NTNU

Norwegian University of
Science and Technology

Resistance analysis of trimaran service vessel using CFD

Anita Solhaug Hatlevik

Marine Technology
Submission date: July 2018
Supervisor: Håvard Holm, IMT

Norwegian University of Science and Technology
Department of Marine Technology

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER THESIS IN MARINE TECHNOLOGY

SPRING 2018

FOR

STUD. TECHN. Anita Solhaug Hatlevik

Resistance analysis of a trimaran service vessel using CFD

In the thesis the candidate shall present her personal contribution to the resolution of problem within the scope of the thesis work.

A numerical simulation of a trimaran ship hull in the free surface shall be performed using OpenFOAM. The candidate shall familiarize herself with the theory behind Computational Fluid Dynamics and the OpenFOAM software.

The candidate shall pay special attention to the resistance force, and make an estimate of the ship resistance. If time permits, the candidate should also optimize the hull geometry with respect to the submergence depth of the center hull.

As a test case for the code, the student shall simulate a cylinder in axial flow. The simulations shall be started out with the cylinder completely submerged in water, before the cylinder is moved closer to the free surface.

The candidate should utilize the existing possibilities for obtaining relevant literature.

Theories and conclusions should be based on mathematical derivations and/or logic reasoning identifying the various steps in the deduction.

The thesis should be organized in a rational manner to give a clear exposition of results, assessments, and conclusions. The text should be brief and to the point, with a clear language. Telegraphic language should be avoided.

The thesis shall contain the following elements: A text defining the scope, preface, list of contents, summary, main body of thesis, conclusions with recommendations for further work, acronyms, reference and (optional) appendices. All figures, tables and equations shall be numerated.

The supervisor may require that the candidate, in an early stage of the work, present a written plan for the completion of the work. The plan should include a budget for the use of computer and laboratory resources that will be charged to the department. Overruns shall be reported to the supervisor.

The original contribution of the candidate and material taken from other sources shall be clearly defined. Work from other sources shall be properly referenced using an acknowledged referencing system.

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

The thesis shall be submitted electronically (pdf) in DAIM.

Supervisor : Associate Professor Håvard Holm
Co-supervisor : Senior Engineer Tufan Arslan
Deadline : 20.07.2018

Summary

Service vessels within the aquaculture industry are normally designed as catamarans, however, the ships experience large resistance, where the contribution from wave resistance is dominant. In order to increase the ship speed and lower the resistnace, it is looked at the possibility of changing the ship hull from a catamaran to a trimaran.

An overview of ship resistance theory is given along with a solid Computational Fluid Dynamics background. A method based on RANS equations and $k - \omega$ SST turbulence modelling has been developed and the code has been tested on a cylinder in axial flow with different submergence depths. The resulting mean value of the drag coefficients agrees well with literature.

Resistance analysis for the trimaran hull with incoming uniform fluid velocity equal to 10 knots has been done. OpenFOAM v1712 is used to perform the simulations. The mesh is generated by using **blockMesh** and **snappyHexMesh**. The simulations of the trimaran hull were run on Vilje using 32 processors. The cylinder test case were run on the CFD2 computer at the Department of Marine Technology at Tyholt.

Resistance data for a catamaran vessel with similar volume displacement and approximately the same L_{OA} is used for comparison. The resistance force computed for the catamaran is equal to 32 kN. The resistance force obtained from the simulations is approximately equal

to 26 kN. The fluid flow in the trimaran simulation did not stabilise due to limited time to run the simulation. Thus, the resistance data are concluded to be not sufficiently reliable to proceed in the design process. However, the setup can be used as a basis for further work which can be a valuable contribution to the design process along with model testing.

Sammendrag

Servicefartøy i oppdrettsnæringen er vanligvis utformet som katamaraner, men fartøyene opplever stor motstand hvor bølgemotstanden dominerer. For å nå høyere hastigheter og redusere motstanden, sees det på muligheten til å forandre skrogformen fra en katamaran til en trimaran.

En oversikt over skipsmotstandsteori er gitt, sammen med en solid bakgrunnsteori i Computational Fluid Dynamics. En metode basert på RANS-ligninger og $k - \varepsilon$ SST turbulensmodellering er utviklet og testet på en sylinder i forskjellig neddykkingsdybde i aksial strømning. Den resulterende gjennomsnittsverdien til drag koeffisienten er i god overenstemmelse med litteratur om emnet.

Motstandsanalyse av trimaran-skroget med en innkommende uniform væskehastighet lik 10 knop er gjennomført. OpenFOAM v1712 er brukt til å gjennomføre simuleringene. Meshen er generert ved å bruke **blockMesh** og **snappyHexMesh**. Simuleringene av trimaran-skroget er kjørt på Vilje fordelt på 32 prosessorer. Sylindersimuleringene er kjørt på CFD2-datamaskinen ved Fakultet for Marin Teknikk på Tyholt.

Motstandsdata for en katamaran med samme volumdeplasement og omtrentlig lik L_{OA} er brukt for sammenligning. Motstanden beregnet i tilfellet med trimaranen er lik 32 kN. Motstanden simulert med kata-

maranen er omtrent lik 26 kN. Fluidstrømningen i trimaran-simuleringen ble ikke stabil på grunn av begrenset tid til å kjøre simuleringen. På grunn av dette, konkluderes det med at motstandsdataene ikke er tilstrekkelig pålitelige til at designprosessen kan fortsette. Derimot kan simuleringsoppsettet brukes som en base for videre arbeid, som kan bli et verdifullt bidrag til designprosessen, sammen med modelltesting.

Preface

This thesis is written during the last semester of a five year-long Master of Science program in Marine Technology at the Department of Marine Technology at the Norwegian University of Science and Technology, where I have specialised in hydrodynamics.

The thesis is based on a preliminary study conducted by myself during the Autumn of 2017, and work on the thesis was started in January 2018.

I would like to thank my supervisor, Associate Professor at the Department of Marine Technology, Håvard Holm, for his guidance. I would also like to thank Managing Director of Heimli Ship Design, Asbjørn Vik, for providing me with problem suggestions, a CAD model of the ship and for giving me the opportunity to work on a real project.

Lastly, I wish to express my sincere gratitude to my co-supervisor, Senior Engineer at the IT Development Section, Tufan Arslan, for interesting discussions and invaluable guidance.

Anita Solhaug Hatlevik
Trondheim, July 20, 2018

Contents

List of Tables

List of Figures

Abbreviations

ROV	Remotely Operated Underwater Vehicle
CFD	Computational Fluid Dynamics
CAD	Computer-Aided Design
CV	Control Volume
FVM	Finite Volume Method
RANS	Reynolds-Averaged Navier-Stokes
LES	Large Eddy Simulation
DNS	Direct Numerical Simulation
SST	Shear Stress Transport
VOF	Volume of Fluid
CFL	Courant-Friedrichs-Lewy
STL	Stereolithography
GAMG	Geometric-Algebraic Multi-Grid

Chapter 1

Introduction

1.1 Background

Service vessels within the aquaculture industry are, due to physical environments and regulatory requirements, typically 15 meters long. Common tasks for these boats are lifting operations, cleaning and inspection of fish nets, treatment of salmon lice and diving operations. Thus, they are usually well equipped with for instance one or more cranes, ROV and pump systems. The vessels are normally catamarans in order to provide adequate deck area and to ensure good manoeuvring characteristics. Because of the combination of heavy lightship weight and rather short length, the boats experience large resistance, where the contribution from wave resistance is dominant. As a consequence the forward speed is limited to approximately 11-12 knots. However, ship owners have requested higher velocities while decreasing fuel consumption and obtaining a better environmental profile.

In order to satisfy the new criteria, it is looked to the possibility of changing the ship hull from a catamaran to a trimaran. A new design is proposed where the center hull is submerged under the free surface, mainly accounting for buoyancy, while slender side hulls ensure some

buoyancy but primarily contribute to the stability of the vessel. In such an early stage of the design process, it is useful to make an estimate of the resistance in order to do an evaluation of the design. For this purpose, it is wished to perform an analysis of the ship by using Computational Fluid Dynamics as a part of a Master Thesis.

Over the past few decades, CFD has been extensively used to evaluate the hydrodynamic performance in ship design. Some advantages of Computational Fluid Dynamics is that it can solve for full scale problems and the geometry is easy to modify if changes need to be made. Consequently, CFD is cheaper and less time consuming compared to for instance experimental testing.

Computational Fluid Dynamics is used as a tool in several fields such as aerospace, power generation, automotive, sports industry chemical manufacturing, medical research etc. Although there are many advantages of applying CFD on engineering problems, there are also challenges related to the issue. As described later in this report, grid generation is an important part of CFD. How the task is approached is crucial to whether the results are correct or practically useless. Thus, adequate knowledge about the topic is highly critical to make appropriate use of CFD tools.

1.2 Objectives

The purpose of this master thesis is to accquire more knowledge and a better understanding of CFD in ship design. More specifically, the objectives of this thesis work are:

- Perform a resistance analysis of the new design using CFD in straight-ahead cruising condition with a forward speed equal to 10 knots.

- Set up a test case of a cylinder in axial flow to validate the integrity of the simulation setup.

- Evaluate the results of the trimaran by comparing the CFD results for an existing design of a catamaran with the same length and displacement. Resistance data in calm water with a forward speed equal to 10 knots and a ship model (hull data) is provided.

- Become more familiar with CFD and the software OpenFOAM.

1.3 Limitations

The numerical simulation of flow around the trimaran hull in this thesis work is limited to:

- Incompressible fluid (constant density)
- Uniform incoming velocity profile with magnitude equal to 10 knots
- Use of symmetry condition
- Total number of cells limited to $8 \sim 9$ million cells in order to prevent the simulations from taking an extensive amount of time.

1.4 Approach

Complex geometries, such as ship hulls, are harder to mesh than simpler objects. In order to get familiar with three-dimensional simulations in OpenFOAM and the meshing utility used in the simulations, it is chosen to set up a simulation for a cylinder in axial flow. Since CFD simulations of bodies in the free surface are complicated, the simulations of the cylinder is started out with the geometry submerged below the surface. Further, it is moved closer to the free surface. The resulting drag coefficient is computed and compared to experimental results. Further, the work is proceeded by replacing the cylinder with the trimaran hull. First, a simulation is done for the side hull separately before the trimaran hull is computed.

1.5 Structure of report

Ch. 2 gives an introduction to ship resistance, the topic which is one of the main concerns in this thesis work. In Ch. 3, the theory behind Computational Fluid Dynamics which is used in this project is presented, followed by a description of the method in Ch. 4. Ch. 5 presents the setup for the cylinder test case. Further, the results from the simulations are shown in Ch. 6 and a discussion of the results and the work is presented in Ch. 7. Lastly, a conclusion of this project is given in Ch. 8.

Chapter 2

Ship resistance

The resistance of a ship is the component of the total net force acting in the opposite direction of the motion of the vessel. The forces can be divided into pressure forces, which act normal to the hull surface, and shear forces acting tangential to the surface. The shear forces are caused by the friction between the body and the fluid, and the resistance due to these forces is called viscous resistance. For a ship hull, the pressure forces are mainly due to wave making [Steen and Minsaas, 2013]. Therefore, in ship resistance theory it is common practice to divide the total resistance into viscous resistance and wave making resistance.

2.1 Viscous resistance

The viscous resistance can be further subdivided into four components: flat plate friction (frictional resistance), roughness effect, form effect on friction and form effect on pressure. In viscous fluid flow, a boundary layer is created along the surface of a body. All the viscous resistance components are related to the boundary layer that develops around a ship hull.

2.2 Wave making resistance

In inviscid fluid flow there are no friction forces. According to d'Alemberts paradox, the total force acting on a body deeply immersed in inviscid fluid flow is equal to zero. Thus, the pressure resistance of a body in inviscid fluid is said to be due to wave making. For a streamlined body such as a ship in inviscid fluid, the shape of the body causes a varying velocity and pressure distribution along the surface according to Bernoulli's equation as can be seen in Figure 5.3 in [Larsson and Raven, 2010].

In the free surface, in order to compensate for the change in pressure the water surface is deformed. As a function of the hull shape, waves are generated near the bow, stern and shoulders of the ship hull [Steen and Minsaas, 2013]. The waves consist of two types: the inner wave system, often referred to as the Bernoulli wave system, and the free waves known as the Kelvin free wave pattern. The inner wave system arises from local disturbances caused by the pressure-velocity distribution around the hull. They are non-oscillatory and

decrease rapidly as they move away from the vessel. The Bernoulli wave system does not cause resistance on the ship directly, but indirectly through the way the Bernoulli waves interfere with the free waves [Steen and Minsaas, 2013].

The Kelvin free wave pattern consists of oscillatory waves moving steadily with the same speed as the ship. Figure 2.1 shows the Kelvin wave pattern from a single pressure point travelling in a straight line over the surface of the water. The section that contains the wave pattern is called the *Kelvin wedge* [Larsson and Raven, 2010]. The pattern consists of transverse waves and diverging waves, the latter usually being the most visible as they are short and often steep. However, especially for conventional ships, it is the effect from transverse waves that is the most important contribution to the wave resistance [Steen and Minsaas, 2013].

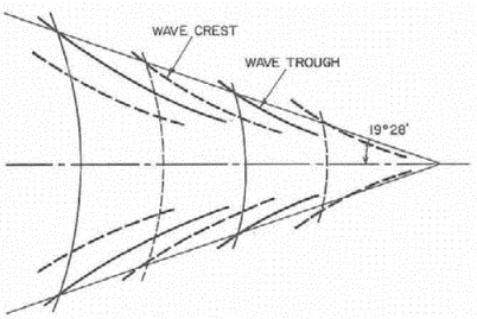


Figure 2.1: The Kelvin free wave pattern from a single pressure point travelling in a straight line over the surface of the water. Figure from *Ship resistance and flow* [Larsson and Raven, 2010].

The angle between the common wave front formed by the diverging and the transverse waves and direction the ship is travelling is equal to 19°28′. It is often referred to as the Kelvin angle. The resistance of the two wave systems for a typical ship is shown in Figure 2.2.

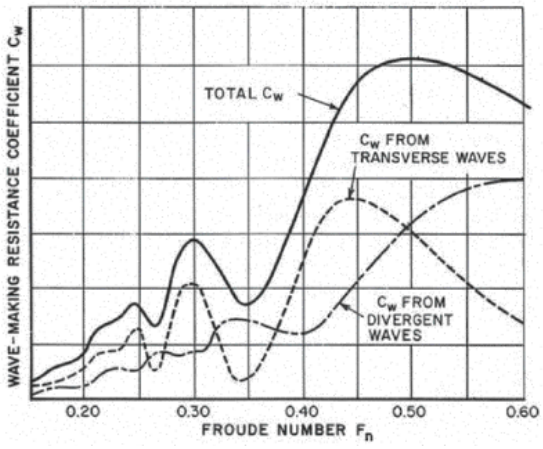


Figure 2.2: Contributions of transverse and divergent waves to the wave making resistance. Figure from *Ship Resistance and Flow* [Larsson and Raven, 2010].

Unlike the wave pattern of a pressure point (Figure 2.1), the waves generated by a ship's bow, stern and shoulders will generate their own wave systems, contained in separate Kelvin wedges that overlap and interfere. The interference is uniquely defined by the Froude Number, F_N, and can be either constructive, meaning the wave systems amplify each other and lead to high wave resistance, or destructive in which waves cancel each other corresponding to lower wave resistance.

2.3 Resistance of Multi Hulls

To operate in the range $0.4 \leq F_N \leq 0.9$, monohulls need to be very slender which typically affect the transverse stability negatively. Therefore, multi hulls are required to maintain a reasonable wave resistance. For a multihull, the waves generated from the hulls will interfere with each other. This interference can be either positive or negative and it is therefore important to study this interference. Ideally, the wave peak from one of the hulls will fill out a wave through from the other hull(s), causing no final transverse wave. This can, for instance, be achieved by placing the hulls in asymmetric relation to each other, namely staggered hulls (see [Steen and Minsaas, 2013]).

Chapter 3

Computational Fluid Dynamics background

In ship design, computational fluid dynamics has become increasingly important especially in the evaluation of ship resistance and propulsion. Computer-aided design (CAD) to generate new hull shapes together with CFD to analyse them allows for rapid design exploration without model testing. In this chapter, fundamental theory of computational fluid dynamics is presented along with an overview of the OpenFOAM software and some of its utilities, namely the ones that are used in this master thesis.

3.1 Governing equations

Computational Fluid Dynamics is based on three fundamental governing equations: the conservation of mass (the continuity equation), the transport of linear momentum and the energy equation. However, in hydrodynamics the temperature is often considered to be constant, thus, it is rarely necessary to solve the energy equation. The equations are based on the conservation principles from assuming a given quantity of matter, which in fluid flows is considered as a spatial re-

gion referred to as a *control volume* (CV). A general reference for this chapter is made to [Ferziger and Perić, 1997] and [Blazek, 2005].

3.1.1 The conservation of mass (Continuity Equation)

For single-phase fluids, the conservation of mass expresses that mass cannot be created nor disappear within a fluid system. Considering a fluid flow through a control volume (CV), the rate of increase of mass in the volume V is equal to the net outflow of the mass trough the surface S, that is

$$\frac{\partial}{\partial t}\int_{\Omega}\rho\, d\Omega + \oint_{S}\rho(\vec{v}\cdot\vec{n})\, dS = 0 \tag{3.1}$$

By applying the Gauss divergence theorem, the continuity equation in Cartesian coordinates becomes

$$\frac{\partial\rho}{\partial t} + \frac{\partial(\rho u_x)}{\partial x} + \frac{\partial(\rho u_y)}{\partial y} + \frac{\partial(\rho u_z)}{\partial z} = 0 \tag{3.2}$$

or, using the Einstein Convention

$$\frac{\partial\rho}{\partial t} + \frac{\partial(\rho u_i)}{\partial x_i} = 0 \tag{3.3}$$

where x_i for $i = 1, 2, 3$ indicates x,y and z in Cartesian coordinates and u_i, or (u_x,u_y,u_z) are the Cartesian components of the velocity vector **v**. By assuming the density of the fluid is constant, the first term on

the left disappears, i.e.

$$\frac{\partial(\rho u_i)}{\partial x_i} = 0 \tag{3.4}$$

3.1.2 The conservation of momentum (the Navier-Stokes Equation)

The conservation of momentum is derived from Newton's second law which states that the variation of momentum is caused by the net force acting on a mass element. For a volume Ω, Newton's second law can be expressed in vector form as

$$\int_\Omega \rho \frac{D\mathbf{V}}{Dt} d\Omega = \oint_S \vec{s}\, dS + \int_\Omega \rho \vec{f} d\Omega \tag{3.5}$$

where $\vec{s}$ are surface forces and $\vec{f}$ are body forces. The x_i-component of the surface force can be written as

$$\oint_S s_i dS = \oint_S \sigma_{ij} n_j dS = \int_\Omega \frac{\partial \sigma_{ij}}{\partial x_j} dV \tag{3.6}$$

Inserted into Eq. (3.5) leads to

$$\rho \frac{Du_i}{Dt} = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho f_i \tag{3.7}$$

Eq. (3.7) is the basic conservation law for momentum in fluid dynam-

ics. For a Newtonian fluid, the viscous stress σ'_{ij} is given by

$$\sigma'_{ij} = \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}) \tag{3.8}$$

where μ is the viscosity of the fluid and δ_{ij} is the Kronecker delta function, which is equal to one when $i = j$ and zero for $i \neq j$. Substitution of Eq. (3.8) into (3.7), results in the *Navier-Stokes equation*

$$\rho\frac{Du_i}{Dt} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}[\mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k})] + \rho f_i \tag{3.9}$$

The body force f is a force that acts directly on the mass of the volume for instance gravity, buoyancy, Coriolis, or centrifugal forces.

3.2 The Finite Volume Method

The Finite Volume Method (FVM) is a discretisation technique used to transform the governing equations on partial differential form to a set of linear algebraic equations which can be solved by a computer. The method is based on dividing the computational domain into a number of grid cells, each cell representing a control volume. OpenFOAM uses a cell-centered finite volume method to solve the flow problem, meaning that information is stored at the cell centroid as shown in Figure 3.1.

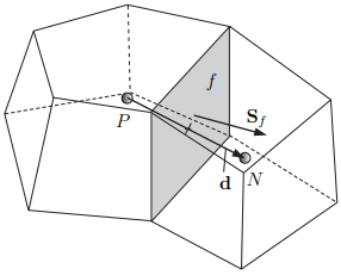


Figure 3.1: Illustration of parameters in finite volume discretisation. Figure taken from *OpenFOAM Programmer's guide*, [Greenshields, 2015]

Here, P and N are the cell centroids of two neighbouring cells with a distance $\mathbf{d}$ between the centroids, f represents the face surface between the cells with normal vector $\mathbf{S_f}$. According to [Moukalled et al., 2016], the steady-state form of the conservation equation for a scalar variable ϕ is given by

$$\nabla(\rho \mathbf{U}\phi) = \nabla(\Gamma^{\phi}\nabla\phi) + Q^{\phi} \tag{3.10}$$

where the expression on the left-hand side is the convective term, the first expression on the right-hand side is the diffusive term, where Γ^{ϕ} is the diffusion coefficient of ϕ, and Q^{ϕ} is the source term. The Finite Volume Method is based on balancing the flux across the cell boundaries. Consider the element with cell centre P. By integrating Eq. (3.10) over the cell volume of P, the conservation equation is transformed to

$$\int_{V_C} \nabla(\rho \mathbf{U}\phi)\, dV = \int_{V_C} \nabla(\Gamma^{\phi}\nabla\phi)\, dV + \int_{V_C} Q^{\phi}\, dV \tag{3.11}$$

Further, the divergence theorem is applied in order to transform the volume integrals of the convective and diffusive terms into surface integrals. Thus, Eq. (3.11) changes to

$$\oint_{\partial V_C} \nabla(\rho \mathbf{U}\phi)\, dS_f = \oint_{\partial V_C} \nabla(\Gamma^\phi \nabla\phi)\, dS_f + \int_{V_C} Q^\phi\, dV \tag{3.12}$$

where $\oint_{\partial V_C}$ is the surface integral over the volume V_C. For element P, the surface integrals in Eq. (3.12) can be replaced by a summation of the flux terms over the faces of cell P. Further, the integrals are transformed into discrete terms, and integrated numerically.

3.3 Turbulence modelling

There are several methods for predicting turbulent flows. The most common methods are the Reynolds-Averaged Navier-Stokes Equation (RANS) method, Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS), where the latter is considered the most exact approach to turbulence simulation. By using DNS, all of the motions contained in the fluid are resolved. The computational domain must be at least as large as the largest turbulent eddy, and the grid size must be small enough to capture all of the kinetic energy dissipation. This results in a relatively large number of grid points which, due to limited processing speed and memory of computers, makes direct numerical simulations applicable at only low Reynolds numbers [Ferziger and Perić, 1997]. DNS solves for all turbulence, hence the information obtained is very detailed and for engineering related cases it provides far more information than what is usually needed. Because

of this, DNS is more or less used for research purposes only.

In cases where the Reynolds number is too high, or the geometry is too complex for application of DNS, an alternative method is the Large Eddy Simulation (LES) or the RANS equations. Large Eddy Simulation only captures the eddies which are larger than one cell size. The simulation is based on the theory that large scale motions are generally much more energetic than small ones, thus they are the most effective transporters of conserved energy. For engineering purposes with steady-state behaviour, the RANS method is a common approach.

3.3.1 The Reynolds-Averaged Navier-Stokes (RANS) Equation

The first approach to treat turbulent flow appropriately was the Reynolds Averaging method presented by Reynolds in 1895 [Blazek, 2005]. The principle behind this method is that the flow variables in turbulent flows fluctuate around a mean value, thus the variable can be divided into a mean and a fluctuating part. There are three different methods of Reynolds Averaging. They are time averaging, spatial averaging and ensemble averaging. For a stationary homogeneous turbulent flow the three methods are equivalent and the process is said to be ergodic.

In chapter two in [Tennekes and Lumley, 1972], the derivation of the Reynolds-Averaged Navier-Stokes Equation begins with decomposition of flow variables. Considering the velocity component, $\tilde{u}_i$, and pressure

component, $\tilde{p}_i$, the decomposed variables can be expressed as

$$\tilde{u_i} = U_i + u_i \qquad \text{and} \qquad \tilde{p_i} = P_i + p_i \tag{3.13}$$

where U_i and P_i are the mean velocity and pressure, and u_i and p_i are the corresponding fluctuating parts as illustrated in Figure 3.2.

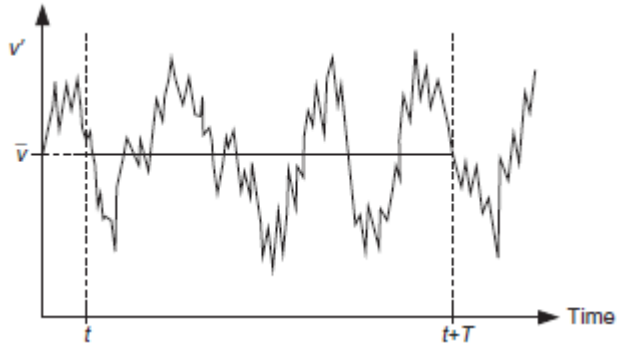


Figure 3.2: Illustration of turbulent velocity fluctuations v' and statistical mean value $\bar{v}$. *Figure from Computational Fluid Dynamics: Principles and Applications [Blazek, 2005]*

The same principle of decomposition is valid for the stress and strain tensors. By substituting the decomposed variables of the stress and strain tensors, along with the variables in Eq. (3.13), into the incompressible mass conservation and the Navier-Stokes equation, the following expressions are obtained

$$\frac{\partial U_i}{\partial x_i} + \frac{\partial u_i}{\partial x_i} = 0 \tag{3.14}$$

$$\frac{\partial(U_i + u_i)}{\partial t} + (U_j + u_j)\frac{\partial(U_i + u_i)}{\partial x_j} = \frac{1}{\rho}\frac{\partial(\Sigma_{ij} + \sigma_{ij})}{\partial x_j} \tag{3.15}$$

where $\Sigma_{ij} = -P\delta_{ij} + 2\mu S_{ij}$ is the mean value of the stress tensor, while σ_{ij} is the fluctuating part of the stress. Now, it applies that the mean value of a spatial derivative of a variable is equal to the corresponding spatial derivative of the mean value of the same variable. Thus, taking the time average of the mass conservation, $\frac{\partial \bar{u_i}}{\partial x_i}$, corresponds to the spatial derivative of the mean velocity. The time average of the fluctuating part is equal to zero and the last term vanishes. Hence, the mean flow will also be incompressible:

$$\frac{\partial U_i}{\partial x_i} = 0 \tag{3.16}$$

Subtracting this term from Equation (3.14), it follows that the turbulent velocity fluctuations are also incompressible:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{3.17}$$

Further, applying the time averaging to the Navier-Stokes expression in Equation (3.15), the expression reduces to:

$$\frac{\partial U_i}{\partial t} + U_j\frac{\partial U_i}{\partial x_j} + \overline{u_j\frac{\partial u_i}{\partial x_j}} = \frac{1}{\rho}\frac{\partial}{\partial x_j}(\Sigma_{ij}) \tag{3.18}$$

The last term on the left hand side can be written as $\overline{u_j\frac{\partial u_i}{\partial x_j}} = \frac{\partial}{\partial x_j}\overline{u_i u_j} - \overline{u_i\frac{\partial u_j}{\partial x_j}}$, where the latter is equal to zero due to continuity. The expres-

sion obtained, together with Equation 3.17, make up the unsteady Reynolds-Averaged Navier-Stokes Equations:

$$\frac{\partial U_i}{\partial x_i} = 0 \tag{3.19}$$

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = \frac{1}{\rho} \frac{\partial}{\partial x_j} (\Sigma_{ij} - \rho \overline{u_i u_j}) \tag{3.20}$$

where the last term, $-\rho \overline{u_i u_j}$, is known as the Reynolds stress tensor. From comparison with the Navier-Stokes equation for non-turbulent flows, the difference between the two expressions is the Reynolds stress tensor, which indicates that the stresses in the fluid are increased for turbulent flows. The RANS equations are not closed, i.e. they contain more variables than equations. Thus, it is necessary to apply a turbulence model in order to approximate the unknowns.

Several attempts have been made to describe the turbulent stresses mathematically. Based on the number of equations the turbulence models add to the system, they can be divided into different categories. Some common models are:

- Zero Equation Model: The Mixing Length model
- One Equation Model: Spalart-Allmaras
- Two Equation Models: The k-ε model, the k-ω model and Shear Stress Transport (SST) model
- Seven Equation Model: Reynolds Stress Model

One of the earliest attempts and important contributions to turbulence modelling, were made by Boussinesq in 1877 [Celik, 1999], namely the Boussinesq hypothesis. Using the notation from [Tennekes and Lumley, 1972], the Boussinesq hypothesis for the Reynolds stress term can be expressed as:

$$-\rho\overline{u_i u_j} = 2\mu_T S_{ij} - \frac{2}{3}\rho k \delta_{ij} \tag{3.21}$$

where μ_T is the eddy viscosity, S_{ij} is the strain rate tensor ($S_{ij} = \frac{1}{2}(\frac{\partial U_i}{\partial x_i} + \frac{\partial U_j}{\partial x_j})$) and k is the mean turbulent kinetic energy ($k = (1/2)\overline{u_i u_i}$).

Another significant contribution to turbulence modelling is the Prandtl number σ:

$$\sigma = \frac{\gamma}{\nu} \tag{3.22}$$

where γ is the thermal diffusivity. The Boussinesq hypothesis (Eq. 3.21) shows that the turbulent transport of momentum is assumed to be proportional to mean gradients of velocity. By analogy, turbulent transport of a scalar φ is taken to be proportional to the gradient of the mean value of the transported quantity [Versteeg and Malalasekera, 1995], that is

$$-\rho\overline{u_i \varphi_j} = \gamma \frac{\partial \Phi}{\partial x_i} \tag{3.23}$$

By substituting the Reynold stress tensor using the Boussinesq hypothesis along with $\Sigma_{ij} = -P\delta_{ij} + 2\mu S_{ij}$ into the RANS equation (Eq. 3.20), the equation becomes

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = \frac{1}{\rho}\frac{\partial}{\partial x_j}(-P\delta_{ij} + 2\mu S_{ij} + 2\mu_T S_{ij} - \frac{2}{3}\rho k \delta ij) \quad (3.24)$$

which can also be written as

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{\partial}{\partial x_i}(\frac{P}{\rho} + \frac{2}{3}k) + \frac{1}{\rho}\frac{\partial}{\partial x_j}(2(\mu + \mu_T)S_{ij}) \quad (3.25)$$

However, the mean turbulent kinetic energy k and the eddy viscosity μ_T still remain unknown. An approach to model the eddy viscosity is the Mixing Length Model. The Mixing-length model assumes that the properties of a fluid parcel is conserved for a characteristic length, the *mixing length*, before mixing with the surrounding fluid. In this way, the turbulent viscosity can be described in terms of a velocity scale ϑ and a turbulent length scale l, i.e. $\nu_T = C\vartheta l$ [Versteeg and Malalasekera, 1995].

The mixing length model is very useful in flows where turbulence properties develop in proportion to a mean flow length scale. The model is easy to implement and cheap with respect to computer resources, and it presents a good prediction for thin shear layers. However, it is completely incapable of describing flows with separation and re-circulation, thus it is not commonly used on its own in CFD [Versteeg and Malalasekera, 1995]. But, the model is embedded in other more sophisticated models, such as the k-ϵ model described in the following chapter.

3.3.2 The $k - \varepsilon$ model

The $k - \varepsilon$ model, as presented in [Versteeg and Malalasekera, 1995], allows for effects from transport of turbulent properties of the mean flow and diffusion from production and destruction of turbulence. It adds two model equations to the system, one for the turbulent kinetic energy k and one for the rate of viscous dissipation ε.

By multiplying the Navier-Stokes equation (Eq. 3.15) with the velocity $\tilde{u_i}$, taking the time average of all terms and subtracting the energy equation for the mean flow, a transport equation for the mean turbulent kinetic energy k can be expressed as

$$\frac{Dk}{Dt} = -\frac{\partial}{\partial x_j}(\frac{1}{\rho}\overline{u_i p} + \frac{1}{2}\overline{u_i u_i u_j} - 2\nu\overline{u_i s_{ij}}) - \overline{u_i u_j}S_{ij} - 2\nu\overline{s_{ij}s_{ij}} \quad (3.26)$$

where the last term $2\nu\overline{s_{ij}s_{ij}}$ is the viscous dissipation ε. The turbulent production P_k is given by

$$P_k = -\overline{u_i u_j}S_{ij} \quad (3.27)$$

Here, the term $-\overline{u_i u_j}$ can be replaced by using the Boussinesq hypothesis (Eq. 3.21). The second term in Eq. (3.21) multiplied with S_{ij} vanishes due to the Kronecker delta function. Thus, the turbulent production can be expressed as

$$P_k = 2\nu_T S_{ij}S_{ij} \quad (3.28)$$

The terms in Eq. (3.26) related to transport due to diffusion are

$$D_k = -\frac{\partial}{\partial x_j}(\frac{1}{\rho}\overline{u_i p} + \frac{1}{2}\overline{u_i u_i u_j} - 2\nu\overline{u_i s_{ij}}) \tag{3.29}$$

At high Reynolds numbers, the last term is very small and can be neglected. The second term can be rewritten using Eq. (3.23) and the Prandtl number (Eq. 3.22). Neglecting the pressure term, the resulting expression for the mean turbulent kinetic energy transport is

$$\frac{Dk}{Dt} = \frac{\partial}{\partial x_i}(\frac{\nu_T}{\sigma_k}\frac{\partial k}{\partial x_i}) + 2\nu_T S_{ij}\frac{\partial U_i}{\partial x_j} - 2\nu\overline{s_{ij}s_{ij}} \tag{3.30}$$

The k-ε model uses the same approach as in the Mixing Length Model by defining a velocity scale and a length scale in terms of ε and k expressed as

$$\vartheta = k^{1/2} \qquad\qquad \ell = \frac{k^{3/2}}{\varepsilon} \tag{3.31}$$

Further, the expressions for the velocity scale and length scale are inserted into the eddy viscosity term

$$\nu_T = C_\mu \frac{k^2}{\varepsilon} \tag{3.32}$$

In accordance with [Tennekes and Lumley, 1972], the governing equation for the viscous dissipation is introduced as follows:

$$\frac{\partial \varepsilon}{\partial t} = \frac{\partial}{\partial x_i}(\frac{\mu_T}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_i} + C_{1\varepsilon}\frac{\varepsilon}{k}2\mu_T S_{ij}\frac{\partial U_i}{\partial x_j} - C_{2\varepsilon}\frac{\varepsilon^2}{k} \tag{3.33}$$

Including the transport equation for the kinetic energy, the system of equations now contains five constants σ_ε, σ_k, $C_{1\varepsilon}$, $C_{2\varepsilon}$ and C_μ. Standard values of these constants are arrived at from a wide range of experiments with turbulent flows. Compared to the Mixing Length method, the $k-\varepsilon$ model is more costly, but more sophisticated and general way to describe turbulent flows [Versteeg and Malalasekera, 1995].

3.3.3 The $k-\omega$ model

Similar to the $k-\varepsilon$ model, the $k-\omega$ model [Wilcox, 1988] adds two equations to the system, but instead of the viscous dissipation rate ε, it uses the vorticity rate ω. The quantity ω is also referred to as the specific dissipation rate. Further, the eddy viscosity is expressed in terms of k and ω. Like the $k-\varepsilon$ model, it considers the viscosity to be isotropic, thus it fails to predict anisotropy of the normal stresses and to account for streamline curvature effects. However, the model is more accurate regarding adverse pressure gradients, which makes it better for solving flows where wall effects are present.

3.3.4 The k-ω Shear Stress Transport turbulence model

The k-ω Shear Stress Transport (SST) [Menter, 1993] is a two-equation turbulence model that adds two new expressions to the RANS equations, one equation for the kinetic energy and another expression for the turbulence specific dissipation rate ω. It switches adaptively between the k-ϵ in the far field fluid flow and the k-ω inside the turbulent

boundary layer. The governing equations for the k-ω SST model are:

$$\frac{Dk}{Dt} = \frac{1}{\rho}\tau_{ij}\frac{\partial u_i}{\partial x_j} - \beta^*\omega k + \frac{\partial}{\partial x_j}[(\nu + \sigma_k n u_T)\frac{\partial k}{\partial x_j}] \tag{3.34}$$

$$\frac{D\omega}{Dt} = \frac{\gamma}{\rho\nu_T}\tau_{ij}\frac{\partial u_i}{\partial x_j} - \beta\omega^2 + \frac{\partial}{\partial x_j}[(\nu + \sigma_\omega n u_T)\frac{\partial \omega}{\partial x_j}] + 2(1 - F_1)\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \tag{3.35}$$

3.4 Turbulent boundary layer approximation

According to [White, 2006], Ludwig Prandtl and Theodore von Kármán deduced that the velocity profile in turbulent flows over a solid surface consist of three layers

- Inner layer: dominated by viscous shear (molecular)
- Outer (intermediate) layer: dominated by turbulent (eddy) shear
- Overlap layer: both types of shear is important

Different laws can be applied to the different layers. For the velocity profile in the inner layer, the inner law yields

$$\bar{u} = fcn(\tau_w, \rho, \mu, y) \tag{3.36}$$

For the outer layer, the flow is assumed to be retarded according to

$$U_e - \bar{u} = f(\tau_w, \rho, y, \delta, \frac{dp_e}{dx}) \tag{3.37}$$

where U_e is the free stream velocity, δ is the layer thickness and p_e is the free stream pressure. Regarding the overlap layer between the outer and inner region, the velocities are assumed to be equal, ensuring a smooth transition between the regions. By introducing the dimensionless variables

$$u^+ = \frac{\bar{u}}{v*} \qquad \text{and} \qquad y^+ = \frac{yv*}{\nu} \tag{3.38}$$

the inner law can be expressed in terms of a function of y^+,

$$\frac{\bar{u}}{v*} = f(\frac{yv*}{\nu}) \tag{3.39}$$

where $v*$ is the wall-friction velocity defined as

$$v* = \sqrt{\frac{\tau_w}{\rho}} \tag{3.40}$$

The same principle can also be applied to the outer law. In order to satisfy the condition for the velocities in the intermediate layer, the functions used to describe the inner and outer velocities need to be logarithmic functions. By integration, the inner variable u^+ can be expressed as

$$u^+ = \frac{1}{\kappa} ln(y^+) + C \tag{3.41}$$

Very near the wall, the turbulence is damped out and the boundary layer is dominated by viscous shear. At very small values of y, the velocity profile is assumed to be linear, i.e.

$$u^+ = y^+ \tag{3.42}$$

This very thin region near the wall in $y^+ \leq 5$, is referred to as the viscous sublayer. The range $5 \leq y^+ \leq 30$ is often called the buffer layer as shown in Figure 3.3. The overlap layer, or logarithmic layer, is usually in the range $30 \leq y^+ \leq 300$.

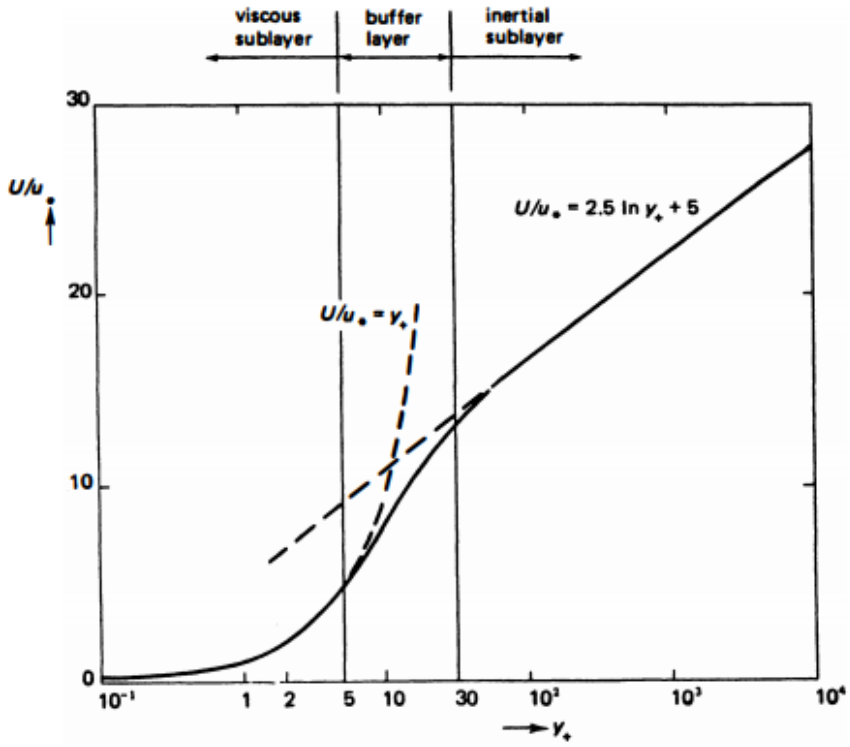


Figure 3.3: The law of the wall. The turbulent boundary layer divided into sublayers. Figure from *A First Course in Turbulence* [Tennekes and Lumley, 1972]

3.5 OpenFOAM

OpenFOAM (Open Source Field Operation and Manipulation) is a free open source CFD software developed primarily by OpenCFD Ltd [OpenCFD, 2018a]. With approximately 100 C++ libraries and 250 pre-built applications, OpenFOAM has an extensive range of features to solve anything from complex fluid flows including chemical reactions, turbulence and heat transfer, to acoustics, solid mechanics and

electromagnetism. The software provides great flexibility by allowing the user to freely create their own applications or modify the existing ones. OpenFOAM consist of *solvers*, designed to solve a specific problem in computational continuum mechanics, and *utilities*, that are designed to perform simple tasks involving data manipulation to ensure consistent data handling across the environments [Greenshields, 2017].

In order to run an analysis in OpenFOAM, a set of sub directories located in the case file is required. The minimum set of directories are *constant*, *system* and *0*, containing files and dictionaries with information about the geometry, case mesh, physical properties, boundary and initial conditions etc.

3.6 Grid generation

The grid generation is a critical part in CFD analysis and there are several challenges related to the issue. The properties of the solution algorithm rely on the choices of the grid, the vector and tensor components and the arrangement of variables on the grid. Thus, a good grid is crucial in terms of generating reliable results and a designer often spends weeks on making the grid.

For complex geometries, the grid generation can be rather advanced. There are several techniques and methods on how to create a mesh, however, the grid is usually referred to as either structured or unstructured. Each node in structured grids can be described by a unique set of indices i, j and k, which can be related to Cartesian coordinates. An advantage of structured grids is that the indices correspond to how the flow variables are stored in the computer memory. In un-

structured grids, the cells and points do not have a particular ordering. They often have a mix of different cell shapes, offering large flexibility in treating complex geometries. For two-dimensional grids, triangles and/or quadrilaterals are normally used while three-dimensional grids are usually built of tetrahedra, hexahedra, prisms or pyramids [Blazek, 2005].

One of the most important requirements is that no holes or gap should appear between the cells. Differences in size and shape between neighbouring elements can affect how the flow is described in the area. Similar cells lying next to each other are preferred in order to obtain a smooth distribution of flow variable across the cells. The length of the cells is also important in terms of how well the flow is described. Considering the Courant number (see Ch. 3.10), the length in flow direction has a significant impact on the results.

Another factor that can influence the solution is the size of the computational domain. It is important that the computational area is large enough in order to not pollute the results. However, a large domain with very many small elements will be time consuming and require a large amount of computer memory to solve. A clever grid setup has small elements where rapid changes in the flow occur and larger elements in areas where the flow field is not so relevant for the flow problem. Regions where it is important to have small cell sizes are for instance in the boundary layer around solid bodies and in the area behind the objects where vortex shedding can occur.

3.7 Creation of mesh using snappyHexMesh

The mesh is created with the OpenFOAM mesh generation utility **snappyHexMesh**, which generates three-dimensional meshes automatically from triangulated surfaces (tri-surface) in Stereolithography (STL) or Wavefront Object (OBJ) format. In order to apply **snappyHexMesh** a starting mesh, called background mesh, is necessary. The cell aspect ratio of the background mesh should be approximately equal to one and consist purely of hexes. There must also be at least one intersection of a cell edge with the tri-surface.

The meshing procedure of **snappyHexMesh** can be divided into five different main steps/sections [Greenshields, 2017] controlled by three switches in the **snappyHexMesh** dictionary: *castellated*, *snap* and *addLayers*. The first two steps, turned on/off by the first switch, involve prescribing the geometry, and refining surfaces and regions in the mesh. In order to preserve the edges of the tri-surface an additional *edgeMesh* file containing information about the edge features is created using **surfaceFeatureExtract**. The file is included in the **snappyHexMesh** dictionary, where a refinement level can be assigned to the feature mesh. Following the feature refinement, cells are selected and split in the locality of the specified surface. According to the settings, either the mesh is kept inside or outside the geometry, and the cells that are not part of the simulation is removed. The surfaces and regions are also refined as specified in the geometry section and by refinement levels.

The next step in the meshing, controlled by the *snap*-switch, involves making changes to the grid in order to snap the mesh onto the surface

of the hull geometry. Cell vertex points are moved onto the tri-surface in order to remove the jagged castellated surface from the previous step. The cells are checked upon the mesh quality parameters. If the parameters are violated, the corresponding vertices are located and their displacements are reduced from their initial value until mesh quality requirements are satisfied.

The resulting mesh from the snapping stage may contain some irregular cells along the boundary surfaces. To avoid a distorted mesh, it is possible to add layers of hexahedral cells aligned with the boundary. This is done by turning on the *addLayers* switch. In viscous fluid flow, it is preferred to have small cells close to the body surface in order to capture the boundary layer. Thus, layers are added close to the tri-surface based on settings defined in the **snappyHexMesh** dictionary.

3.8 Volume of Fluid method

To capture the free surface between water and air, OpenFOAM uses a phase-fraction based Volume of Fluid (VOF) method [Hirt and Nichols, 1981]. The method is an Eulerian modelling technique which uses weighted averages based on the fractions to calculate physical properties. The phase fraction, α, is initially defined equal to 1 for water and 0 for air. At the interface, or free surface, the average density and kinematic viscosity are calculated from the properties of air and water, i.e.

$$\rho = \rho_W + (1 - \alpha)\rho_L \tag{3.43}$$

$$\mu = \mu_W + (1 - \alpha)\mu_L \tag{3.44}$$

The transport equation for the phase fraction is

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \overset{u}{\rightarrow}) = 0 \tag{3.45}$$

Since the phase fraction can have any value between 0 and 1, the interface is not sharply defined but occupies a volume around the region where the interface should exist [Greenshields, 2017].

3.9 Boundary conditions and Wall Functions

In OpenFOAM, the boundaries in the computational domain are given a boundary type and initial conditions are assigned to the different flow variables. When turbulence modelling is used, it is necessary to define boundary and initial conditions also for the turbulent variables. At high Reynolds numbers, the boundary layer is very thin. Thus, in order to resolve the boundary layer using CFD, it is necessary to have elements very close to the body surface. This presents a challenge to the mesh generation and the total number of cells becomes very large.

Wall functions are approaches on how to model the boundary layer very close to the body surface. They define a distribution of the variables from the wall to the first cell. In this way, it is possible to capture the boundary layer with the first cell defined further away from the body surface.

The wall functions are designed to treat the viscous part of the boundary layer, hence they are not valid across the entire boundary layer.

Therefore, it is important that the first cell from the wall is within the logarithmic layer (the overlap layer) mentioned in Chapter 3.4. Normally, in order to satisfy this condition the height of the first cell shouldbe within $30 \leq y^+ \leq 300$.

In order to compute the ship flow numerically with the k-ω SST model, a set of initial conditions for k and ω are required. According to [Saxena, 2014], the equations for k and ω are given by:

$$\omega = \frac{0.09k}{\beta\nu} \tag{3.46}$$

$$k = \frac{3}{2}(UI)^2 \tag{3.47}$$

where I is the turbulence intensity, which for the external flows can be as low as 0.05% in the freestream [Saxena, 2014]. The turbulence viscosity ratio β is defined as

$$\beta = \frac{\nu_t}{\nu} \tag{3.48}$$

where ν_t represents the eddy kinematic viscosity.

3.10 Time step control

The choice of time step influence the simulations. If the time step is too large, a fluid particle will travel a distance that is larger than the cell size between each time the governing equations are solved. In this way, the fluid flow will not be described correctly. Therefore, the time step

needs to be sufficiently small in order to ensure a stable and converging solution. However, if the time step is too small, the computational time will be larger than what is required. Thus, it is desirable to use a time step that is as large as possible while maintaining a stable solution.

The Courant-Friedrichs-Lewy (CFL) condition is a measure of how far a fluid particle might travel relative to the cell size and it is an important stability criterion for hyperbolic equations. The CFL condition is

$$CFL = u\frac{\Delta t}{\Delta x} \leq 1 \tag{3.49}$$

where u is the fluid particle velocity and CFL is referred to as the Courant number. If CFL is equal to 1, the distance travelled by a fluid particle is equal to the length of the element in flow direction. In OpenFOAM, the maximum Courant number can be specified in the *fvSolution* dictionary file. By using adjustable time step, the value of Δt can be controlled by determining the maximum value of the Courant number.

Chapter 4

Method

In this chapter, the method used to set up and run the simulations is presented. A workflow diagram (see Fig. 4.1) in the following chapter shows an overview of the method, before the procedure is explained. A vast amount of time has been spent on creating the mesh. Several approaches to make the grid has been tried out before a successful method was established. The *DTCHull* tutorial included in OpenFOAM has been used as a starting point for the simulation setup. *Paraview* and MATLAB are used to visualise and process the results.

4.1 Work flow

The main steps in the procedure for the CFD simulations in this master thesis are shown below in a workflow diagram in Fig. 4.1.

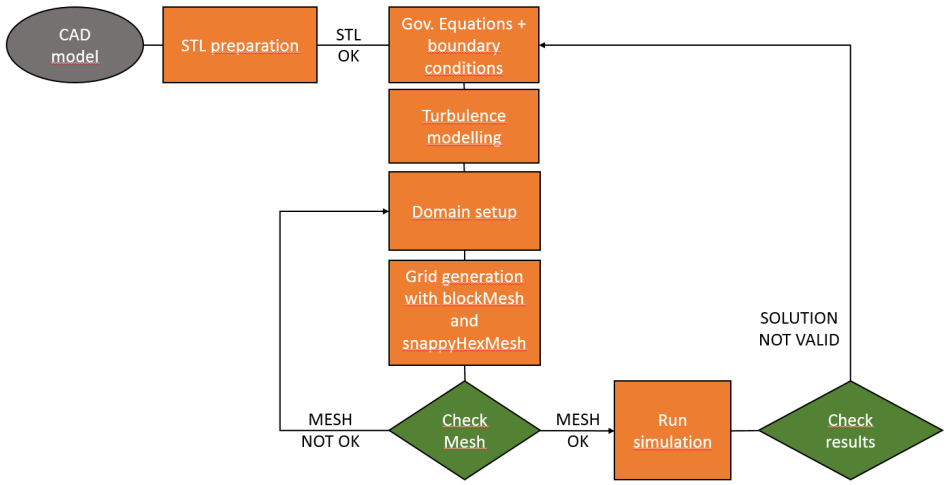


Figure 4.1: Workflow diagram showing the main steps in the simulation procedure in this thesis work.

4.2 Preparation of STL geometry

The model of the trimaran hull (see Fig. 4.2) is provided by Heimli Ship Design as a CAD model in .igs format. **SnappyHexMesh**, however, does not support .igs format and it is therefore necessary to convert the model to .stl format, which presents the hull as a triangular surface mesh. It is important to note that the surface mesh is only used to define the surface of the geometry in **snappyHexMesh**, not the grid that is used to solve the Navier-Stokes equations.

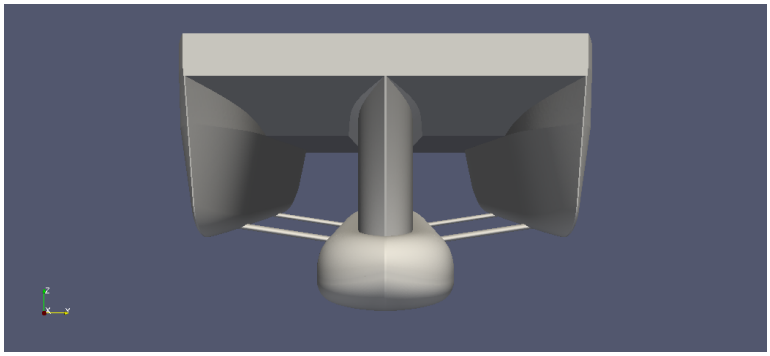
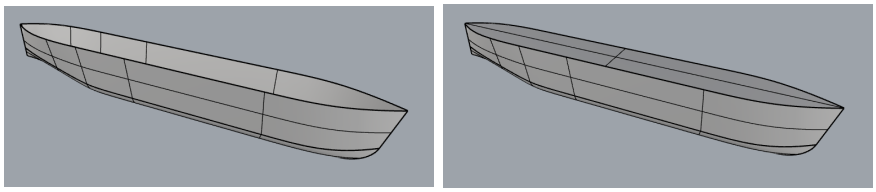


Figure 4.2: CAD model of the trimaran hull provided by Heimli Ship Design

In order to convert the trimaran hull data, the model is imported in Rhinoceros 6 (Rhino), which provides the user with a great deal of flexibility in the meshing procedure. In Rhino, the side hull and the center hull is extracted separately from the complete hull. Fig. 4.3 shows the CAD model of the side hull. Here it can be seen that the geometry is open and not closed. If snappyHexMesh is applied to an open surface, a mesh will not only be created on the outside the geometry but also on the inside. The intention with using snappyHexMesh

with the trimaran hull, is to create a grid on the outside of the ship hull. Therefore, it is necessary to close the geometry by creating a surface from the open edges. Further, the surfaces are joined into a closed geometry as shown in Fig. 4.3b.



(a) CAD model of side hull before closing it.

(b) CAD model of side hull after closing it.

Figure 4.3: CAD model of the side hull open and closed, after it is extracted from the trimaran hull.

In order to check if the geometry is completely waterproof, a naked (open) edge analysis is done. Further, the surface is exported to .stl format. Regarding the side hull, the keel, sides and the top surface is exported separately. In order to make sure that the triangular surfaces are still watertight, the .stl files are imported into Rhino and the geometries are checked again for naked edges. For the side hull, the *MatchMeshEdge* tool is used to ensure that the triangular mesh match over the surface edges.

The origin of the CAD model is defined with the symmetry plane of the trimaran hull in the XY-plane. Further, the model is translated in z-direction such that the free surface is defined in $z = 0$. The dimensions of the trimaran hull is presented in Table 4.1 below.

Table 4.1: Table of dimensions of the trimaran hull

	Symbol	**Unit**	**Value**
Length over all	L_{OA}	m	15.0
Draught	T	m	2.650
Volume displacement	∇	m^3	83
Wetted surface (roughly estimated)	S	m^2	152

After a triangular surface mesh of the hull is created, the separate .stl files representing the side hull is combined into one .stl file with the OpenFOAM utility *surfaceAdd*. A surface check is done in order to ensure that the geometry is closed.

4.3 Domain setup

A good domain is as small as possible in order to save computational resources without polluting the problem solution. Ideally, a convergence test should be performed in order find a suitable domain size, however, this is considered beyond the scope of this thesis work. In order to save computational time and memory, it is decided to make use of the symmetry of the hull, and compute only the flow around half of the vessel.

The choice of computational domain for the trimaran hull is based on the domain presented in [S. Khalil Shariati, 2017] where similar analysis were done for an underwater vehicle near the free surface. However, based on discussion with co-superviser Tufan Arslan, it was decided to define the atmosphere boundary closer to the hull in order to reduce

the number of cells in the atmosphere. In [S. Khalil Shariati, 2017], the distances of the bottom and atmosphere boundaries are defined with respect to the free surface, and the length and width of the domain are calculated with respect to the centre of the vessel. For the trimaran, the origin of a Cartesian coordinate system is defined with the free surface in $Z = 0$, the symmetry plane in the XZ-plane and the stern of the vessel in $X = 0$. The final dimensions of the domain with respect to the origin is shown in Figure 4.4 below. The ship length L_{OA} is chosen as the reference length L.

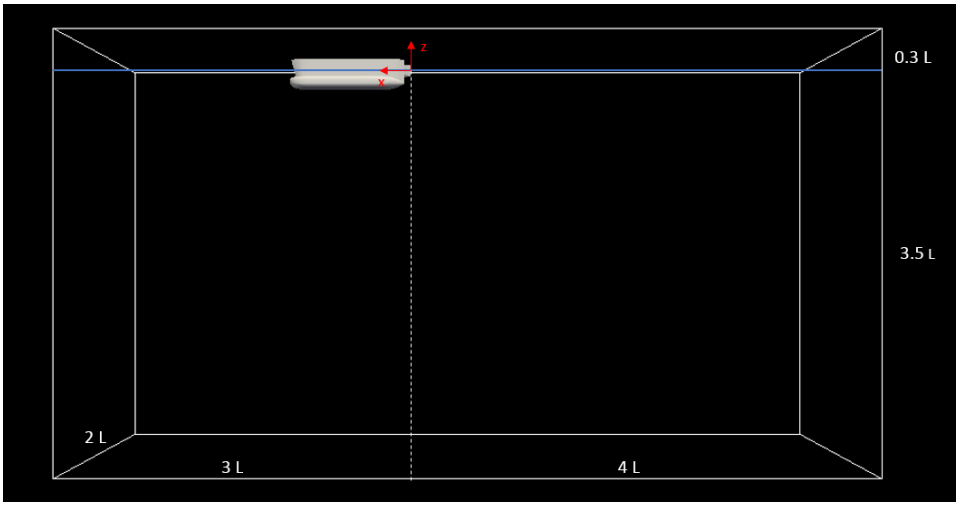


Figure 4.4: Picture showing the outline of the computational domain with dimensions. The reference length L is defined as the ship length over all L_{OA}.

4.4 Boundary and initial conditions

In the simulations, a slip condition is applied to the bottom and side of the domain. For the symmetry plane, a symmetry boundary condition

is used. The boundary types of the side and the bottom are defined as *patch*, which does not contain any topological or geometric information about the mesh. The boundary conditions and initial conditions for the remaining patches are shown in Table 4.2. The inlet values of the turbulence parameters, k and ω, is computed according to the equations presented in Chapter 3.3.4.

The different types of boundary conditions used in these simulations are *fixedValue*, *zeroGradient*, *totalPressure* and *noSlip*. The *fixedValue* condition supplies a fixed value constraint, while the *zeroGradient* applies a zero gradient condition from the patch internal field onto the patch faces. The *totalPressure* is applied to the pressure in the atmosphere, where the static pressure is calculated from the total pressure which in this case is set to zero [Greenshields, 2017]. For the ship hull, a no-slip condition is applied to the velocity. Further, wall functions are assigned to the turbulent variables.

Table 4.2: Boundary conditions used in the simulations. For the bottom and side of the domain, a slip condition is applied. A symmetry condition is used for the symmetry plane.

	Atmosphere	**Inlet**	**Outlet**	**Hull**
U	slip	fixedValue	zeroGradient	noSlip
p_{rgh}	totalPressure	zeroGradient	fixedvalue	zeroGradient
ν_T	slip	fixedValue	zeroGradient	wall function
k	slip	fixedValue	zeroGradient	wall function
ω	slip	fixedValue	zeroGradient	wall function

4.5 Grid generation

The grid is generated using the OpenFOAM mesh utility **snappyHexMesh**. The background mesh, consisting of hexahedral shaped cells with dimensions equal to (1x1x1), is created by using **blockMesh**. Below the ship hull, a simple grading of the cells in the z-direction is applied with the intention of minimising the number of cells where changes in the flow are relatively small. After the background mesh is created, regions and surfaces are refined where it is important to capture the gradients in the fluid flow. The refinement of the mesh is separated into two different stages, isotropic and an-isotropic refinement, as can be seen in Figure 4.1. After the grid is refined and the mesh is snapped onto the hull surface, layers are added around the

hull. The total number of elements is presented in Table 4.3 below.

Table 4.3: Refined grid

Simulation	**Total number of cells**
Side hull	4.318.854
Trimaran hull	8.464.269

4.5.1 Isotropic refinement

The isotropic refinement is performed within snappyHexMesh according to the settings specified in the snappyHexMesh dictionary. As mentioned in Chapter 3.7, the snappyHexMesh process is controlled by three different switches. During the isotropic refinement stage, the castellated switch is turned on while the others are turned off. The areas that are refined is the region close to the hull and the wake regions behind the vessel. Table 4.4 presents the refinement regions and their corresponding levels for the complete hull geometry (using symmetry condition about the symmetry axis). It is chosen to create a volume refinement of the ship hull limited by a specified distance to the tri-surface.

Table 4.4: The isotropic refinement regions used in the simulations of the trimaran hull

Region	**Refinement mode**	**Refinement level**
Hull	distance	5
Inner Box 1	inside	4
Outer Box 1	inside	2
Wake Region 1	inside	1
Wake Region 2	inside	2

4.5.2 Anisotropic refinement

In order to capture the free surface in an appropriate manner, it is preferred to do an anisotropic refinement in the z-direction of the free surface. This is done by using the OpenFOAM utilities **topoSet** and **refineMesh**. In the **topoSet** dictionary, a cell set within a volume enclosing the free surface is chosen and the region is refined by setting the refinement direction to *normal*, defined as the z-direction, in the refineMeshDict. By applying **refineMesh** once, the area is refined by splitting an element into two smaller elements. In order to obtain a fine refinement of the free surface, the sequence is repeated several times, each time the z-range of the cell set defined in **topoSet** is smaller than the previous cell set. This is done in order to ensure a graded refinement of the free surface and to avoid getting small and large elements next to each other.

In order to run **snappyHexMesh** smoothly, the mesh enclosing the tri-surface should have an aspect ratio approximately equal to 1. Thus, during the an-istropic refinement the cells that are close to the ship hull are removed from the cell set defined in **topoSet** before the region is refined. In this way, the free surface is refined, but the shape of the cells close to the ship hull stays cubic.

4.5.3 Boundary layer generation

After the free surface is an-isotropically refined, **snappyHexMesh** is run again but with *castellated* switched off, and *snap* and *addLayers* turned on. The boundary layer generation is based on the usage of wall functions for the turbulence parameters, and it is aimed to maintain a y^+ value in the range of $60 - 80$ on the sides of the hull. In **snappyHexMesh**, it is possible to generate layers with a thickness relative to the cell size in the background or defined as an absolute length [Engys, 2012]. In this master thesis it is chosen to use absolute sizes. The meshing utility is set to generate 6 layers with a minimum layer thickness and an overall final layer thickness as presented in Table 4.5 below. The feature angle is defined equal to 280° in order to generate layers at the edges of the hull.

Table 4.5: Settings for boundary layer specified in snappyHexMesh.

Name	**Unit**	**Value**
Minimum layer thickness	m	0.0002
Final layer thickness	m	0.01
Expansion ratio	(-)	2.18
Feature angle	°	280

An estimate of the layer thickness based on a y^+ value in the range of $30-300$, is found to be approximately in the interval $0.0002-0.002$m. The range of values is based on calculations using grid spacing tools online [Pointwise, 2018] made for computing the grid spacing based on target values for y^+. However, these values are only an estimate and the resulting y^+ values need to be checked after the first simulation.

OpenFOAM v1612+ and newer versions come with an alternative layer generation setting for the mesh shrinking, namely DisplacementMotionSolver. The alternative mesh shrinker can be used to avoid local distortion of the mesh resulting from the use of displacementMedialAxis [OpenCFD, 2017]. Figure 4.5 shows the different results from using the default algorithm and the displacement motion solver. Each picture show two boxes, where the left box is meshed with specified feature refinement, while the right box is without feature refinement. By studying the layers around the right box, it can be seen that the default algorithm is not very successful in adding layers at the corners when the mesh is coarser compared to the displacement motion solver

[OpenCFD, 2017].

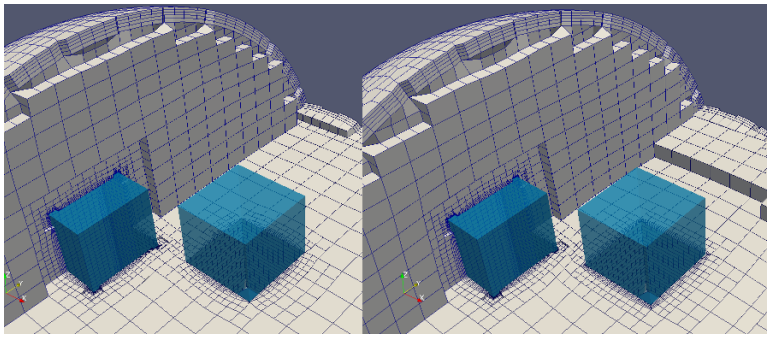


Figure 4.5: Layer generation from different mesh shrinking methods. The picture to the left shows the default algorithm, *displacementMedialAxis*, while the right picture shows the displacement motion solver. The left box in each picture is refined with specified feature refinement, while the right box is without the feature refinement. Figure from *OpenFOAM v1712 Extended Code Guide* [OpenCFD, 2017]

4.6 Simulation setup

The setup for the simulations are presented in Table 4.6. It is chosen to use adjustable time step, which means that the time increment is determined based on the Courant Number. The maximum value of the Courant Number is set to 0.5.

Table 4.6: Simulation setup

Settings	
OpenFOAM solver	interFoam
Turbulence model	$k-\omega$ SST
Time step (s)	ajustable time step
Maximum Courant Number	0.5
Reynolds number	$7.7 \cdot 10^6$
Initial turbulence intensity	0.1 %
Initial turbulence viscosity ratio	1

The fluid properties used in the simulations are shown in the table below:

Table 4.7: Fluid properties

Property	**Value**
Density, water (kg/m^3)	1025
Density, air (kg/m^3)	1
Kinematic viscosity - water, ν (m^2/s)	$6 \cdot 10^{-6}$
Kinematic viscosity - air, ν (m^2/s)	$1.48 \cdot 10^{-5}$

4.7 Solver settings

The solver settings for the simulations are specified in the *fvSchemes* and *fvSolution* dictionaries. In *fvSchemesDict*, the numerical schemes used for time derivatives, gradient and convective terms etc. are specified. The numerical schemes and solvers used in this master thesis is presented in Table 4.8 below.

Table 4.8: Numerical schemes and solvers

	Settings
Time scheme	CrankNicolson 0.5
Gradient scheme	Gauss linear
Convective schemes	Gauss linear/Gauss linearUpwind
Laplacian scheme	Gauss linear corrected
Interpolation scheme	linear
SN Gradient scheme	corrected
Solver for pressure	GAMG
Solver for other variables	smoothSolver
Tolerance, pressure	$1 \cdot 10^{-7}$
Tolerance, velocity	$1 \cdot 10^{-7}$

The time discretisation schemes used in this master thesis is CrankNicolson 0.9. The method is a second order scheme which provides a blending between Euler and Crank-Nicolson scheme according to a user determined coefficient between 0 and 1, where 0 corresponds to an Euler scheme and 1 corresponds to Crank-Nicolson. A value of 0.9 is considered to be a good compromise between accuracy and robustness [OpenCFD, 2017], and was also recommended by co-supervisor Tufan Arslan. However, due to some convergence challenges , the blending factor was reduced to 0.5.

In the *fvSolution* dictionary, information which controls the equation solver, tolerance and algorithm is contained. The solvers are generally of iterative type and the aim is to reduce the equation residual over successive solutions. Running the analysis can be a time-consuming process, thus it is desired to stop the iterations when a certain level of accuracy, defined by the solution tolerances, is reached.

The solver chosen for the pressure equation in this thesis work is the GAMG (geometric-algebraic multi-grid) solver. To solve the other equations, a smooth solver is applied. The solution algorithm used for the simulations is the PIMPLE algorithm. The PIMPLE algorithm combines the SIMPLE and PISO algorithm, thus allowing for running simulations with Courant Number higher than one. All three algorithms solves the same governing equations, the main difference is how they loop over the equations, which is controlled by specific input parameters.

In this case, the PIMPLE algorithm is set to work in PISO mode, meaning that maximum Courant Number, specified by $maxCo$, cannot exceed a value of one. In order to ensure a stable solution, $maxCo$ is

defined equal to 0.5. The PISO algorithm starts by solving a pressure equation to enforce mass conservation. The equations are very often solved multiple times within one time step. The number of times the pressure equation is solved at the same time step, can be determined by the **nCorrectors** input parameter.

Chapter 5

Test analysis of setup for a cylinder in axial flow

In order to get familiar with using the meshing utility **snappyHexMesh** to generate a grid around a triangulated surface, the thesis work is started out by setting up a test case for a cylinder in axial flow. The cylinder's diameter is equal to 1, while the length is equal to 3. The fluid flow is assumed to be turbulent with a Reynolds number equal to 10^6. The turbulence is modelled using the k-ω SST turbulence model. It is decided to use one of OpenFOAM's ship hull tutorials, *DTCHull*, as a starting point for creating a numerical setup for the cylinder. However, changes are made to the grid generation, domain setup, boundary conditions and the solver settings.

To validate the results a comparison with experimental tests is done. The comparison is made to [Hoerner, 1965], where the drag coefficients from experiments with different cylinders in axial flow is presented. According to Figure 21 in [Hoerner, 1965], the drag coefficient for a cylinder with an l/d ratio equal to $1/3$, where L is the length and d is the diameter, lies in the range $0.8 - 0.9$.

A vast amount of time has been spent to get an understanding of the meshing techniques. Many attempts have been carried out without

success. The first approach leading to a C_D close to the numerical results is set up with a computational domain based on the domain in [S. Khalil Shariati, 2017] (see Ch. 4.3). The numerical schemes and *fvSolution* dictionary are adopted from the $DTCHull$ tutorial. The mesh refinement is done isotropically in **snappyHexMesh** and anisotropically using **topoSet** and **RefineMesh**.

Resistance analysis of bodies in the free surface are very complicated and difficult to simulate. After discussions with supervisor, it was decided to begin the simulations with cylinder submerged below the free surface and to further move the cylinder closer to the free surface. The distance from the free surface to the cylinder centre is defined as a dimensionless constant D^*, defined as the ratio between the submergence depth H and the cylinder diameter d, i.e.

$$D^* = |\frac{H}{d}| \tag{5.1}$$

The following depths are studied:

The boundary conditions are altered from the tutorial to the conditions presented in Table 4.2. The results are presented and discussed in Ch. **??**.

After the simulations, the thesis work was proceeded with the trimaran hull instead of the cylinder. From further studies and conversations with co-supervisor Tufan Arslan, slightly new changes where adopted into the method. Ch. 4 above describes the final method used for the analysis of the trimaran hull. It was also discovered that a boundary condition causing small surface roughness was applied to the body, which was replaced as the ship hull is assumed to be smooth. The

Table 5.1: Overview of the different values of D^* used in the simulations of the cylinder.

Case number	D^*	**Unit**
1	4.4	m
2	2.2	m
3	1.3	m
4	0 (free surface)	m

time scheme used in the *DTCHull* tutorial is *localEuler*. This time scheme is a first order, pseudo transient scheme designed for steady state analysis, thus not considered to be appropriate for the resistance analysis of the trimaran hull. Therefore, the time scheme was altered for the simulations to a Crank Nicolson scheme with a blending factor equal to 0.9. The results are presented in Ch. 6.1

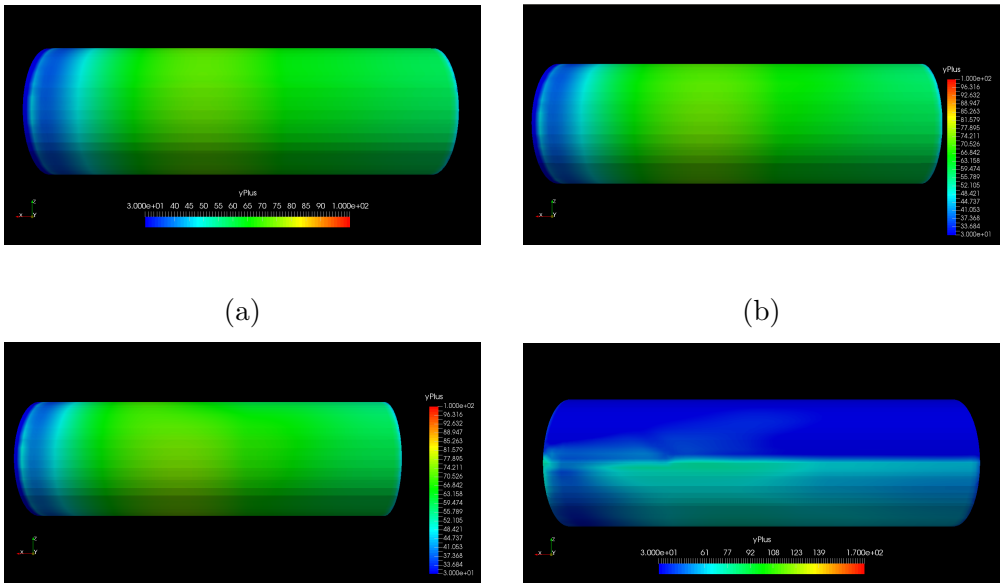
Chapter 6

Results

The results of each case is presented in the following chapters. The results from the simulations of the cylinder are presented first, followed by the sidehull and the trimaran simulations. The simulations for the side hull and the complete hull were run on Vilje using 32 processors. The cylinder cases were run on the CFD2 computer at The Department of Marine Engineering at Tyholt, using 8/16 processors, depending on the case. It is important to note that the flow variable plots are pictures of instantaneous values and not the mean flow.

6.1 Cylinder in axial flow

Fig. 6.1 shows the distribution of y^+ for the cylinder submerged at the depths $D^* = 4.4$, $D^* = 2.2$, $D^* = 1.3$ and in the free surface, respectively.



(a) (b)

(c) (d)

Figure 6.1: Plot of y^+ values for the cylinders submerged at (a) $D^* = 4.4$, (b) $D^* = 2.2$, (c) $D^* = 1.3$, (d) Free surface

The plots of the drag coefficients are shown in Fig. 6.2. For the case where $D^* = 4.4$, the drag coefficient stabilised after approximately 130-140 seconds and the simulations were stopped when reaching 200 seconds of simulated time. For the two following cases, however, the oscillations varied more but the mean value of the oscillations was

approximately constant for the last 100 seconds. For the case where the cylinder is placed in the free surface, the results would not converge when using the same setup as for the other cases. It was tried to lower the time step from 0.01 second to 0.005 second, but the problem still remained unsolved. It was also tried to alter the mesh, but the changes did not have any effect of the convergence. A solution was to change the time step to adjustable run time. This approach was also used for the ship hull simulations.

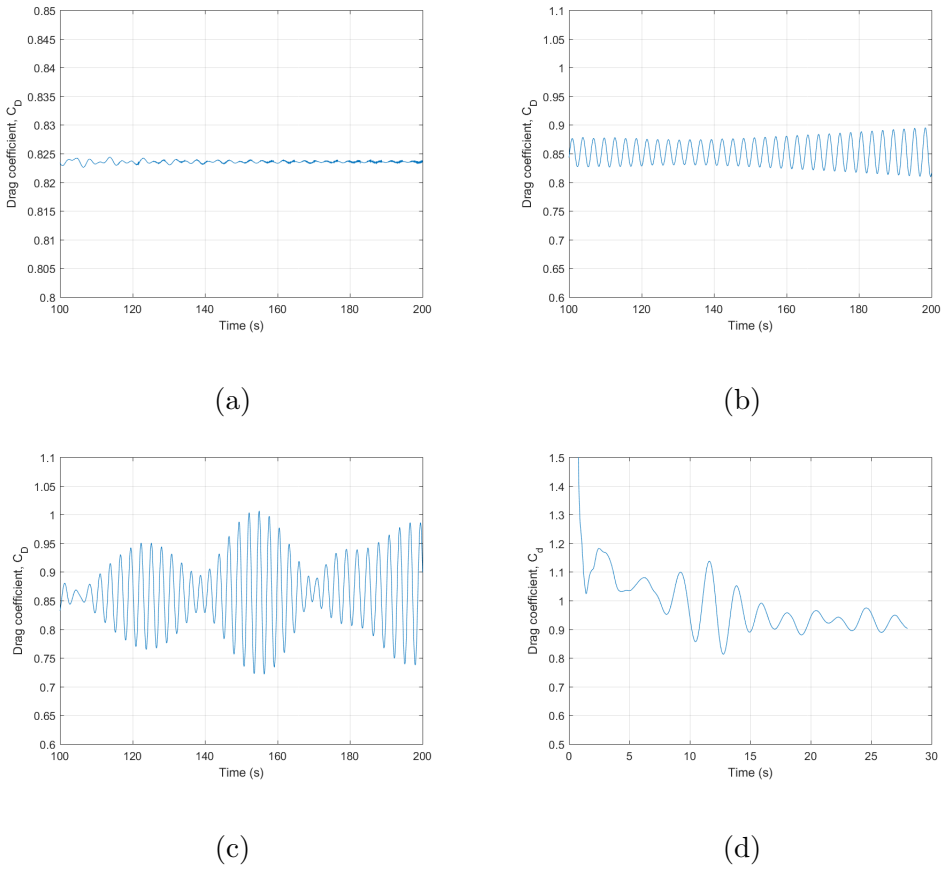


Figure 6.2: Plot of the drag coefficients for the cylinders submerged at (a) $D^* = 4.4$, (b) $D^* = 2.2$, (c) $D^* = 1.3$, (d) Free surface

The average y+ values for the final time step and the drag coefficients are presented in Table 6.1 below. The drag coefficients are computed by taking the average value of the oscillations, when the results seem to have stabilised.

D^*	**Average** C_d	**Average** y^+
4.4	0.824	58.74
2.2	0.852	59.19
1.3	0.863	59.73
Free surface	0.9452	46.01

Table 6.1: Results for cylinder in axial flow

After having achieved satisfactory results for the cylinder simulations, confirming the integrity of the simulation setup, the work is proceeded by replacing the cylinder with the more complex trimaran hull model.

6.2 Side hull simulation

For simplicity's sake, analysis are performed for the side hull separately before simulating the trimaran hull. The distribution of the y^+ values for the side hull is shown in Fig. 6.3.

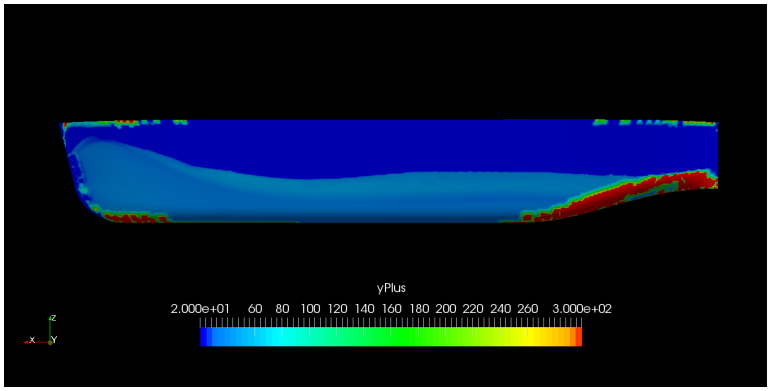


Figure 6.3: Distribution of y^+ for the side hull.

6.2.1 Flow variable plot

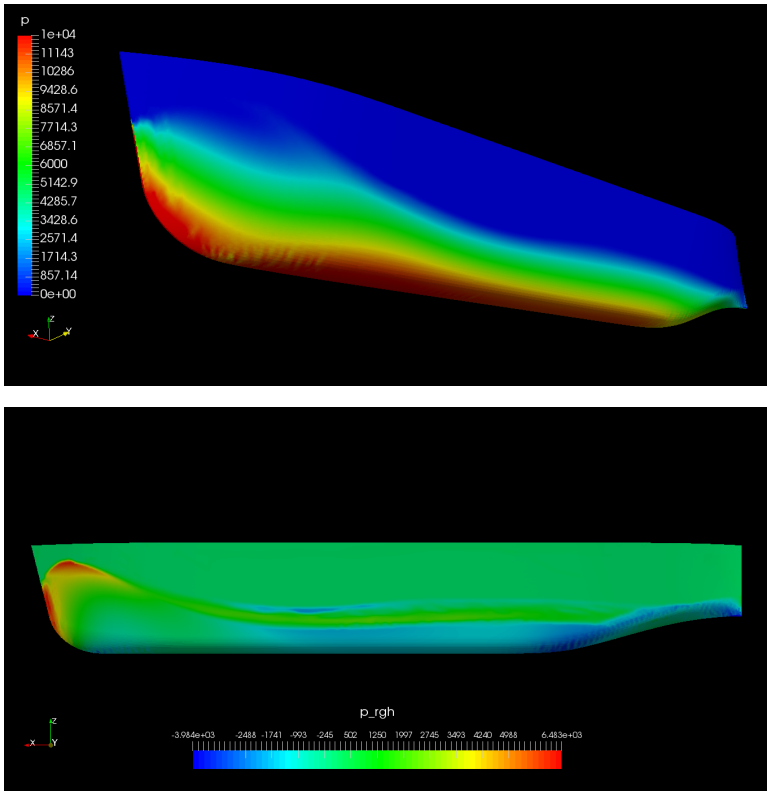


Figure 6.4: Pressure plot for side hull, the lower picture showing the pressure $p_r gh$, i.e. without hydrostatic pressure.

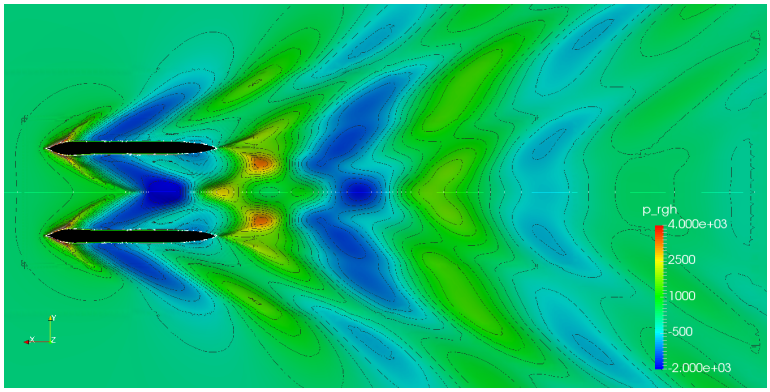


Figure 6.5: Pressure distribution in the free surface around the side hull

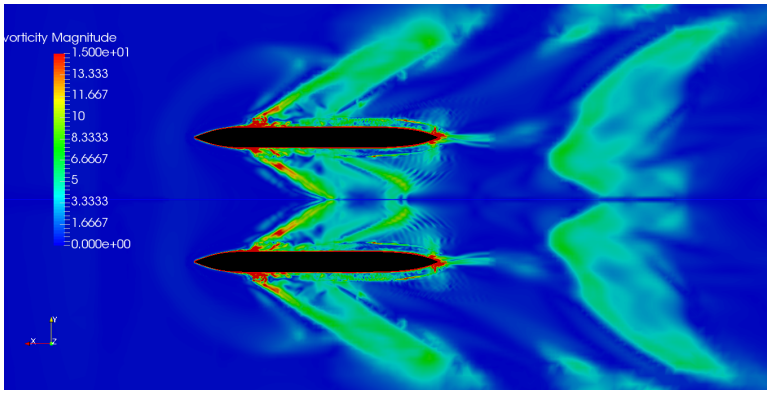


Figure 6.6: Vorticity plot for the sidehull (XY cutting plane)

6.2.2 Forces

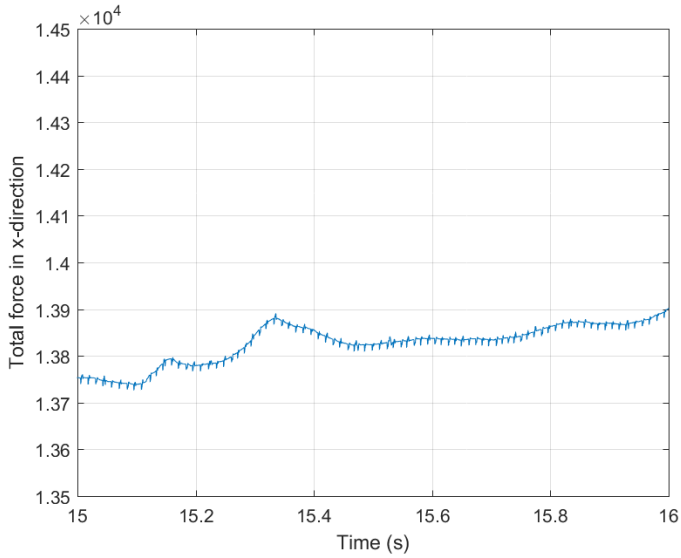


Figure 6.7: The total force in x-direction of the side hull.

6.3 Trimaran simulation

The y^+ distribution for the trimaran hull is shown in Fig. 6.8.

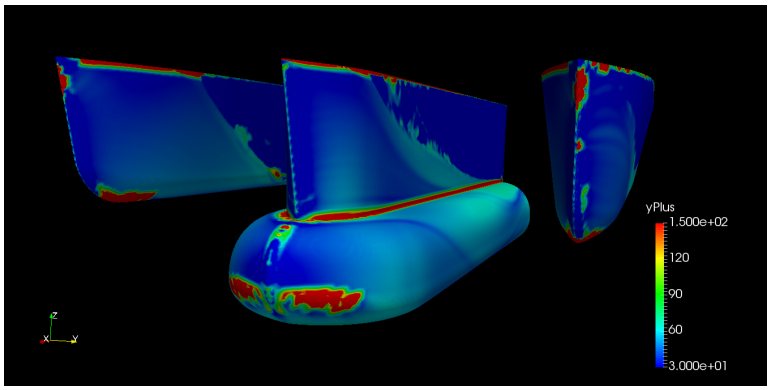


Figure 6.8: Distribution of y^+ for the trimaran hull

6.3.1 Flow variable plots

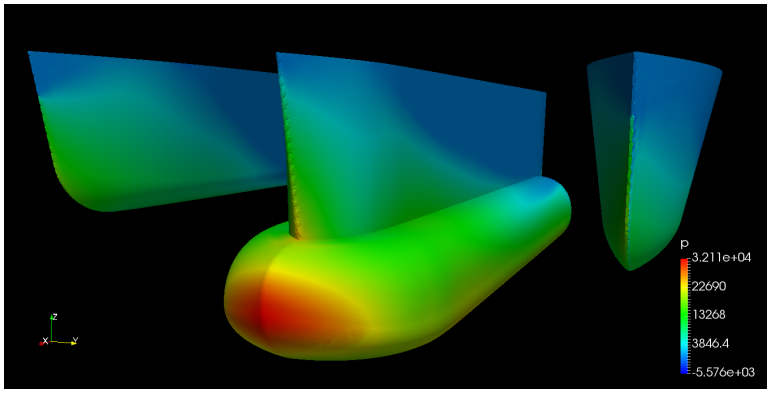


Figure 6.9: Distribution of pressure for the trimaran hull

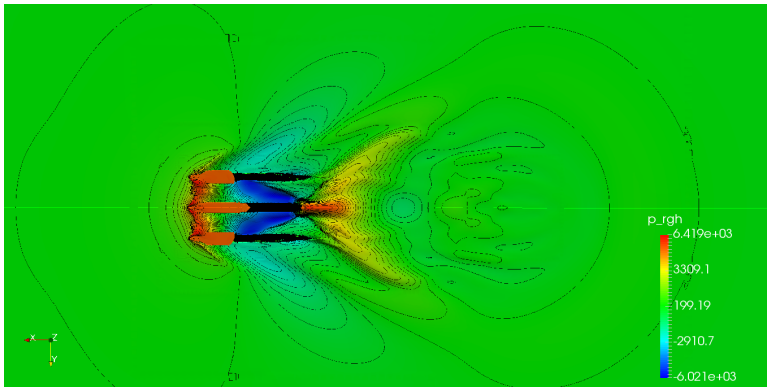


Figure 6.10: Plot of the pressure in the free surface for the trimaran. It can be seen clearly that the simulation requires more computation time before the flow is stable.

6.3.2 Forces

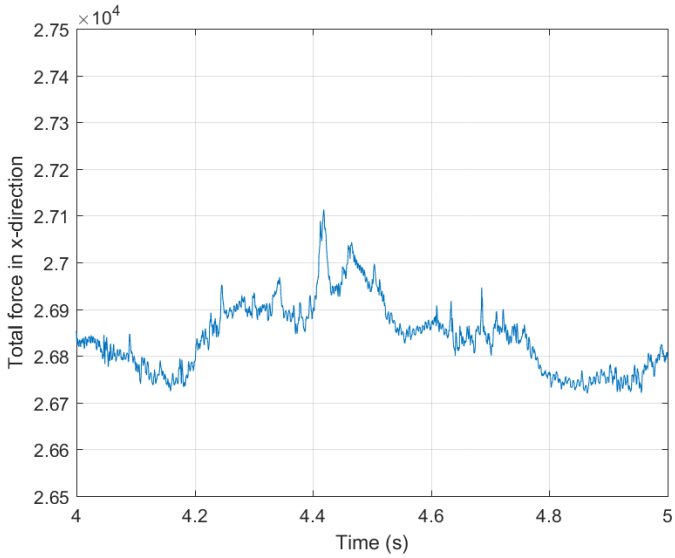


Figure 6.11: The total force in x-direction of the trimaran hull.

Chapter 7

Discussion

This chapter presents a discussion of the results and the work that has been done in this thesis. Many meshing attempts have been carried out and simulations have been run without success. Several analyses have crashed or stopped, results have diverged to unrealistically high values or the resistance has approached zero. Thus, indicating errors in the procedure. Continuously in this thesis work, new features and "tricks" have been discovered and more knowledge about CFD have been obtained.

As mentioned in the introduction, the objectives for this thesis work are:

- Perform a resistance analysis of the new design using CFD in straight-ahead cruising condition with a forward speed equal to 10 knots.
- Evaluate the velocity, pressure and y^+ distribution of the trimaran hull.
- Evaluate the results of the trimaran by comparing the CFD results for an existing design of a catamaran with the same length and displacement. Resistance data in calm water with a forward

speed equal to 10 knots and a ship model (hull data) is provided.

- Become more familiar with the software OpenFOAM and CFD.

Considering the latter objective, it has been learned much more about CFD and how to use OpenFOAM. However, there is a lot more to the topic than what this thesis concerns. The objectives for this project was developed in the beginning of the work. An outline of work progress for the period was established in the beginning, however, several challenges occurred unexpectedly which delayed the work significantly. For instance, a lot of effort had to be put in the preparation of the STL files. Different types of CAD software was tried out before an acceptable tri-angulated surface was established. Another significant issue was a bug in the **snappyHexMesh** layer generation code v1712, which was later fixed and a new version was released on the 29th of June, 2018 [OpenCFD, 2018b].

From the beginning of this master thesis work, the intention was to establish a convergence study of the grid to strengthen the simulation results. However, the simulation setup from STL preparation to post-processing of the results required much more time than what was foreseen and sufficient data for a convergence study of the trimaran hull has not been obtained. Thus, the results presented in Ch. 6 can be questionable and further work should be done before fully reliable results are achieved. However, the simulation results can be used as a basis for further studies. In the following chapter, the results are discussed in the same order as they are presented.

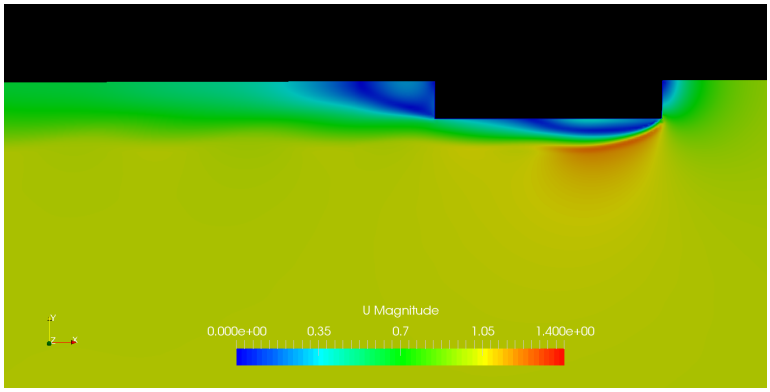


Figure 7.1: Velocity field around the cylinder submerged at $D^* = 2.2$ in the XY cutting plane.

7.1 Cylinder in axial flow

In Fig. 6.1 it can be seen that the y^+ values on most part of the cylinder surface are within the range of $30 \leq y^+ \leq 300$. The y^+ value close to the edge at the fore part of the cylinder is below the range. At the edge, the flow will separate and reattach, as can be seen in the velocity plot for the cylinder with $D^* = 2.2$ in Fig. 7.1. After the separation point, the velocity reduces before the flow attaches to the cylinder surface. This decrease in the velocity is shown in the y^+ distribution as the value is lower than 30. Ideally, a varying layer thickness should be generated in order to maintain satisfactory y^+ values all over the geometry. However, this is considered to be beyond the scope of this master thesis.

The plots of the drag coefficients are shown in Fig. 6.2. The results for the submerged cylinders are close to the reference value range $C_D \in [0.8, 0.9]$ in [Hoerner, 1965] found from experimental testing. As

the cylinder's depth decreases, the drag coefficient increases which is expected as the cylinder approaches the free surface, where for instance wave making will contribute to the total resistance. As the cylinder is moved closer to the surface, the drag coefficients vary more and tends to be more unstable especially for $D^* = 1.3$ and for the cylinder in the free surface. At the same time, disturbances are observed in the free surface. It would have been interesting to see if the oscillations stabilised if the simulation time increased. However, the mean value of the oscillations are close to the experimental results in [Hoerner, 1965]. The free surface case show a tendency to settle around a mean value of C_D equal to 0.95 (see Table 6.1). Thus, it was chosen to proceed with the ship hull analysis.

7.2 Trimaran ship hull simulations

For the side hull simulation, it can be seen in Fig. 6.3 that the majority of the sides below the water have y^+ values in the range $30 \leq y^+ \leq 300$. However, at the areas around the keel and the bow, the values are larger. By studying the mesh layers close to the keel in the XZ cutting plane in Fig. 7.2, the large y^+ values can be explained by the bad layer generation. Several attempts were made in order to fix the problem, however the layer generation was still not satisfactory around the keel area.

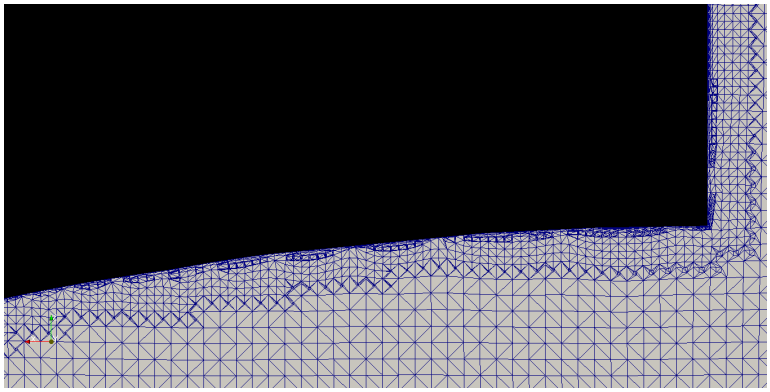


Figure 7.2: Layers of mesh at the keel of the side hull (XZ cutting plane). The picture shows that layers are not completely generated at the keel.

Similarly for the trimaran hull, it can be observed in Fig. 6.8 that the y^+ distribution is within reasonable limits for most part of the hull surface, but the side hull keels still remain a problem. High values of y^+ can also be observed at the fore part of the center hull. As mentioned in the beginning of the this chapter, OpenFOAM v1712 has a bug in the layer generation code in **snappyHexMesh**. The bug is related to the treatment of feature angle setting which determines at which degree layers will not get extruded. If the feature angle in v1712 is set to a value larger than 60 degrees, all layer extrusion is disabled. The author was not made aware of this error before in July, close to the deadline for this thesis work.

In the pressure plot shown in Fig. 6.4 it can be seen that the pressure is relatively high in the keel area. At the bow, the flow will stagnate before it is forced to the sides by the hull. This agrees with the high pressure at the bow. Some abnormalities is observed at the keel close to the bow. This can be explained by the bad layer generation in the

mesh. In Fig. 6.9 it is clearly shown that the flow is stagnated at the fore part of the center hull, which corresponds well with flow theory.

A snapshot of the pressure distribution for the catamaran used as a reference, is shown below in Fig. 7.3. The data is provided the author by Heimli Ship Design. Unfortunately, no values of the pressure is shown, but the color range is interpreted as the red color meaning high pressure values and blue begin low pressure areas. The high pressure areas at the bulb and where the bow wave arises is seen in Fig. 6.9 and 6.4.

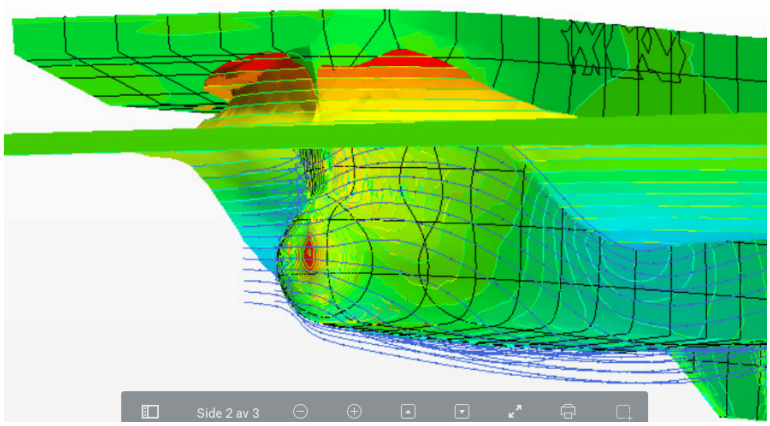


Figure 7.3: Pressure plot for catamaran made available for comparison of the results found in thesis work.

Fig. 7.3 also shows drastic change in the water elevation at the fluid meets the ship hull. This large rise in the water level at the wave bow is also seen in the side hull simulations in Fig. 6.5. From Fig. 6.10 it is not reasonable to discuss the wave pattern as it is clearly showed that the pattern is far from fully developed as the duration of the simulation is too short. It is also observed that water is flowing over the hull. Without similar vessel data for comparison, it is difficult

to point out the reason why, but most probably the flow will settle if the simulation is continued.

It is chosen to include a plot of the vorticity for the side hull simulation shown in Fig. 6.6. Some fluctuations can be observed close to the hull, however, vortex shedding does not appear clearly in the figure. A solution to this problem can be to refine the grid even further in the wake region. However, in the master thesis written by Øyvind Andre Hagen Emblemsvåg [Øyving Andre Hagen Emblemsvåg, 2016], it was concluded that the k-ω SST turbulence model did not capture the vortex shedding as well as the realizable k-ε model. Since the k-ω SST turbulence model is used for the simulations in this thesis work, it can not be ruled out that the a different turbulence model would be a better choice for the trimaran simulations.

In the Fig. 6.7 and 6.11, the plots of the total forces in varies during the simulation. Ideally, the duration of the simulations should have been longer in order to see if the variations would stabilise more. The forces shown in the plots are computed from the output force data and multiplied with two in order to account for the complete ship hull, and not only the half part used in the analysis. The resistance force computed for the catamaran used for comparison is equal to 32 kN. By taking the total force for the last time step, the side hull simulation gives a resistance force approximately equal to 13.9 kN, while the resistance computed for the trimaran hull is found to be approximately 26 kN. As discussed in this chapter, the results are not considered to be reliable due to the lack of sufficient convergence study and the rather short duration of the simulations. However, if the forces are compared with the catamaran, the new trimaran design tend to experience lower resistance.

Chapter 8

Conclusion

A numerical simulation of the trimaran service vessel hull has been performed using OpenFOAM. The resistance force is computed for the trimaran hull and the side hull separately using symmetry condition. The simulation setup has been tested on a cylinder in axial flow with different submergence depths, which showed satisfactory results compared to literature.

The flow in the trimaran simulation has not stabilised and the simulation time should be significantly longer. The last recorded value of the resistance force is approximately equal to 26.5 kN, which is lower than the value for the catamaran used for comparison. The resistance of the catamaran is equal to 32 kN. Unfortunately, the results in this thesis work are not reliable as the majority of time has been spent on setting up the simulations and generating the mesh. The grid for the ship hull has been generated using OpenFOAM v1712, which turned out to have a bug in the code related to layer generation using **snappyHexMesh**. For further studies it is recommended to use OpenFOAM v1806 which includes a bug fix for the problem. Also, a proper grid sensitivity study is recommended for further work.

Another interesting subject for further work, would be to study the effects on the results from applying a different turbulence model, for

instance the realisable k-ε model. After a certain reliability in the results is established, an optimisation of the hull design is of interest. Suggestions here are to vary the submergence depth of the center hull and the spacing between the side hulls and the center hull.

Although the results in this thesis work are not considered to be up to par by the author, a solid ground for obtaining further knowledge and experience with CFD has been established.

Bibliography

[Blazek, 2005] Blazek, J. (2005). *Computational Fluid Dynamics Principles and Applications*. Elsevier Ltd, 84 Theobalds Road, London WC1X 8RR, UK.

[Celik, 1999] Celik, I. B. (1999). Introductory turbulence modeling. Part of syllabus in the course *Numerical solution of PDE's* Fall 2014. Accessed: 15.07.18.

[Engys, 2012] Engys (2012). A comprehensive tour of snappyhexmesh. `https://openfoamwiki.net/images/f/f0/Final-AndrewJacksonSlidesOFW7.pdf`. Training sessions given at the 7th OpenFOAM Workshop by Engys Ltd. Accessed: 06.02.2018.

[Ferziger and Perić, 1997] Ferziger, J. H. and Perić, M. (1997). *Computational Methods for Fluid Dynamics*. Springer-Verlag Berlin Heidelberg, Germany.

[Greenshields, 2015] Greenshields, C. J. (2015). Openfoam programmer's guide. `http://foam.sourceforge.net/docs/Guides-a4/ProgrammersGuide.pdf`. Accessed: 28.04.18.

[Greenshields, 2017] Greenshields, C. J. (24th July 2017). Openfoam user guide version 5.0. `http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf`. Accessed: 02.12.17.

[Hirt and Nichols, 1981] Hirt, C. and Nichols, B. (1981). Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225.

[Hoerner, 1965] Hoerner, S. F. (1965). Fluid-dynamic drag : practical information on aerodynamic drag and hydrodynamic resistance.

[Larsson and Raven, 2010] Larsson, L. and Raven, H. (2010). *Ship Resistance and Flow*.

[Menter, 1993] Menter, F. R. (1993). Zonal two equation kappa-omega turbulence models for aerodynamic flows - nasa-tm-111629. Technical report.

[Moukalled et al., 2016] Moukalled, F., Mangani, L., and Darwish, M. (2016). *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*, volume 113 of *Fluid Mechanics and Its Applications*. Springer International Publishing.

[OpenCFD, 2017] OpenCFD, L. (2017). Openfoam v1712 extended code guide. `https://www.openfoam.com/documentation/cpp-guide/html/`.

[OpenCFD, 2018a] OpenCFD, L. (2018a). `https://www.openfoam.com/`. Accessed: 02.12.2017.

[OpenCFD, 2018b] OpenCFD, L. (2018b). `https://www.openfoam.com/releases/openfoam-v1806/notable-bug-fixes.php`. Accessed: 10.07.2017.

[Pointwise, 2018] Pointwise, I. (2018). `http://www.pointwise.com/yplus/`. Online tool for computing grid spacing for a given $y+$ value. Accessed: 11.05.2018.

[S. Khalil Shariati, 2017] S. Khalil Shariati, S. H. M. (2017). The effect of appendages on the hydrodynamics characteristics of an underwater vehicle near the free surface.

[Saxena, 2014] Saxena, A. (2014). Guidelines for specification of turbulence at inflow boundaries. `http://support.esi-cfd.com/esi-users/turb_parameters/`. Accessed: 08.03.2018.

[Steen and Minsaas, 2013] Steen, S. and Minsaas, K. (2013). *TMR 4220 Naval Hydrodynamics: Ship resistance*. NTNU Department of Marine Technology.

[Tennekes and Lumley, 1972] Tennekes, H. and Lumley, J. L. (1972). A first course in turbulence.

[Versteeg and Malalasekera, 1995] Versteeg, H. K. and Malalasekera, W. (1995). *An introduction to computational fluid dynamics, The finite volume method*. Longman Scientific & Technical, Essex CM20 2JE, England.

[White, 2006] White, F. M. (2006). *Viscous Fluid Flow*. McGraw-Hill, 1221 Avenue of the Americas, New York, NY 10020.

[Wilcox, 1988] Wilcox, D. C. (1988). Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal, Vol. 26. No. 11.*

[Øyving Andre Hagen Emblemsvåg, 2016] Øyving Andre Hagen Emblemsvåg (2016). Numerical simulations of flow around a simplified hull form.