Grid	No. cells	C_x[%]
Coarse	1.275M	88.30
Medium	1.7M	87.61
Fine	2.55M	85.24

Chapter 7

Unsteady simulations

Two transient simulations, one URANS, the other LES, was performed at the end of the project.

7.1 Geometry

The mesh in this case is created in OpenFOAM and consists of $2.7 \cdot 10^6$ cells. An LES simulation would certainly benefit from having more cells, but due to time and resource limitations a larger mesh could simply not be used.

7.2 Approach

First a steady-state $SST\ k-\omega$ RANS simulation will be performed, for both comparison sake and as initiation for the transient simulations. All simulations will be performed in OpenFOAM.

7.3 Simulation settings

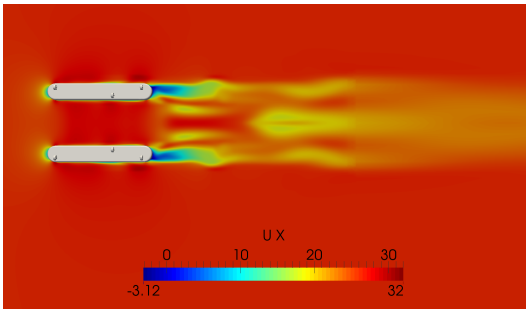
Even the transient simulations had trouble to achieve convergence using second order schemes. Thus first order schemes for the convective terms was used for all simulations.

7.4 Results

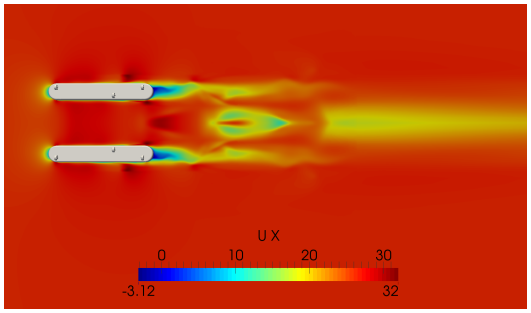
The transient simulations show the same tendencies as the steady-state ones of the previous cases; they cannot converge without first order schemes. When running on first order schemes, they end up with similar values as those of steady-state simulations in OpenFOAM.

The x-direction drag force coefficient ended up as $C_X = 76.7\%$ for LES and $C_X = 74.6\%$ for URANS. The RANS simulation prior to these had a drag coefficient of, $C_X = 71.8\%$.

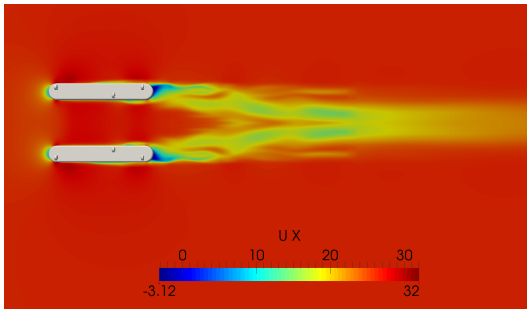
In the following plots, the velocity field is shown for LES, RANS and URANS in two planes, as well as a plot of the pressure field. It can be seen how the flow of LES looks most realistic, as it should. In the velocity plots for RANS, it is quite evident that the wake probably should be better resolved, as it is quite blurry.



(a) RANS

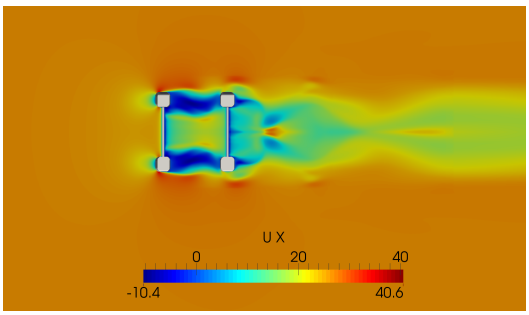


(b) URANS

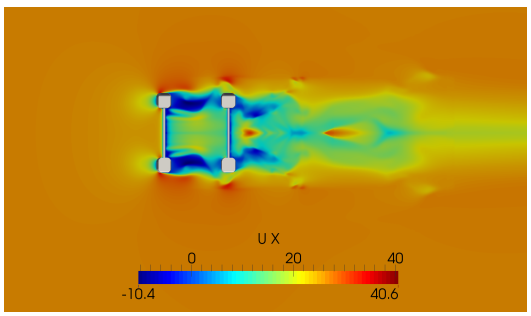


(c) LES

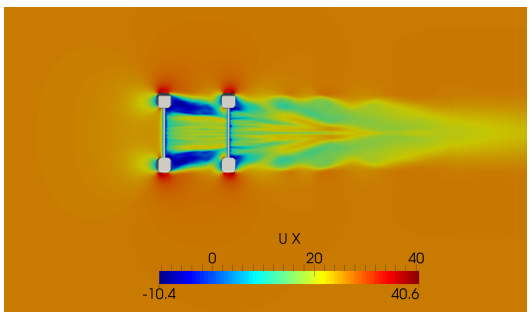
Figure 7.1: Velocity field in terms of x-velocity in the x-y plane, intersecting the pontoons of the model.



(a) RANS

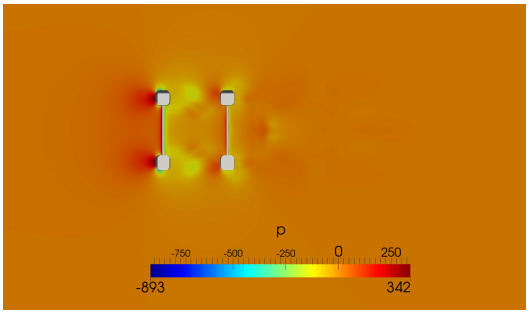


(b) URANS

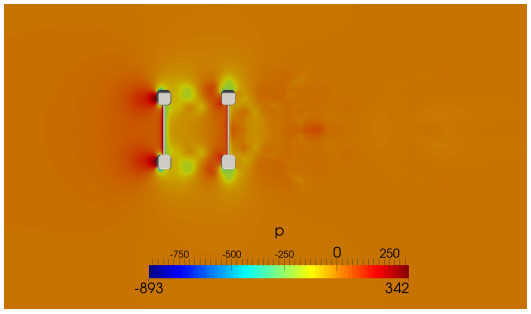


(c) LES

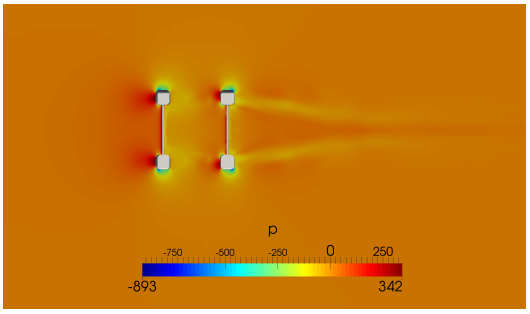
Figure 7.2: Velocity field in terms of x-velocity in the x-y plane, intersecting the legs and bracings of the model.



(a) RANS



(b) URANS



(c) LES

Figure 7.3: Pressure field in terms in the x-y plane, intersecting the legs and bracings of the model.

Chapter 8

Conclusions and recommendations for future work

8.1 Conclusions

This work was aimed to validify the use of CFD on the hull of a semi-submersible offshore structure. Two CFD codes have been used, a commercial code of Fluent and an open source code of OpenFOAM. The simulations are meant to recreate a physical experiment that has been conducted in a wind tunnel, where the even keel current loads have been measured. It is of interest to explore if there are instances where steady-state simulations can be performed. Furthermore, in the conducted experiment, a boundary layer fence was installed upstreams of the model, in order to reduce the boundary layer effects on the measured results. There was however a suspicion, that vortices, generated when the fluid interacts with the fence, might influence the measured forces. The study underwent four stages.

In the first stage, the wind tunnel experiment was reenacted in both OpenFOAM and Fluent, using the same mesh for both codes. The simulations were steady-state, using $SST\ k-\omega$ as turbulence modeler, and showed that discretization schemes have a large impact on OpenFOAM, as it could not converge using second order schemes for the convective terms. None of the codes were able to match the experimentally measured drag coefficient. One possible reason could be that mesh needs additional refinement. For instance more inflation layers could be added to resolve the boundary layer better, as well as refine the mesh upstreams, where the fluid is separated from the fence, and the wake flow downstreams of the model. Another reason could be that the problem requires it to be solved as a transient problem.

The following case was a mesh dependency study - investigating the influence of choice of turbulence model and mesh grid density. Instead of using the entire wind tunnel domain, as in the previous case, just the region within the boundary of the fence was used, with a slip condition on the ground. The case was conducted in Fluent and used two turbulence models, $SST\ k-\omega$ and Realizable $k-\epsilon$ with non-equilibrium wall functions, as well as a laminar solver. They produced different results, where the $SST\ k-\omega$ results were almost the same as the ones of the previous case, and the Realizable $k-\epsilon$ results were significantly smaller. It is difficult to tell why without investigating it further, but it indicates how important the choice of turbulence model can be. Both models yield a lower value compared to the laminar results. This perhaps signals that the problem, yet again, must be treated as time-dependent. The measured side force was much lower in this simulation without the fence, implying that the fence might affect the flow around the model.

Case number three investigates the influence that the choice of discretization of the convective terms have in OpenFOAM for steady-state simulations. In case one, it was found that the simulations could not converge properly by using second order schemes, linear upwind to be exact. Using a mesh created in OpenFOAM, simulations with four different settings of discretization were performed. The results, yet again, show that the drag coefficient is dropping once a second order scheme is used. Simulations with first order schemes, upwind, showed to be clearly mesh dependent, as the coefficient decreases with increasing mesh density. Upwind schemes are sensitive to false diffusion, especially at low mesh densities. It is likely that the coefficient is overestimated due to the false diffusion.

The last case is studying the effect of solving the problem using the transient methods of URANS and LES. Unfortunately, these simulations were performed at the very end of the project, and as proper transient simulations requires an equal proper mesh, these results were more indicative, as they were performed with a relatively small mesh and using first order upwind schemes. They both however yielded similar drag coefficient to previous RANS.

8.2 Recommendations for future work

- The meshes require further refinement. Especially important is the wake, as the wake flow is crucial for the estimation of the drag resistance. But also the number of inflation layers could be increased, in order to better capture the boundary layer.
- A proper transient simulation should be performed, preferably using LES, since the wake flow for certain is fluctuating. It is probably possible to achieve a good estimation using RANS as well, but more work is needed to prove this.
- The experimental data in this work only submitted a total resistance for the whole body. By giving information for each component of the body, it is easier to find which instances of the body where work needs to be done in order to make RANS work.
- There is the question of experimental certainty as well.

Bibliography

[1] J. P. Hooft. *Hydrodynamic aspects of semi-submersible platforms*. PhD thesis. Wageningen University, 1972.

[2] GVA Consultants. *The GVA 8000 - Semi-Submersible Drilling Rig*. Retrieved May 2015 from: http://www.gvac.se/Documents/GVA_8000_Drilling_Unit.pdf

[3] DNV. *Environmental Conditions and Environmental Loads*. Recommended Practice DNV-RP-C205. Norway, 2007.

[4] DNV. *Global Performance Analysis of Deepwater Floating Structures*. Recommended Practice DNV-RP-F205. Norway, 2010.

[5] F. Van Walree and H. J. J. Van den Boom. *Wind, Wave, and Current Loads on Semisubmersibles*. In Proceedings of the 23^{rd} Annual Offshore Technology Conference, OTC 6521, Houston, 1991.

[6] J. L. Cozijn, B. Buchner, and R. R. T. van Dijk. *Hydrodynamic Research Topics for DP Semi Submersibles*. In Proceedings of the 31^{st} Annual Offshore Technology Conference, OTC 10955, Houston, 1999.

[7] S. K. Chakrabarti. *Handbook of Offshore Engineering, Volume II*. Offshore Structure Analysis, Inc. Plainfield, Illinois, USA, 2005.

[8] GVA Consultants. *Global Performance*. Retrieved May 2015 from: http://www.gvac.se/Engineering/Global-Performance/

[9] WAMIT. *Home page*. Retrieved February 2017.

[10] DNV GL. *Frequency domain hydrodynamic analysis of stationary vessels - Wadam*. Retrieved February 2017 from: https://www.dnvgl.com/services/frequency-domain-hydrodynamic-analysis-of-stationary-vessels-wadam-2412

[11] J. W. Kim, R. Izarra, H. Jang, J. Kyoung and J. O'Sullivan. *An application of the EOM-based numerical basin to dry-tree semisubmersible design.*. In Proceedings of the 19^{th} Offshore Symposium, Houston, 2014.

[12] C. Eskilsson, J. Palm, J. P. Kofoed, and E. Friis-Madsen. *CFD study of the overtopping discharge of the Wave Dragon wave energy converter*. RENEW 2014, Lisbon.

[13] J. W. Kim, H. Jang, A. Baquet, J. O'Sullivan, S. Lee, B. Kim, A. Read and H. Jasak. *Technical and Economic Readiness Review of CFD-Based Numerical Wave Basin for Offshore Floater Design*. OTC-27294-MS, Houston, 2016.

[14] S. Zhang, S. Yang, L. Wang, and H. Yang. *Numerical Study of Wind Loads on Semi-Submersible Platform by CFD*. In Proceedings of the ASME 2010 29^{th} International Conference on Ocean, Offshore and Arctic Engineering, OMAE2010-20209, Shanghai, 2010.

[15] G. Vaz, O. J. Waals, H. Ottens, F. Fathi, T. Le Souëf, and K. Kiu. *Current affairs: Model tests, semi-empirical predictions and CFD computations for current coefficients of semisubmersibles*. In Proceedings of the 28^{th} International Conference on Ocean, Offshore and Arctic Engineering, OMAE2009-80216, Honolulu, 2009.

[16] A. Koop and A. Bereznitski. *Model-scale and full-scale CFD calculations for current loads on semi-submersible*. In Proceedings of the 30^{th} International Conference on Ocean, Offshore and Arctic Engineering, OMAE2011-49204, Rotterdam, 2011.

[17] E. Croonenborghs, T. Sauder, S. Fouques, and S-A. Reinholdtsen. *CFD Prediction of Wind and Current Loads on a Complex Semi-Submersible Geometry*. In Proceedings of the 12^{th} International Conference on Computer and IT Applications in the Maritime Industries, COMPIT'13, Cortona, 2013.

[18] E. Croonenborghs, T. Sauder, S. Fouques, and S-A. Reinholdtsen. *Comparison of Various of Methods for the Assessment of Wind and Current Loads on a Semi-Submersible Platform*. OTC Brasil, OTC 24399, Rio de Janeiro, 2013.

[19] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics - The Finite Volume Method*. Pearson Education Limited, 2007.

[20] H. Jasak. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. PhD thesis. Imperial College of Science, Technology and Medicine, 1996.

[21] O. Reynolds. *An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels*. None, 1883.

[22] S. Pope. *Turbulent Flows*. Cambridge University Press, 2000.

[23] P. Schlatter. *Large-eddy simulation of transition and turbulence in wall-bounded shear flow*. Diss., Technische Wissenschaften ETH, Zürich, 2005.

[24] L. Davidson. *An Introduction to Turbulence Models*. Chalmers University of Technology, Göteborg, 2003.

[25] J. Boussinesq. *Théorie de l'écoulement tourbillant. Mem. Présentés par Divers Savants Acad. Sci. Inst. Fr.*, 23:46–50, 1877.

[26] A.N. Kolmogorov. *Equations of turbulent motion of an incompressible fluid*. Physics, 6:56–58, 1942.

[27] W.P. Jones and B.E. Launder. *The prediction of laminarization with a two-equation model of turbulence*. International Journal of Heat and Mass Transfer, 15:301–314, 1972.

[28] P.L. Davis, A. T. Rinehimer, and M. Uddin. *A Comparison of RANS-Based Turbulence Modeling for Flow over a Wall-Mounted Square Cylinder*. Technical paper, CD-adapco. Available from: http://www.cd-adapco.com/sites/default/files/technical_document/pdf/PRU_2012.pdf

[29] H. Tennekes and J.L. Lumley. *A First Course in Turbulence*. Massachusetts Institute of Technology, Cambridge, 1972.

[30] B. Nebenführ. *OpenFOAM: A tool for predicting automotive relevant flow fields*. Master's thesis. Chalmers University of Technology, Göteborg, 2010.

[31] G. Kalitzin, G. Medic, G. Iaccarino, and P. Durbin. *Near-wall behavior of RANS turbulence models and implications for wall functions*. Journal of Computational Physics 204, 2004.

[32] B.E. Launder and D.B. Spalding. *The numerical computation of turbulent flows*. Comp. Methods Appl. Mech. Eng., 3:269-289, 1974.

[33] T. von Kármán *Mechanische Ähnlichkeit und Turbulenz*. Nachr. Ges. Wiss., Göttingen, pp 68, 1931

[34] OpenFOAM. *User guide, version 2.3.1*. 2014.

.