Investigation of open-Source CFD software on shipyards

Analysis of combustion outlets on large yachts using openFoam®

Master of Science Thesis

CARL JASCHA MORDHORST

Department of Shipping and Marine Technology
Division of Sustainable Ship Propulsion
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2011

Report No. X-11/262

Investigation of open-Source CFD software on shipyards

An analysis of combustion outlets on large yachts using openFoam®

By Carl Jascha Mordhorst for the Degree of Master of Science

Department of Shipping and Marine Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
SE- 412 96 Gothenburg
Sweden

Cover: MY 'Madsummer' from luerssen.com
Gothenburg, Sweden 2011

Abstract

Modern engineering requires detailed knowledge of forces, flow directions and velocities, when designing the vehicles of the future. In the ship building industry this is traditionally done in towing tank and wind tunnel experiments or by the long experience of skilled engineers. In the last years computational fluid dynamic (CFD) methods developed to a stage, where they become interesting, not only from a financial but also from a performance point of view, for larger ship yards as an extra input and in long term thinking as a full alternative to the experiments.

The special branch of luxury yacht building yards is like every business in strong competition and needs to work economical. It can be very expensive to carry out both methods on the one hand. Since on the other hand these high technology yards have special requirements to fulfill that might change more frequently than for commercial ships, CFD is often a desirable method for the yards to master. The open source software (OSS) openFoam® offers here a possibility to include a modern CFD method without extensive investment costs into the design process and to gain "free" experience with a technology of the future.

The present thesis shows the strength and weaknesses of openFoam® in the ship building industry for and in close collaboration with the Friedrich Lürssen shipyard in Bremen. The yard has specialized in the construction of luxury yachts over 60m and naval vessels over 30m.

Due to openFoam®´s classical case folder structure, several tutorials were created to investigate this purpose. They reflect typical flow situations needed by the yard for future analysis, i.e. a wind tunnel setup, a towing tank setup, a sinkage and trim tutorial, a heat analysis tool and a particle injector. Besides the important resistance calculations the main focus lay on the analysis of the exhaust gas behavior. Mega yacht customers demand lowest possible air pollution on deck, so that extensive wind tunnel experiments are done during the early design stages. The aerodynamic tutorial was created as an alternative to these experiments. Satisfying results were achieved with openFoam® when compared to tests conducted by the Hamburgische Schiffbau- Versuchsanstalt (HSVA). To show the differences in performance to commercial software the hydrodynamic results of openFoam® solver were compared to StarCCM+ by CD-adapco.

The investigation now clearly showed that openFoam® has competitive performance abilities to experiments and commercial CFD software. When no expensive openFoam® training is paid for the time it takes to gather experience in openFoam® handling is long, but can be shortened immensely by the usage of formerly prepared tutorials. In ship design the most important simulations do not change very often, so that this structure can be a help for the yard. But the unusual structure and the unsatisfying product documentation set the limits quite high for the time restricted ship building business to risk the use of the academically developed tool.

Essentially, all models are wrong, but some models are useful
--*George E. P. Box (Professor Emeritus of Statistics)*

Acknowledgments

First of all I would like to acknowledge the Friedrich Lürssen Yard for supporting this thesis project and everyone at the yard for a very good working environment.

I thank especially my supervisor at the yard Dr. Robert Stück, who was a great help and a lively input every time I had questions. Under the aspect of spending much time beside his own work, makes his effort even more valuable. I hope that he can use what I have prepared.

Prof. Rickard Bensow supervised the thesis from the Chalmers University side with great interest and helped define the thesis profile when help was needed. Thank you very much for an open ear and stimulating discussions during the last six month.

Also I want to thank the whole ship theory department of the yard for all the hints and tips during the thesis, for sharing your immense naval architecture wisdom and for being such nice colleagues. These were in detail Matthias Söhnlein, Lester Schelhorn, Konny Bartels and last but not least Felix Tietgen, without whom I would have had problems arriving at the yard at all most of the days.

Sascha Grube invested much of his time helping with the CAD models and showed great skills in design and visualization of all kinds of computations. Thank you also for letting me use your own yacht design as a test base.

In the end I would like to thank my family and friends for being a great support at all times, especially the love of my life Kerstin Ewert.

Table of Contents

Table of Figures

List of Tables

Table of Appendix Figures

Abbreviations

BL	– Boundary Layer
C_D	– Drag Coefficient
CDS	– Central Differencing Scheme
CFD	– Computational Fluid Dynamics
C_L	– Lift Coefficient
CV	– Control Volume
DoF	– Degrees of Freedom
DNS	– Direct Numerical Simulation
FN	– Froude Number
FVM	– Finite Volume Method
GUI	– Graphical User Interface
GCI	– Grid Convergence Index
HSVA	– Hamburgische Schiffbau- Versuchsanstalt (Hamburg Ship Model Basin)
HVAC	– Heating / Ventilation / Air Conditioning
IT	– Information Technology
Kts	– Knots
LES	– Large Eddie Simulation
L_{OA}	– Length Overall
MPI	– Message Passing Interface
MUSCL	– Monotone Upstream-centered Schemes for Conservation Laws
NSE	– Navier-Stokes Equations
PLIC	– Piecewise-Linear Interface Calculation
RANS	– Reynolds-Averaged Navier-Stokes
OSS	– Open Source Software
Re	– Reynolds Number
RSM	– Reynolds Stress Model
sHM	– snappyHexMesh
SIMPLE	– Semi-Implicit Method for Pressure-Linked Equations
UDS	– Upwind Differencing Scheme
V&V	– Verification and Validation
VOF	– Volume of Fluid

Symbols & Definitions

ε	Turbulent Dissipation
κ	Von Kármán Constant
μ, mu	Dynamic Viscosity [Pa*s]
ν, nu	Kinematic Viscosity [m^2/s]
ρ	Density [kg/m^3]
ω	Specific Dissipation [m^2/s^3]
Γ	Diffusion Coefficient
ϕ	Velocity Potential
ϕ_v	Convected Variable
k	Turbulent Kinetic Energy [m^2/s^2]
p	Pressure [Pa]
t	Time [s]
T	Temperature [K]
U	Flow Velocity [m/s]
V	Turbulent Velocity [m/s]
$\mathbb{v}$	Velocity Vector [m/s]
$\mathbb{v}'$	Fluctuating Flow Velocity Vector [m/s]
$\mathbb{V}$	Mean Velocity Vector [m/s]
u, v, w	Flow Velocities in x, y, z [m/s]
u', v', w'	Fluctuating Flow Velocities in x, y, z [m/s]
U, V, W	Mean Flow Velocities in x, y, z [m/s]
x, y, z	Cartesian Coordinates

1 Introduction

Identifying flow directions, velocities and forces is a very important task in most of the many engineering industries. To summon information about the flow around a ship in particular is interesting for marine engineers around the world for a long time now. This is typically done in wind tunnels and towing tanks or collected from long time gathered experience. Since for most complex flow situations an analytic solution is not available, rule-of-thumbs are not detailed enough and experiments are expensive, computer generated solutions become more and more popular with increasing computational power and better mathematical models. This field of research is called Computational Fluid Dynamics (CFD) and was the general topic of this thesis. The current chapter will show now where exactly the thesis is located in this field of study (Chapter 1.2) and why it was performed (Chapter 1.1). In chapter 1.3 the structure and the approach of the thesis is presented.

1.1 Aim of the thesis work

The Master Thesis at hand aims at the development of several different CFD standard models for the Fr. Lürssen shipyard in Bremen, Germany – a builder of custom made luxury yachts with a L_{OA} of more than 60m - using the open source code openFoam® (Version 1.7.1 on Ubuntu 10.04) and testing the applicability of open source software on ship yards. The models should be able to calculate the exhaust gas flow above and underneath the design waterline around any new developed Lürssen yacht. It was meant to be an input for both the mechanical engineering and the ship theory department of the yard.

1.2 Project background

In the last century sophisticated CFD methods changed from being only a scientific to an industry established tool. They do not provide a completely correct solution yet, but can give comparable results for an experienced user in shorter time and with lower costs than it is done by experiments [1]. While hydro- and aerodynamic resistance predictions are most important for most ship yards, CFD methods can also be used to investigate other flow situations. For cruise vessels and large motor yachts knowledge of the distribution of exhaust gas on deck and under the waterline is crucial to satisfy the demanding customer needs. Similar to naval vessels, the flow around helipads is also important to comprehend in order to guarantee the safety of helicopter operations.

Analyzing the flow with wind tunnel and towing tank tests is costly and thereby only used in later more definite design stages. To be able to study and vary the design of exhaust outlets CFD methods can be used also already in early stages. Research in this area was done before by Kulkarni [2] or Moctar [3]. OpenFoam® itself starts to be of interest for the ship building industry since it is free and highly developed. One cause for its low distribution at the moment might be its pure academic background and the unusual handling of openFoam®. A reliable graphical user interface (GUI) is not available today.

To satisfy the different needs of the two main computations the thesis was split in two parts - an aerodynamic and a hydrodynamic part. The models for the aerodynamic calculations were two new Lürssen yachts with a L_{OA} of more than 100m. Wind tunnel experiments performed at the HSVA were used to validate the calculation setup. Since the yachts are not built yet they were called "MYacht" and "AYacht" in this report. Furthermore all picture material showing geometry and important dimensions of the vessels had to be distorted in some way or another. For the hydrodynamic part an older yacht was used that was named "CYacht" in this thesis. She can be found in the same length area as the other two yachts and was especially interesting due to her distinct underwater exhaust outlets – the research object of that part.

1.3 Approach and structure of the thesis

The time that it takes to prepare and run as many CFD calculations as it was necessary for such a project with standard workstations, made it vital to structure and organize the work in the correct way. It was desired to overlap the times of actual computing with the preparation of the next problem as much as possible. The thesis follows the actual order of work steps quite closely and shows thereby how the problem was approached. After an extensive research on today's methods and possibilities a general approach was identified. To give background information on the chosen approach the required theory is presented in chapter 2 which contains the governing equations 2.1 and a few words on the computation of turbulent and multiphase flows 2.2 .

A CFD calculation is only as good as its setup and needs ongoing assessment of the outputs along the way. Several coarse calculation loops are necessary in the beginning to gather information about the prerequisites of the problem at hand. As an output from these loops the calculation setup of both the aerodynamic and the hydrodynamic calculations is presented in chapter 3. To verify the grid quality a grid convergence study was performed for the grid that was used in the aerodynamic calculation and the performance of the used methods was evaluated (see chapter 4). From this point iterations with smaller changes were done which resulted in a setup that was validated and enhanced further until the outcome showed good resemblance with the experiments. In particular the aerodynamic computations were checked against a wind tunnel experiment and the hydrodynamic calculation was compared to commercial software. The validated and completed results are shown in chapter 5 and finally discussed in chapter 6. An outlook on the future work is given in chapter 7. The different parts of the thesis are also visualized in Figure 1.1.

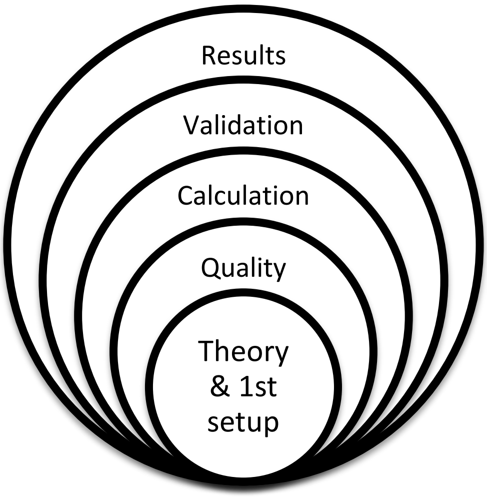


Figure 1.1 - Structure of the work and the thesis

2 Theory

The dynamic behavior of any flow situation can be described by several general laws in physics. The first section 2.1 presents the most important equations that were derived from those laws and also how the equations can be simplified to ease the solving process. Solving these equations analytically is difficult to achieve for most engineering applications as has been said previously. This is why CFD methods are used which approach the problem numerically. The important methods and some additional theory are now shown in the second section 2.2.

For the interested reader who requires a broader view of the topic on CFD at this point please refer to the standard text books of Ferziger and Peric [4], Versteeg and Malalasekra [5], Blazek [6] or similar. By far the most information in these chapters was taken from these books.

2.1 Governing equations

The earlier mentioned physical laws are the laws of conservation of mass, energy and momentum, which are described in the next two sections. These principal properties of fluids are mostly governed by the viscosity η and the density ρ and are mainly functions of the temperature T, the pressure p and the velocity U. Since flow problems are limited in space it is looked at the flow of a fluid through an arbitrary single control volume (CV) to gain knowledge about these properties.

Fluid flows can be distinguished in laminar and turbulent flows. In this thesis turbulent flows were examined that occur when U varies unordered in time and space. The turbulence magnitude is correlated to the flow velocity and viscosity magnitudes, i.e. the Reynolds Number (Re). All turbulence phenomena in the flow are basically unpredictable, but with certain simplifications (see chapter 2.1.2) one can get a good understanding of the macro scale flow behavior.

In the ship building industry high turbulence is most of the times unwanted since it increases the frictional resistance of a ship or destroys a smooth airflow around the superstructure. In the following sections it is shown that you use different computational methods for laminar and turbulent flow.

2.1.1 Continuity & Navier-Stokes equations

The law of conservation of mass states that the total net transport of the mass across the boundaries (i.e. the mass flux) of a system, that is in this case the control volume, is zero if sources are not considered. This means that no process inside the control volume can change the mass quantity, i.e. no mass can be destroyed or created. The law is represented by the so called continuity equation (Equation 2.1).

Equation 2.1

$$\frac{\partial\rho}{\partial t} + div(\rho\mathbb{V}) = 0$$

A similar law exists also for energy. The law of conservation of energy states that the total energy in a closed system (again in this case the control volume) remains unchanged over the time [7]. The corresponding relationship simply says that the sum of all the potential and all the kinetic energy in a system equals the total energy.

To get a detailed view of the flow the Navier-Stokes Equations (Equation 2.2 - Equation 2.4) (NSE) for a compressible fluid are examined. Together with the continuity equation and the energy law, they give a complete picture of the flow characteristics.

Equation 2.2

$$\boldsymbol{\left(\frac{\partial(\rho u)}{\partial t} + div(\rho \mathbb{v} u)\right) = -\frac{\partial p}{\partial \mathrm{x}} + div(\mu\, grad\, u) + S_{mx}}$$

Equation 2.3

$$\boldsymbol{\left(\frac{\partial(\rho v)}{\partial t} + div(\rho \mathbb{v} v)\right) = -\frac{\partial p}{\partial \mathrm{y}} + div(\mu\, grad\, v) + S_{my}}$$

Equation 2.4

$$\boldsymbol{\left(\frac{\partial(\rho w)}{\partial t} + div(\rho \mathbb{v} w)\right) = -\frac{\partial p}{\partial \mathrm{z}} + div(\mu\, grad\, w) + S_{mz}}$$

The NSE was developed from the momentum equation, which is an application of the Newton laws of motion for a continuum (For the detailed derivations refer to the mentioned text books). It consists of several parts of which each represents a driving force on the fluid which are:

• $\frac{\partial(\rho u)}{\partial t}$	Unsteady Acceleration (In this case for all exemplary in x-direction)
• $div(\rho \mathbb{v} u)$	Convective Acceleration
• $-\frac{\partial p}{\partial \mathrm{x}}$	Pressure Gradient
• $div(\mu\ grad\ u)$	Viscosity Forces
• S_{mx}	Other Body Forces

To simplify the equation for different flow situations, some of its parts shown above can be neglected or modeled. This saves computational time or makes the NSE analytically solvable, but also sometimes introduces errors. Typical approximations and modeling techniques can be seen in the following section.

2.1.2 Simplified equations and flow approximations

An engineer working on a flow problem will always get the best results computing the NSE directly. Since computational resources usually do not allow this, a more practicable approach is to simplify certain parts of the equation or to fully erase a part when it is not needed. For an example the smallest eddies in the flow are usually not accounted for in practical purposes, since their influence is quite small on the overall solution, but they would be included if the complete NSE is solved.

A first and very general approximation has already been made in this thesis. For the following equations and calculations incompressible flow was assumed. This means that the time derivative of ρ becomes zero. The approximation can be made for the flow media that were used in this thesis - water and low speed air (Mach number < 0.3) [1].

2.1.2.1 Boussinesq approximation

The Boussinesq approximation involves modeling of gravitation free flow in which density variations can be neglected in the convective and unsteady part of the NSE and moderate temperature gradients are allowed. Its advantage is the reduction of complexity by assuming one overall density for the different substance streams in the flow. This can be made use of e.g. in ocean current calculations or natural ventilation in homes.

2.1.2.2 Euler & Potential flow

A different approach is used by the Euler and the potential flow theory. Here not the density is manipulated but the viscosity is completely neglected. If the viscous forces in the flow are disregarded the NSE simplifies to the so called Euler equation (Equation 2.5). The equation can be used to compute compressible flows with high Mach numbers [8]. This can be done since the implied no-slip condition (i.e. No viscous forces on walls are calculated) is not a major problem at high velocities where viscous effects are only important in the small area close to the wall.

Equation 2.5

$$\frac{\partial \mathbb{v}}{\partial t} + \mathrm{grad}(\mathbb{v} * \mathbb{v}) + \frac{1}{\rho}\mathrm{grad}(p) = 0$$

If now a rotation free flow ($\nabla \times \mathbb{v} = 0$) is considered a velocity potential Φ must exist that fulfills the equation $\mathbb{v} = -\nabla\Phi$. Taking this into account the incompressible continuity equation (Equation 2.6) becomes the Laplace equation for the potential (Equation 2.7). This equation can be used to compute a free stream non turbulent flow, e.g. in the water flow field far away from the ship. The Laplace equation is analytically solvable even for more complex flows by combining simple cases (Sources, Sinks, Free Stream, etc.) and can be used as an input to solve the integrated momentum equations, the so called Bernoulli equations.

Equation 2.6

$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0$$

Equation 2.7

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} + \frac{\partial^2 \Phi}{\partial z^2} = 0$$

2.1.2.3 Boundary layer approximation

When the fluid is close to bounding surfaces the near wall area is called the boundary layer (BL). Here viscous effects cannot be neglected any longer. For low Re the flow stays laminar and can be calculated using the simplified NSE called the partial differential equation (Equation 2.8).

Equation 2.8

$$\rho(\mathbb{v} * \mathrm{grad}(\mathbb{v})) = -\mathrm{grad}(p) + \mu\nabla^2\mathbb{v}$$

In theory the fluid has no velocity directly on the surface and thereby a velocity gradient appears normal to the surface that causes frictional forces in the fluid. A further approximation is made by assuming the pressure across the BL (in direction normal to the surface) to be constant [9]. The effect on the computations of these assumptions is presented in the next section.

2.2 Computational fluid dynamics

From chapter 2.1 a set of general equations is known that can be used to solve the flow problems at hand. Several methodologies exist that can carry out this tasks. Since viscous calculations are the main topic of this thesis the most commonly used techniques in this field are described. Laminar flow methods are shortly illustrated. The most known general approach in a CFD work is to create a mesh of the geometry and the flow domain, to choose the numerical techniques, to solve the equations and then to view and control the results. These steps are called:

- Pre-processing (Including the meshing process)
- Solving
- Post-processing

The need for pre-processing is based on the fact that each flow simulation has different needs in terms of start values, grids, schemes, solvers, etc. In Figure 2.1 the different steps are shown that need user input in the pre-processing stage. Especially in the modeling and the discretization stage many methods and schemes are available - the user needs experience and understanding of CFD to choose the right ones. The information in Figure 2.1 was taken from the educational slides of T. S. Craft [10].

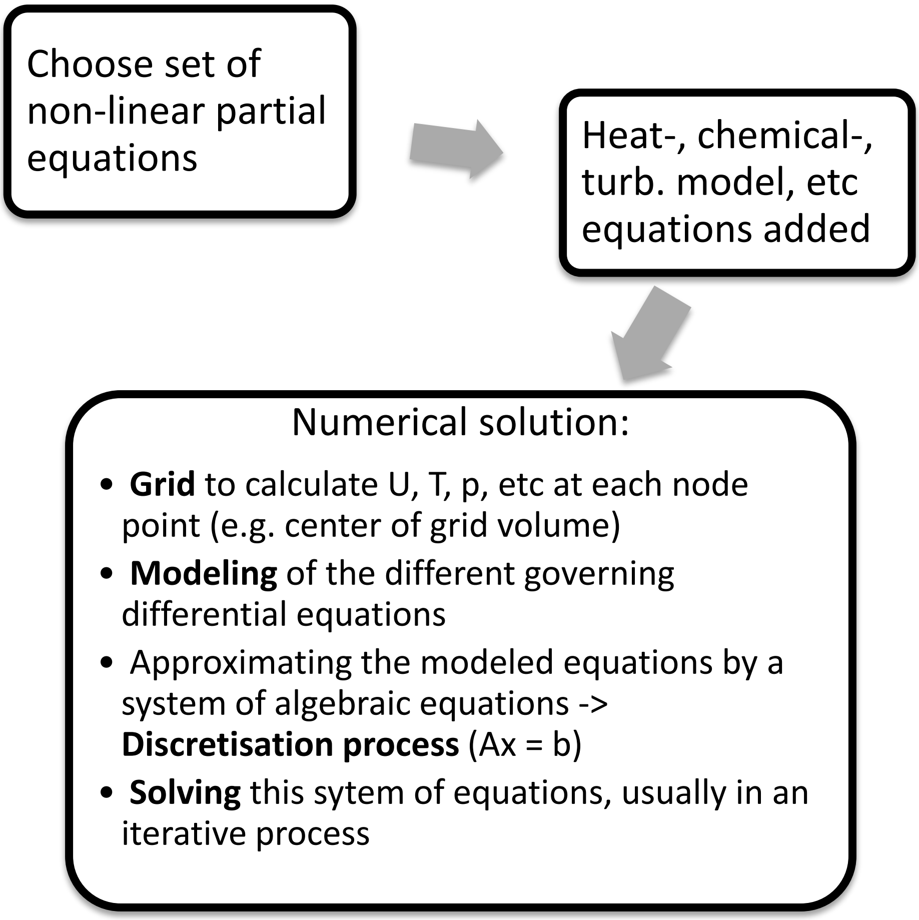

Choose set of non-linear partial equations
Heat-, chemical-, turb. model, etc equations added
Numerical solution:
• Grid to calculate U, T, p, etc at each node point (e.g. center of grid volume)
• Modeling of the different governing differential equations
• Approximating the modeled equations by a system of algebraic equations -> Discretisation process (Ax = b)
• Solving this sytem of equations, usually in an iterative process

Figure 2.1 - Important steps in a regular CFD calculation setup

2.2.1 Errors and uncertainties

No matter if the flow is laminar or turbulent CFD is always just an approximation of the reality for the reasons explained earlier. There are many steps where errors and uncertainties can occur [1]. It is reasonable to distinguish between those two since many wrong calculations occur due to lack of knowledge of the tools (uncertainty) and not only due to mathematical errors e.g. in the modeled equations or discretization process (More information on this in chapter 4) . The process of checking for errors is called Verification & Validation (V&V). Verification is an in depth comparison of computed scientific test cases with their analytical results to verify the code while validation compares your computed case results with experimental data (e.g. from a wind tunnel test) [11]. In a way validation thereby checks also for uncertainties which can then be removed by the user until ideally only the real CFD code errors remain. Extensive verification of the openFoam® code was done e.g. by H. Jasak [12], validation of the results is shown in chapter 5. To assess the quality of the mesh, and thereby of the results, a grid convergence study was performed in chapter 4 as has already been mentioned.

2.2.2 Non turbulent flow methods

One way of removing turbulence in the flow is to neglect the viscosity. If Equation 2.7 is solved with certain boundary conditions its CFD application is called the potential flow method. Its general application in the marine industry is to calculate the wave height and wave resistance of a ship (Figure 2.2). It is a very fast and simple method and used broadly in the industry, especially for a bulbous bow optimization.

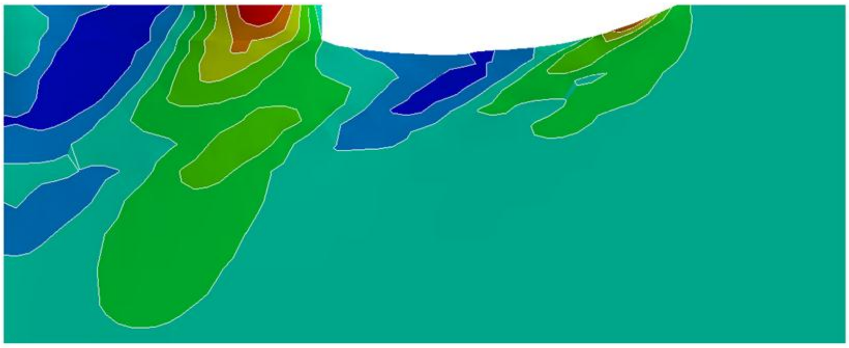


Figure 2.2 - Wave height contours of a yacht hull generated with a potential flow code [13]

The potential flow method can also be used to calculate the outer edge pressure of the BL. Using the assumption in chapter 2.1.2.3 the pressure in the whole BL is then known. CFD codes based on this assumption are called Boundary Layer Methods. They save a lot of computational time compared to turbulence modeling CFD methods (see chapter 2.2.3) where the pressures and velocities have to be matched but work only fine in flows without separation [9] [11]. Viscous resistance or lift and drag calculations cannot be solved with this method [11] which is the main reason why it is not further elaborated in this thesis. Another way of calculating without turbulence is necessary in some cases when using turbulence models (see Chapter 2.2.3.1). Many solvers diverge when starting to compute a complex flow situation directly with the turbulence switched on. You can force it to be a laminar flow until a steady state is reached. With this developed pressure and velocity field one can then start a turbulent calculation that is more likely to run through. This was necessary to do for the water flow around the ship hull.

2.2.3 Turbulent flow methods

Nowadays computational power is not enough to solve the full NSE directly for almost all flow phenomena, which engineers are interested in. It is possible for some academic study cases or very simple flow situations. The method used when solving the full NSE is called Direct Numerical Solution (DNS). The first approximation of the DNS is called Large Eddie Simulation (LES), which solves only the large scale turbulence in the flow. The smaller eddies are filtered by a modification of the NSE. The smaller eddies affect the resistance calculations in ship hydrodynamics less than larger ones and can thereby be neglected in practical use. It is possible to calculate larger structures like ships with LES codes, but is still not feasible to do in ship yards due to the computational power and time needed.

The most common solution and the methods also used in this thesis remove all turbulence and solve the NSE only for the mean flow. The methods based on the Reynolds-Averaged Navier-Stokes (RANS) Equations (Equation 2.9 - Equation 2.11) are called the RANS method. The equations show the unsteady RANS. In this thesis steady RANS calculations were performed where the time depended part equals zero.

Equation 2.9

$$\frac{\partial(U)}{\partial t}+div(U\mathbb{V})=-\frac{1}{\rho}\frac{\partial P}{\partial x}+v div\big(grad(U)\big)+\frac{1}{\rho}\left[\frac{\partial(-\rho\overline{u'^2})}{\partial x}+\frac{\partial(-\rho\overline{u'v'})}{\partial y}+\frac{\partial(-\rho\overline{u'w'})}{\partial z}\right]$$

Equation 2.10

$$\frac{\partial(V)}{\partial t}+div(V\mathbb{V})=-\frac{1}{\rho}\frac{\partial P}{\partial y}+v div\big(grad(V)\big)+\frac{1}{\rho}\left[\frac{\partial(-\rho\overline{u'v'})}{\partial x}+\frac{\partial(-\rho\overline{v'^2})}{\partial y}+\frac{\partial(-\rho\overline{v'w'})}{\partial z}\right]$$

Equation 2.11

$$\frac{\partial(W)}{\partial t}+div(W\mathbb{V})=-\frac{1}{\rho}\frac{\partial P}{\partial z}+v div\big(grad(W)\big)+\frac{1}{\rho}\left[\frac{\partial(-\rho\overline{u'w'})}{\partial x}+\frac{\partial(-\rho\overline{v'w'})}{\partial y}+\frac{\partial(-\rho\overline{w'^2})}{\partial z}\right]$$

In the RANS equations the velocity and pressure parts are split up into a mean velocity/pressure (*U/P*) and a fluctuating velocity/pressure component (*u´/p´*) that displays vortices in the flow. The fluctuating velocities can be found on the right hand side and seen as additional turbulent stresses on the mean velocity. These so called Reynolds stresses can be represented by different turbulence models shown in chapter 2.2.3.1. This is a severe simplification of the NSE but results in less required computational time and memory compared to LES or DNS methods.

Besides the need of closing the system of mean flow equations, a turbulence model should be widely applicable, accurate and simple. Since RANS codes are used nowadays throughout most industries, their economical aspect is also very important. If these basic requirements are not met, the method will not be used. This holds especially true for the ship building industry.

2.2.3.1 Turbulence modeling

The previously mentioned Reynolds stresses have to be modeled, if a RANS method is used. This can be done in various ways, which has a large effect on the later results. Most turbulence models are based on the eddy viscosity concept where the effect of the turbulence on the flow processes is described by an increased viscosity. The Reynolds stresses are then described by the so called Boussinesq Hypothesis which is represented by Equation 2.12, where k is the turbulent kinetic energy.

Equation 2.12

$$\tau_{ij} = -\rho\overline{u_i'u_j'} = \mu_t\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \frac{2}{3}k\rho\delta_{ij} \text{ with } k = \frac{1}{2}\overline{u_i'u_i'}$$

As the name already states k represents the energy in the turbulence. μ_t is the turbulent viscosity and can be written as displayed in Equation 2.13 including a constant C_μ, the turbulent velocity V and the length scale for large-scale turbulent motion L.

Equation 2.13

$$\mu_t = \rho C_\mu L\sqrt{2k} = C_\mu LV$$

To solve Equation 2.12 one needs a way to express k and μ_t to calculate V and L. The different approaches to do this divide the models into algebraic, zero-equation, one-equation and two-equation models, which are all linear eddy viscosity models. Not discussed in this thesis are non-linear eddy viscosity models and the Reynolds-stress models, which compute the Reynolds stresses directly.

Industry standard and also the models used in this thesis are the two-equation models. It is quite simple to implement these kinds of RANS equations into a CFD program. It is basically the same code as for laminar flow, but has two differences, which are:

- $\mu_{eff} = \mu + \mu_t$
- Solving of two transport equations extra

This allows effects of the transport of turbulence properties by convection and diffusion and of production and destruction of turbulence. The two most common methods and their extra transport equations are presented in the following two sections.

2.2.3.1.1 K-ε model

Based on the fact that kinetic energy is dissipated into internal energy at high Re another way of computing the turbulent length is using the turbulent dissipation ε (≈ $k^{3/2}$ / L). The transport equations for k and ε can be seen in Equation 2.14 and Equation 2.15 as well as the related turbulent viscosity Equation 2.16.

Equation 2.14

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho\overline{u_j}k)}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\mu\frac{\partial k}{\partial x_j}\right) - \frac{\partial}{\partial x_j}\left(\frac{\rho}{2}\overline{u_j'u_i'u_i'} + \overline{p'u_i'}\right) - \rho\overline{u_i'u_j'}\frac{\overline{\partial u_i'}}{\partial x_j} - \mu\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_i'}{\partial x_k}}$$

Equation 2.15

$$\frac{\partial(\rho\varepsilon)}{\partial t} + \frac{\partial(\rho u_j \varepsilon)}{\partial x_j} = C_{\varepsilon 1} P_k \frac{\varepsilon}{k} - \rho C_{\varepsilon 2} \frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j}\left(\frac{\mu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_j}\right)$$

Equation 2.16

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon}$$

The turbulent model that is based on these equations is called the k-ε model and is probably the best known method in CFD codes [4]. The constants are set by the developers of the programs and vary from software to software. Several popular developments that build up from this model are available, e.g. the realizable k-ε [14] or the RNG k-ε model [15]. Its strong non-linearity makes it necessary to solve the transport equations gradually with an iterative method (like most CFD codes). Underrelaxation is also needed for stability reasons. U, k and ε usually get the same relaxation values between 0.5 (for a bad grid) and 0.8 (for a good grid). Figure 2.3 shows a Wigley hull wave pattern calculated with the k-epsilon model.

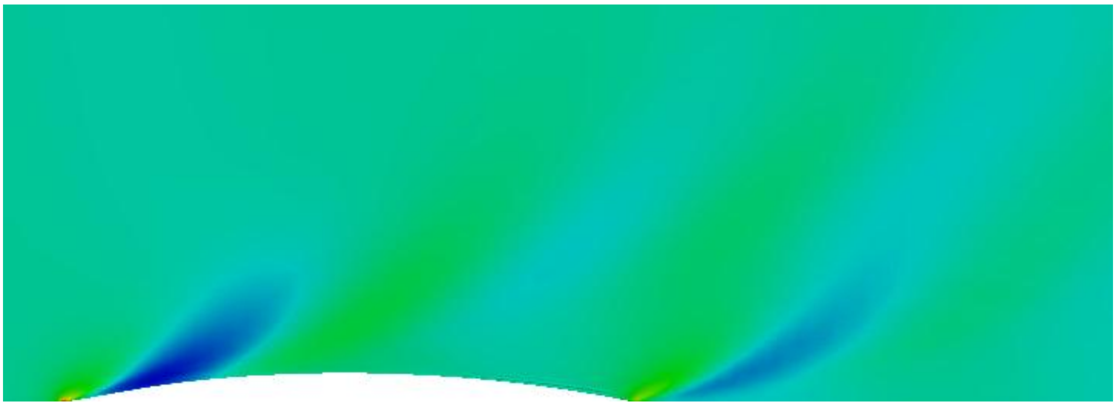


Figure 2.3 - Wigley hull at FN 0.32, calculated with interFoam, k-epsilon model

Close to walls k and ε are higher while U stays similar to the normal flow, but they use the same grid. This can lead to convergence problems especially when higher order schemes are used. At high Re the boundary layer on the wall gets so thin that it is difficult to resolve it with enough grid points. In this very common case wall functions can be applied to the model. They are based on the existence of an area close to the wall, in which the wall normal velocity profile can be described with the logarithmic law shown in Equation 2.17 - the so called law of the wall.

Equation 2.17

$$u^+ = \frac{1}{\kappa}\ln(y^+) + B$$

u^+- Dimensionless velocity
y^+- Dimensionless wall coordinate
B - Case specific constant

In general the k-ε model is limited in its applications. It supplies only good results for flows with small pressure gradients. Problems like swirling and rotating flow, strong separation, compressors or nozzles should be avoided [16].

2.2.3.1.2 K-ω model

The k-ω model follows basically the same rules as the k-ε model but incorporates the specific dissipation ω (= ε / kβ*) instead of ε to represent the scale of turbulence. The corresponding transport equations plus the equation for the turbulent viscosity can be seen in the Equation 2.18 - Equation 2.20. It was introduced by Wilcox [17] in 1998.

Equation 2.18

$$\frac{\partial(\rho k)}{\partial t}+\frac{\partial(\rho\overline{u_j}k)}{\partial x_j}=P_k-\rho\beta^{*}k\omega+\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_k^{*}}\right)\frac{\partial\omega}{\partial x_j}\right]$$

Equation 2.19

$$\frac{\partial(\rho\omega)}{\partial t}+\frac{\partial(\rho u_j\omega)}{\partial x_j}=\alpha P_k\frac{\omega}{k}-\rho\beta\omega^2+\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_\omega^{*}}\right)\frac{\partial\omega}{\partial x_j}\right]$$

Equation 2.20

$$\mu_t = \rho\frac{k}{\omega}$$

An advantage compared to the k-ε model is its applicability in the entire boundary layer without further adjustments. A problem can occur from its high sensitivity to the inlet boundary conditions of internal flows [18]. An advancement, also because of this disadvantage, of the k-ω model was introduced by Menter [19] [20] ; the so called k-omega SST model. It is a hybrid using k-ω close to walls and k-ε in the turbulent flow. The k-omega SST model is especially used in aerodynamic calculations and has a very good overall performance even for complex flows compared to other two equation models [16].

2.2.3.2 Finite Volume Methods

There are three commonly used numerical methods to solve the conservation equations - the Finite Element Method, the Finite Difference Method and the Finite Volume Method (FVM). For the RANS equation, the Euler equation and the NSE the Finite Volume Method is the standard approach. It is easy to program and thereby most frequently used. As it is usually done in the FVM it is assumed in the following explanation that the velocity field and the fluid properties are known from the boundary conditions.

When applying the FVM the physical space, in which the flow is calculated, is discretized into many control volumes by a mesh. This mesh defines, via its grid points, the edges of the control volumes and not the calculation points like in a Finite Difference Method. The integral conservation equations need to be calculated now for each volume individually and can thereby be computed for the whole domain. A generic form of the conservation equation can be seen in Equation 2.21.

Equation 2.21

$$\frac{\partial(\rho\phi_v)}{\partial t}+\frac{\partial}{\partial \mathrm{x_i}}\left(\rho U\phi_v-\Gamma_{\phi_v}\frac{\partial\phi_v}{\partial \mathrm{x_i}}\right)=S_{\phi_v}$$

A generic CV is shown in 1D in Figure 2.4 and in 3D in Figure 2.5. The flux over the boundaries of the CV equals the sum of the integrals. The equation variables (denoted ϕ_v) are calculated in the cell center whereas the control volume surface is interpolated via its node values. The resulting surface and volume integrals can be approximated with numerical methods which are displayed in the next section 2.2.3.3. The node value in the cell center is denoted P and the neighboring nodes according to the flow direction.

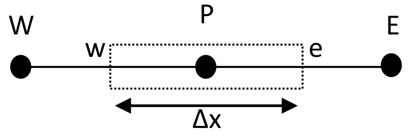


Figure 2.4 - 1D Control volume with nodal points

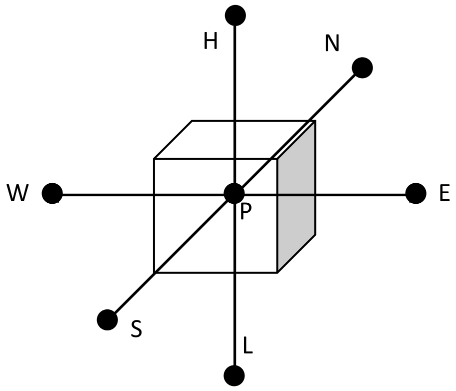


Figure 2.5 - Control volume in 3D with face direction

How good a discretization works, depends to a large extend on the scheme with which the fluxes are evaluated. Also the design and look of the volume is important. A single control volume should be as regular in shape as possible to reduce numerical errors. What 'regular' means in this case is explained further in the chapter 2.2.4. Holes and overlapping or negative volumes in the grid result in wrong calculations or program failures as well.

2.2.3.3 Numerical schemes

Interpolation is done when values of ϕ are needed in other points than the CV center to calculate the convective and diffusive terms and to compute the unknown volume integrals. Depending on what information is discretized, different methods have to be used. Some methods are known for their robust and stable behavior, others are particularly accurate. In this part a few important numerical schemes are shown, which are generally used to interpolate the convective terms. Its special behavior makes convection a crucial part in the discretization process and is the cause for many schemes that were developed for it. It is done exemplary for the east end of the CV.

2.2.3.3.1 Upwind Differencing Schemes (UDS)

The upwind interpolation approximates the cell face value ϕ_e with the ϕ of the cell-center value upstream of the CV. The dependency of the value on the flow direction is shown in Equation 2.22. It is unconditionally bounded and very stable (1st order), but may cause severe numerical diffusion, if the flow direction is skewed relative to the grid. When a finer grid is used, upwind interpolation will produce a much better solution.

Equation 2.22

$$\phi_e = \begin{cases} \phi_P \text{ if } (v * n)_e > 0 \\ \phi_E \text{ if } (v * n)_e < 0 \end{cases}$$

2.2.3.3.2 Central Differencing Scheme (CDS)

A simple and often used 2nd order scheme is the CDS or also called 'linear interpolation'. It approximates the middle point values of the CV sides by the linear approximation between the two neighboring nodes. The Equation 2.23 holds for the node values in the point 'e'.

Equation 2.23

$$\phi_e = \phi_E * \lambda_e + \phi_{PE} * (1 - \lambda_e)$$

The linear interpolation factor is here defined as $\lambda_e = \frac{x_e - x_P}{x_E - x_P}$.

Second order schemes are more precise than first order schemes, but not as stable. Since the CDS is unbounded, non-physical behavior can arise in areas of strong convection. Thereby sometimes schemes are used which combine the best of the two, e. g. the Hybrid Differencing Scheme. Here the two schemes are blended in critical areas into each other to achieve convergence.

2.2.3.3.3 High Resolution Schemes

High resolution schemes can be distinguished in linear and non-linear schemes (regarding their defining function φ(r)) all represented by the flux limiter formulation (Equation 2.24 [21]). A few common functions are displayed below for the given CV.

Equation 2.24

$$\phi_e = \phi_E + 0.5 * \varphi(r) * (\phi_E - \phi_P)$$

$\varphi(r) = 0$	1st order upwind	
$\varphi(r) = r$	2nd order differencing	With $r = \frac{x_e - x_P}{x_E - x_P}$
$\varphi(r) = (r + \lvert r \rvert)/(r + 1)$	Non-linear van Leer	
$\varphi(r) = \max(0,\ \min(2r,\ 0.5r + 0.5, 2))$	Non-linear MUSCL	

2.2.3.4 *Solving the equations*

Many algorithms have been developed that are able to solve the discretized NSE sets produced by the numerical schemes. Which solver type is used depends always on the given flow problem that is usually solved in an iterative process. The handling of non-linearity and the coupling of the pressure and the velocity are the main aspects of a NSE solver. To show the function of a solver the loop of the famous Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm is displayed in Figure 2.6. SIMPLE is a commonly used CFD solver for incompressible flow developed by Patankar [22] and was deployed extensively in this thesis.

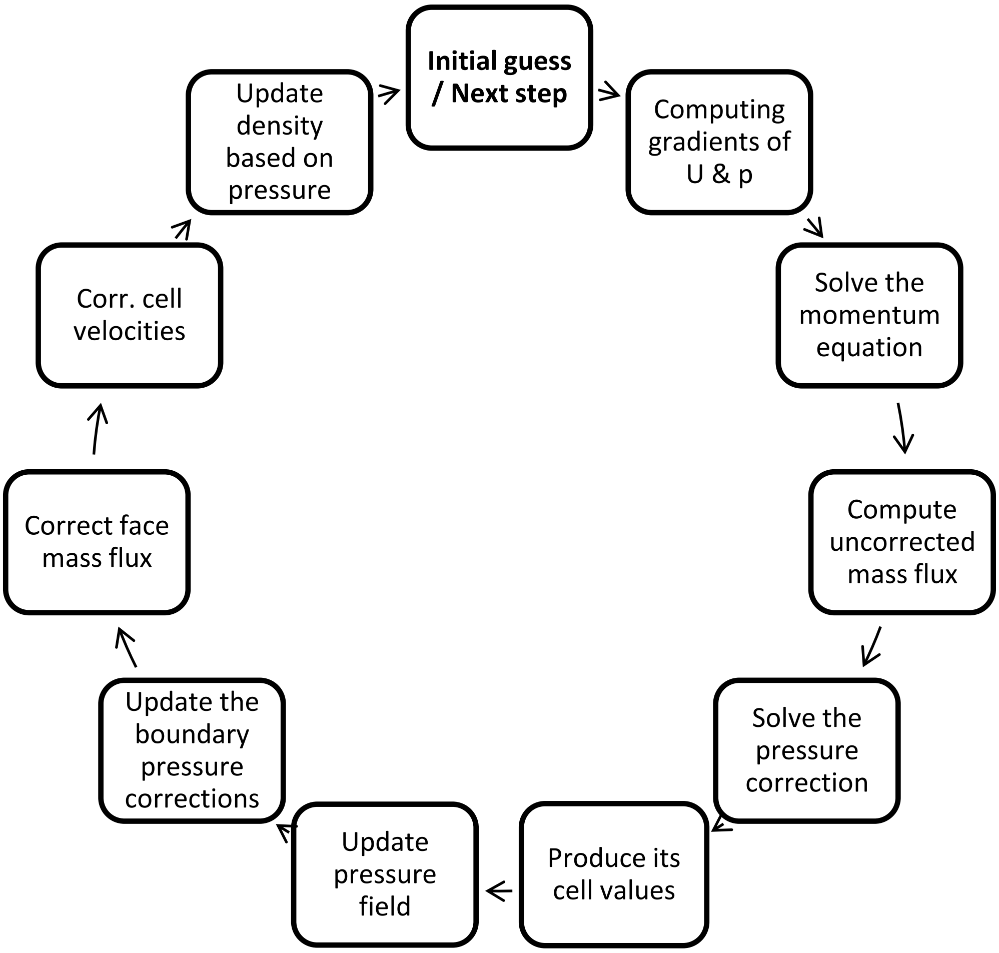

Initial guess / Next step
Computing gradients of U & p
Solve the momentum equation
Compute uncorrected mass flux
Solve the pressure correction
Produce its cell values
Update pressure field
Update the boundary pressure corrections
Correct face mass flux
Corr. cell velocities
Update density based on pressure

Figure 2.6 - The SIMPLE loop [22]

For an iterative steady state solution it is not needed to resolve the linear pressure velocity coupling. The velocity field is approximated by solving the momentum equation while the pressure gradient results from a previous pressure distribution. The velocities can be corrected after the new pressure distribution is calculated via the pressure equations. With these updated flow parameters the new resulting fluxes can be computed [22].

Other methods are SIMPLEC (SIMPLE Consistent), SIMPLEM (SIMPLE Modified), SIMPLER (SIMPLE Revised), PISO (Pressure Implicit with Split Operator) or PRIME (Pressure Implicit Momentum Explicit) - just to name a few. In chapter 3.2 and 3.3 openFoam® solvers and their properties are explained a bit further.

2.2.3.5 Volume of fluid method

The hydrodynamic calculations were performed using the industry established volume of fluid (VOF) approach. It is an Eulerian method to locate the interface between two different phases, usually water and air. This so called advection scheme tracks the form and location of the free-surface (the fluid/fluid interface), which can be seen in Equation 2.5 for a floating cube. Here the mesh needs to be dynamic in order to adapt to the floating geometry.

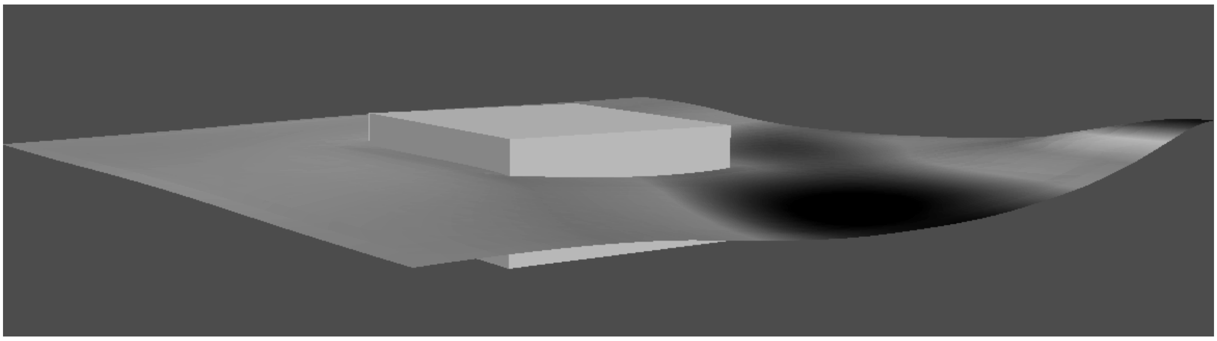


Figure 2.7 - Floating cube example of fluid/fluid interface with VOF

The VOF method is based on the function C. Looking at a control volume, C is defined as the volume fraction of a continuous phase 1, like water, in that CV. When the cell is empty of phase 1, C equals 0 and phase 2, like air, is filling the cell to 100%. The other conditions are treated accordingly. If the water/air interface cuts the cell C is 0 < C < 1 and when it is full of phase 1, C equals 1.

The free-surface is computed separate from the NSE. The code needs to solve the transport Equation 2.25 and also compute the fractions of the density and the viscosity.

Equation 2.25

$$\frac{\partial C}{\partial t} + \frac{\partial U_i C}{\partial x_i} = 0$$

How the phases are distributed in a control volume can be very different and depends on the method used, e.g. Piecewise-Linear Interface Calculation (PLIC), Hirt&Nicols [23], Youngs [24], Compression Scheme (for openFoam® [25]), etc.

2.2.4 Grid types

As mentioned earlier the points to calculate the different properties in the finite volume or finite element methods are defined by a grid ; a discretized version of the solution area. Producing a high quality grid can erase many errors before any calculation has started. Much time was spent in this thesis to match the mesh to the task and various recommendations were given to the yard how to produce a quick and good working mesh with openFoam®. The most important grid types and their functionalities are explained below.

2.2.4.1 Aspects of grid quality

A large error from the grid comes of course from its inadequate display of reality. But also the structure of the grid itself can introduce errors that have an effect on the solution quality. The most important errors are described here.

Orthogonality is a measurement of how close the angles of neighboring CV faces or CV edges are to the optimal values, i.e. of 90° for quadrilateral elements or 60° for triangular elements. Strong non-orthogonality will increase the sources and the amplitude of discretization errors in the solution as well as it leads to poorer convergence.

Expansion relates to the change of size between two adjacent cells. Small size gradients between the control volumes are preferred to ensure a controlled calculation of the field variables. Otherwise it increases the error source coming from the discretization of transient and body force terms.

Aspect Ratio is the degree of stretch of a control volume. It is usually defined as the relationship of the largest distance between the cell and the face center to the minimum distance between two edge nodes of the control volume [26]. It can produce round-off errors and leads usually to convergence problems during the solution.

Skewness is the disagreement between the shape of a cell in the grid and the shape of an equilateral cell that has the same volume as the first one. A result of high skewness can be a decrease in accuracy and a destabilized solution.

In openFoam® these properties can be tested after the grid generation with the checkMesh utility. There are mathematical definitions and recommended values for each of these incorporated in the program. In this thesis not every mesh was perfect in the checkMesh sense, e.g. high skewness was a problem for meshes generated with sHM, but was always kept in reasonable limits.

Another factor here is that the occurrence of an error does not lead per se to a problem; its location in the mesh also plays a role. For an example in areas of large pressure gradients the solver will possibly produce a diverging solution while in areas of small gradients the same grid error might still lead to an overall converging solution. This phenomenon was seen often during the thesis, e.g. in single highly skewed cells close to the hull surface where large pressures were calculated without a physical reason [26].

2.2.4.2 Structured grids

One way of dividing the domain is the structured grid. It is composed of sets of intersecting lines, one set for every dimension, and a mesh point that is placed at the intersection of only one line for each set. This composition leads to quadrilateral structures in 2D and hexahedral structures in 3D. A point has four neighbors in 2D and six neighbors in 3D. It is very efficient from a CFD point of view in terms of computational time, accuracy and memory requirements compared to unstructured grids [27]. One big disadvantage though is that this method takes much time to adapt to complex geometries and that the grid cell size cannot be varied much throughout the whole grid. For an example, when refining an area close to a wall, the cell size of the far away structures is reduced as well.

2.2.4.3 Block structured grids

This often used approach deals with the need for different cell sizes in different areas of the domain. A two or more part splitting of the computational domain is possible by defining differently structured regions that are connected by boundary faces. It is also possible to overlap the region borders. These grids are called chimera grids. In openFoam® block structured meshes are produced with the blockMesh function.

2.2.4.4 Unstructured grids

When dealing with more complex geometries structured grids offer only average usability. A widely used technique to handle these geometries is the unstructured grid. Here no implicit structure of co-ordinate lines is imposed by the grid. They can handle basically all geometries and have no limitations from the neighboring elements/points. For 2D applications the faces are usually triangular or rectangular, for 3D applications tetrahedrons or hexahedrons. OpenFoam® uses the snappyHexMesh tool for this taking a block structured grid as a domain input. A snappyHexMesh grid can be seen in Figure 3.2 where a block including a more complex half sphere was meshed.

The simple and fast meshing and refinement of local areas is an advantage compared to the structured grids. A program designed for these grids needs no changes when a refinement is done which enlarges the flexibility of work a lot. Orthogonal behavior of the cells can be forced to a certain degree as well which creates a better solution. To keep the discretization advantages of the structured grids it is also possible to apply unstructured grids directly on the wall and follow them by several layers of structured cells normal to the geometry.

Since prisms, pyramids and tetrahedrons are special forms of a hexahedron, meshes can have control volumes with less than six sides. This very flexible approach is than called a hybrid mesh.

2.2.4.5 Grid generation

The first step in generating a grid is to create a list of the knots with their x, y and z co-ordinates. Each CV is defined by e.g. 4 or 8 knots. With this information a list of CVs with their indices and the related knots is established, where the order of the points decides the position of the current face, e.g. the first four points define the downward face of a hexahedron. As a third step the list with the information of the neighbors of each CV is stored as well as a list of the faces of each control volume. The sides facing boundaries like walls, input, etc. are listed separately in a last step as displayed in Figure 2.8.

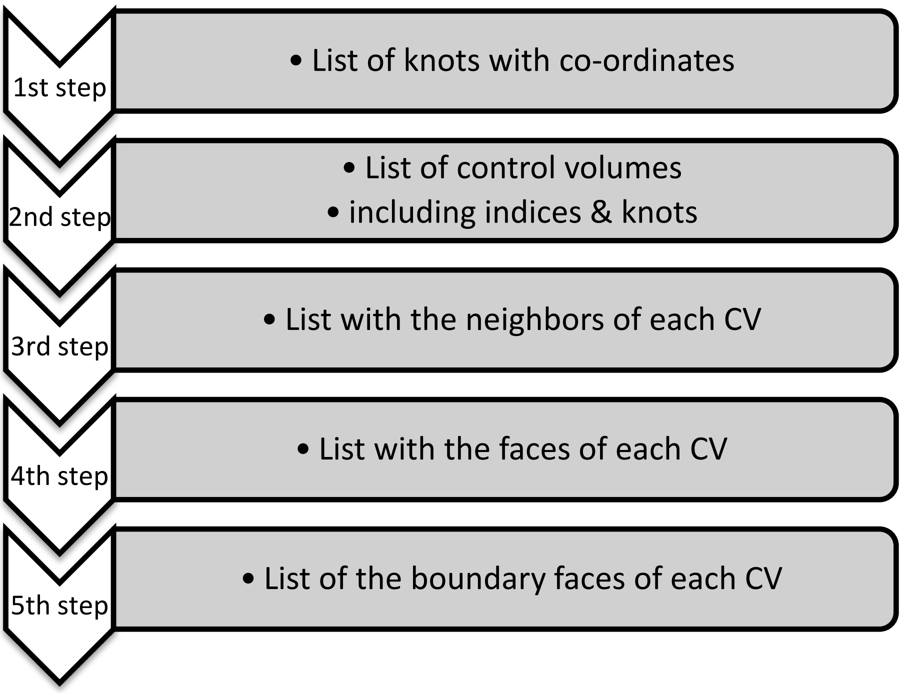

1st step
• List of knots with co-ordinates
2nd step
• List of control volumes
• including indices & knots
3rd step
• List with the neighbors of each CV
4th step
• List with the faces of each CV
5th step
• List of the boundary faces of each CV

Figure 2.8 - Grid generation process

Calculations in an unstructured grid are done in several loops. First all faces that belong to two different control volumes and then all the boundary faces are calculated to get the surfaces values. After this a loop through every CV is done to get the volume integral values.

3 Calculation setup

OpenFoam® is a program which demands experience to work with. Since you do not have a graphical user interface (GUI), but command the program through text files, its spreading in the industry is not as large as in academic use. Starting from the yards own knowledge base in openFoam® guidelines were produced by the author to speed up the progress of integrating the program into the yards workflow. Since there is also no online support available, the best procedure is to use given tutorials and prepare them for your specific problem. Several of those tutorials or case models were produced during this thesis mirroring typical flow situations that need to be investigated at the yard. The two most important models, the funnel investigation and the hydrodynamic calculation, are portrayed and further explored in the following sections.

The other models were a "hotRoom" - tutorial that allows the yard to analyze the radiation of hot components. The Figure 3.1 shows an example calculation for the mechanical engineering department to investigate the temperature distribution around a large cylinder like geometry. The vertical line to the right represents the distance to a close by component. Also a "sinkage and trim" - tutorial that supports the analysis of floating characteristics with dynamic mesh handling was developed, but not fully tested. The same holds for a utility to include particles into established flow fields.

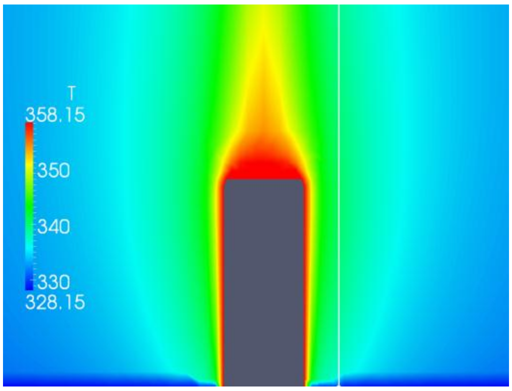


Figure 3.1 - Heat distribution around a hot cylinder

OpenFoam® itself offers also a selection of tutorials, but they are seldom directly related to ship design.

The best setup for the funnel investigation was achieved by tests on "MYacht" (Chapter 3.2). The investigation of MYacht was as important as producing the model cases for the yard and thereby extensively revisited and validated. Several setup loops were necessary to guarantee the usability and the validity of the case models for this and later projects on the yard. To test the models in a real design situation the performance was validated a second time against the wind tunnel results of the early-stage design "AYacht".

The research interest for the yard was the exhaust distribution close to HVAC (heating/ventilation/air conditioning) inlets, the magnitude of turbulence around the funnel structure or the influence of on-deck appendages on the flow for different funnel configurations (See chapter 3.2.1).

The hydrodynamic model "CYacht" was used for the underwater investigations (Chapter 3.3). The yard focused on viscous resistance, the pressure distribution on the hull for the backpressure of the exhaust outlet and the wake velocities.

Also a help booklet to explain the preparation of the model cases for other yachts/geometries was given to the yard.

3.1 Meshing setup

In this chapter as a first step the openFoam® function called snappyHexMesh (sHM), with which the domains were produced, is presented and its peculiarities are further explained. As a second step the two most important domains used for this thesis are described.

3.1.1 SnappyHexMesh

During the thesis many meshes and mesh setups were produced and compared to each other, thanks to the quick and simple usage of snappyHexMesh. On the other hand this was sometimes necessary, since sHM is not as simple to configure as it is to use. Small changes in the settings can make the difference between a diverging and a converging solution.

Starting with a block structured mesh that defines the extensions of the domain sHM includes automatically any given .stl file (usually used for stereo lithography). The solver then starts to split all cells close to the .stl surface. When the refinement is finished, all cells inside the .stl are removed. The user defines the outside with a Cartesian coordinate beforehand. Specific refinement regions (boxes, cylinders or spheres) can be assigned. It is a very good tool to emphasize specific parts of the domain, like the wake for the hydrodynamic or the funnel for the aerodynamic calculation. This needs to be and was done carefully for each calculation. If a box border is located in areas of larger pressure/velocity gradients, errors can be introduced from the change of the cell size. How detailed a mesh (and thereby the represented flow) becomes depends to a large extend on the refinement level that was chosen for the boxes and the .stl geometry [28]. A good example for a meshed .stl in a blockMesh with refinement in vertical direction can be seen in Figure 3.2.

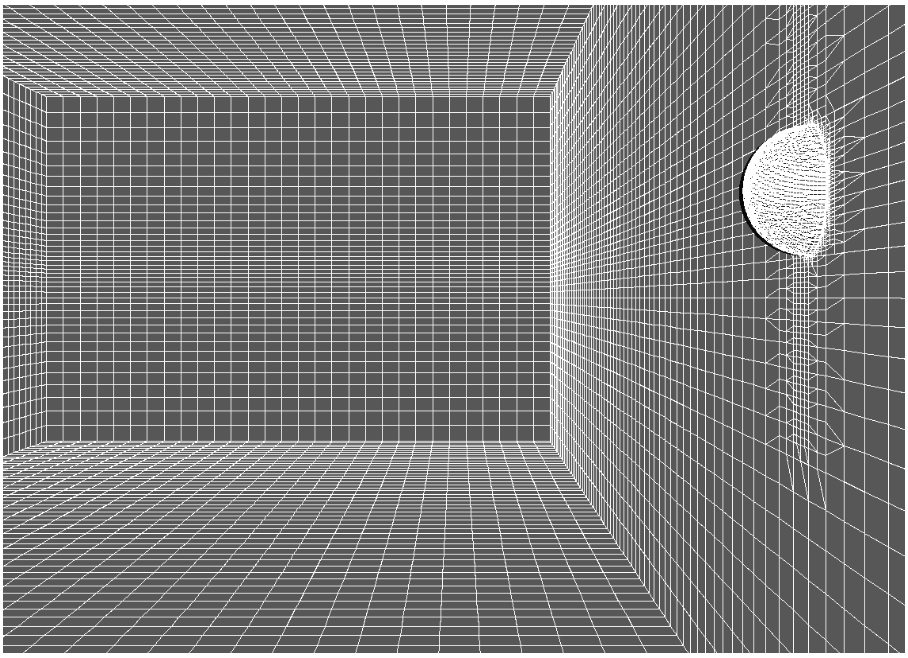


Figure 3.2 - blockMesh with meshed .stl file

A former study at the Fr. Lürssen Yard regarding the parameters of sHM [29] recommended a refinement level of (n n+1) for complex structures, where the 2nd number specifies the refinement at the wall and the 1st the level further away. During the thesis it could be shown that a level smaller than 5 and a denser block mesh is always preferable to a high refinement level and a coarse background mesh. A reason for the better converging solutions can be the more evenly distributed grid sizes in the overall domain.

Apart from the grid refinement sHM incorporates many parameters to ensure a good quality mesh. Using an automatic tool like sHM speeds up the meshing process to a matter of minutes rather than hours, but it also entails risks. Especially high skewness can occur for complex grids. Performed studies proved that this can be massively reduced by a cube-like block mesh grid. Also the reduction of the parameters "maxBoundarySkewness" and "maxInternalSkewness" have positive effect on the quality of the mesh but the user should keep in mind that decreasing the values below 5 and 2 respectively can diminish the mesh details as well. The meshQualityControls part contains further parameters which govern the mesh quality but skewness is usually the most severe problem for complex geometries. Other CFD methods (like starCCM+) offer the possibility to handle corrupt cells in a special way, e. g. enable especially robust schemes just in these cells. With openFoam® the setSet function helps to manipulate cells but so far only to delete them. Here might be a chance for further development of the code.

Another unsolved problem with sHM is its inability to always produce sharp edges where they are needed. Often an edge is displayed chamfered even though it should be right-angled. The snapEdge tool suggested by Niklas Nordin promised a cure [30]. After testing it for several ship geometries it was clear that the tool could not handle this complexity and the missing sharp edges were considered of minor importance for the fine meshes handled in the present cases.

Another important function is the layer control. sHM enables the user to specify a layer amount and thickness of structured grid blocks on the wall to resolve the BL in a better way. The wall functions need a high quality mesh so that this function is important for a good ε and ω convergence.

The laminar underwater flat plate boundary layer thickness of CYacht was roughly calculated with Equation 3.1.

Equation 3.1

$$\delta(0.99) = \frac{5x}{\sqrt{Re_x}} \approx 0.025m$$

The grid generator allows now the user to define the expansion factor, the final layer thickness and the minimum thickness. With the relationship

firstLayerHeight = Background_mesh_cellheight*finalLayerRatio/(expansionRatio)^surfaceLayers

one can calculate the cell height of the first cell directly on the wall in openFoam®. Five layers were applied for the hydrodynamic calculations. The aerodynamic boundary layer is usually thicker than the hydrodynamic one so that a coarser mesh could be utilized.

For the two following domains the snappyHexMeshDict was prepared in a way that only very small adjustments are needed to set up a similar yacht experiment with low mesh errors and the refinement boxes already in the right areas.

3.1.2 Calculation Domains

Two main grid domains were produced during the thesis after testing many different setups. One was a single block domain that cuts the ship on the free surface for the airflow (model A) and the other a two block domain for the hydrodynamic calculations cut by the X, Z symmetry plane (model B). The ship models were imported from rhino3D. Model A was a very complex version of MYacht, and AYacht respectively, and had all details around the funnel in high resolution where the flow situation was studied in particular. Model B was a simpler version of CYacht, but incorporated the real underwater body and an underwater exhaust outlet. Both models had the flowing properties:

- The [0, 0, 0] point is located at the aft perpendicular at keel level
- The ship´s bow points in positive x-direction
- Portside is in positive y-direction
- All values were calculated for full scale models

Regarding the aerodynamic calculation, the whole part of the ship above the waterline was modeled, because different angles of attack were under investigation. The model was cut at the design waterline like it is normal for wind tunnel mega yacht models. The chosen block dimensions were recommendations from experience at the yard related to the ship´s dimensions:

- Min. x-value: -300m; max. x-value: 300m
- Min. y-value: -73.5m; max. y-value: 73.5m
- Min. z-value: ~5m; max. z-value: 73.5m

A performed grid quality study (See chapter 4.2) resulted in meshes with roughly 4 million cells with a blockMesh cell distribution of [70 18 18], which is close to an equal edge length in x and y direction and an edge length with a factor of ½ in z-direction (See Figure 3.3). As earlier mentioned the cube-like or half cube-like cell lead to lower skewness in the later hexahedral mesh with MYacht included and produced a much better convergence behavior. In any CFD calculation several starting conditions need to be set up before the calculation can start successfully (See 3.2.2 Boundary Conditions). These change for different turbulence models or different solvers, but always have in common that the pressure and the velocity need to be defined at certain places of the domain, called patches in openFoam®. In Figure 3.3 you can see which parts were patches for the airflow calculation. Exhaust outlets require a separate patch to include a different velocity or temperature at that place.

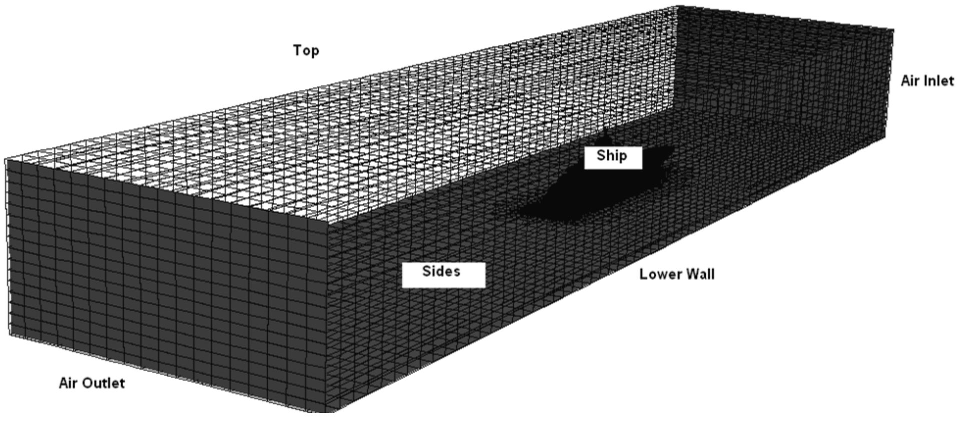


Figure 3.3 - Boundaries of the wind tunnel domain

During the hydrodynamic calculation the ship stays parallel to the flow the whole time. To save computational effort the block was thereby cut in the x, z plane and only the starboard side was investigated. The ship was cut off at seven meters height (I.e. no superstructure included) to allow a finer mesh on the underwater body with a lower quantity of cells. This typical setup is also done in most validation test cases, like the "Wigley Hull" or the Model 5415 [31]. The simple geometry leads to a mesh with no failures reported by checkMesh and overall quality and surface finish that were considered high. To take the different phase inflows into account the domain was split into an air block and a water block with separate inlet boundaries (See Figure 3.4). The water block was deep enough to be considered 'deep water' for the low velocities investigated. The models work for shallow water as well, as it has been proven in performed tests. The blockMesh properties were as follows:

• Water block -	Min. x-value: -300m;	max. x-value: 300m
• Water block -	Min. y-value: 0m;	max. y-value: -200m
• Water block -	Min. z-value: ~-45m;	max. z-value: ~5m
• Water block -	Cell distribution [150, 100, 25]	
• Air block -	Min. x-value: -300m;	max. x-value: 300m
• Air block -	Min. y-value: 0m;	max. y-value: -200m
• Air block -	Min. z-value: ~5m;	max. z-value: ~25m
• Air block -	Cell distribution [150, 100, 10]	

Another feature that can be seen well in Figure 3.4 is the denser cell area in front and behind the ship. This is not a patch but two refinement boxes which give better results in the region where a higher resolution is needed, i.e. in the wake and on the free water surface, which is the area were the function C equals 0.5 (see chapter 2.2.3.5). The top patch was not displayed for a better view into the domain. All patches defined can be seen in Figure 3.4 except for the bottom wall.

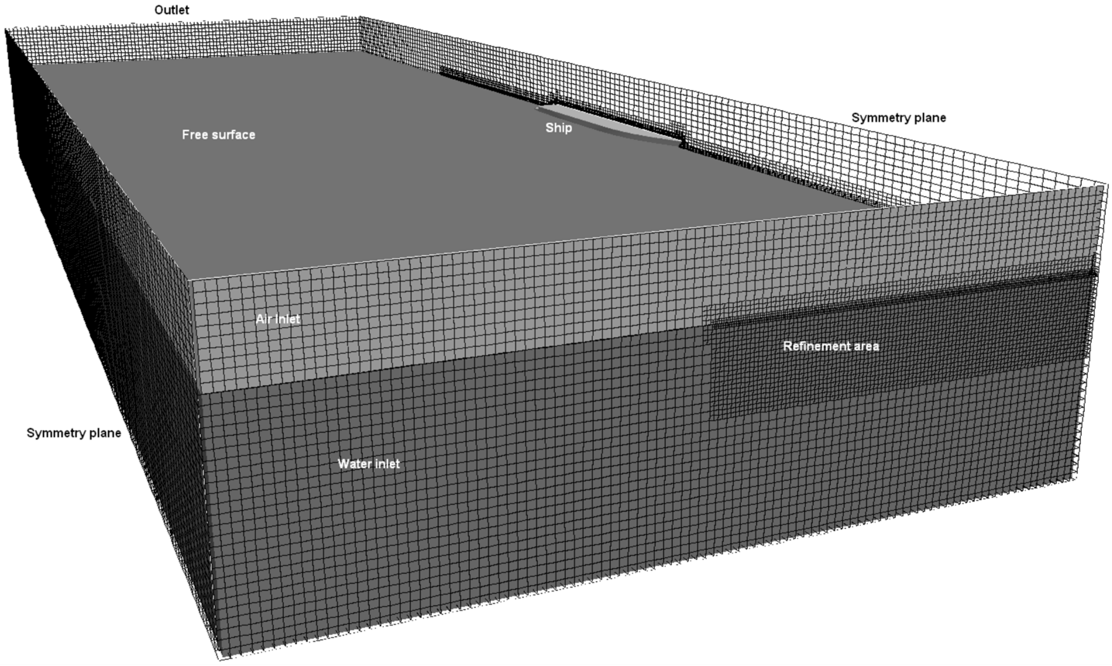


Figure 3.4 - Important areas and patches of the towing tank domain

Another important fact is that the total amount of cells used, differed between model A and B quite significantly. The simpler flow situation allowed in the aerodynamic calculation and the higher complexity of the model required a much finer mesh. Over three million cells, like used in model A, would result in a very long computation time for the hydrodynamic calculation. The cell amount in the hydrodynamic calculation (n < 1,500,000) requires already a quite low time stepping (> 0.02s). A further reduction is unpractical for the ship yard and linked to problems with the openFoam® solver.

But this is not a huge problem since the modeling error is still considered small for this calculation due to the low detail level of the hull geometry compared to the superstructure. How good details are represented always depends on their complexity and the cell size around them. An example for a typical modeling error can be seen in Figure 3.5. The same detail is modeled in two very different refinement states depending on the refinement level.

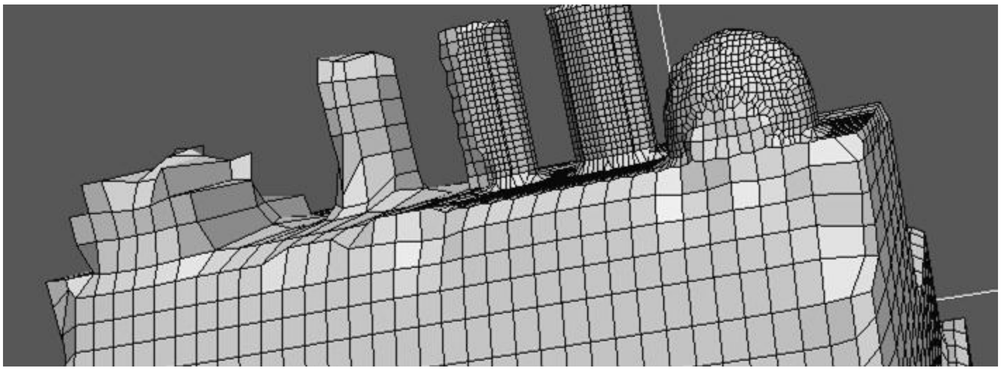


Figure 3.5 - Different levels of resolution on a hexahedral mesh

3.2 Aerodynamic calculation

Much work was spent preparing the important wind tunnel simulation during this thesis. The setup, the boundary conditions and the used solver are described in this chapter.

3.2.1 Setup

Three different models were investigated:

- Regular airflow around the hull
- Regular airflow around the hull plus airflow out of the pipes
- Regular airflow around the hull plus airflow out of the pipes plus heat

Air was assumed to have similar characteristics as exhaust. Particles can be included afterwards in openFoam® if required (See chapter 3.2.3). A variation of the pipes height and diameter resulted in four different variants of MYacht which were calculated for the second of the named models:

- #1 low pipes (ca. 1m), 0.6 - 0.7m pipe diameter, ~30m/s pipe outflow
- #2 medium pipes (ca. 2m), 0.6 - 0.7m pipe diameter, ~30m/s pipe outflow
- #3 high pipes (ca. 3m), 0.6 - 0.7m pipe diameter, ~30m/s pipe outflow
- #4 high pipes (ca. 3m), 0.4 - 0.5m pipe diameter, ~50m/s pipe outflow

The yard has done wind tunnel tests at the HSVA which were used for validation. The values were taken from this test to ensure a reasonable comparability (Pictures can be found in Appendix A). Other setup values were:

- Ship speed: 19,5 kts   True wind speed: ~10m/s
- True wind angles: 0° , 30°, 60°   Exhaust pipe angle: ~15°

Since the changing rate from zero speed to full speed in one time step can be quite large, a stepwise increase of the velocity similar to the hydrodynamic setup (See chapter 3.3) was chosen if the solution does not converge with this pre-settings right away. The same effect might have the switch off of the turbulence models until a stable pressure and velocity profile has been established.

3.2.2 Boundary Conditions

Boundary conditions can influence the outcome of a calculation to a large extent. Not only do they define the start properties of the flow, like the velocity or the temperature, but also the start values for the turbulence models. If k or omega is not initialized correctly the solver will need more time to converge or even diverge. To include the right values similar tutorials and online forums [18] were consulted. In severe cases, where a very accurate start value is needed, formulas for most parameters can be found in the named text books.

Table 1 - Main parameters and boundary conditions for the wind tunnel

	Velocity	**Pressure**	**k**	**ω**	**nut**
Inlet	fixedValue	zeroGradient	fixedValue	fixedValue	calculated
Outlet	inletOutlet	fixedValue	inletOutlet	inletOutlet	calculated
Sides	slip	slip	slip	slip	calculated
L. Wall	fixedValue	zeroGradient	kqRWallFunction	omegaWallFunc.	nutWallFunct.
Top	slip	slip	Slip	slip	calculated
Ship	fixedValue	zeroGradient	kqRWallFunction	omegaWallFunc.	nutWallFunct.

Table 1 now shows the main parameters and which boundary conditions were applied for the air and pipe flow. Similar boundary conditions need to be applied for the temperature, if the heat model is calculated. Boundary conditions in openFoam® are divided into basic, primitive and derived types. The basic types include the patches and walls, on which the other conditions are applied. Most commonly used for this are the primitive types [28]. The most important ones are:

- fixedValue: A constant pressure/velocity/etc is applied
- fixedGradient: A constant gradient of a parameter is applied
- zeroGradient: The gradient is zero
- calculated: The Boundary field is derived from other fields

There are many different derived types for very specific boundary conditions. The slip condition, as the opposite of the "no-slip" condition on walls, or all wall functions are part of this class.

3.2.3 Solver

For the incompressible aerodynamic calculations only steady state solvers delivered with the openFoam installation were used. For problems like this, openFoam® offers a broad band of specialized solvers suited for various flow situations. All are based on the earlier explained SIMPLE algorithm.

SimpleFoam was used for wind tunnel like computations with no heat influence. It can handle different air inlets and turbulence in the flow. It is one of the stable standard solvers in openFoam®.

*BuoyantBoussinesqSimpleFoam w*as taken when heat should be included in the calculations. For this a Temperature BC and gravity was added to the setup. Thus this solver does not include radiation, only diffusive and convective heat transfer.

In the case of a computation with no air movement, the *buoyantSimpleRadiationFoam* solver was chosen. Here heat distributions including radiation can be calculated, which is important if still air is a start condition, e. g. like in the "hot-room" tutorial.

injectorSolidParticleFoam is a particle flow solver developed by A. Persson [32]. It uses a steady state simpleFoam calculation to introduce a previously defined particle into the stream and follow it through the domain. It is helpful especially when heavy particles (e.g. exhaust ashes) should be traced that do not follow the streamlines as a lighter one as can be seen exemplary for two different particles introduced to the same flow situation in Figure 3.6 and Figure 3.7. It is not included in the openFoam® package but can be added easily.

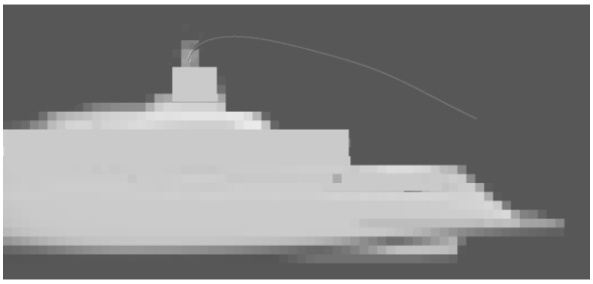


Figure 3.6 - Path of a heavy particle

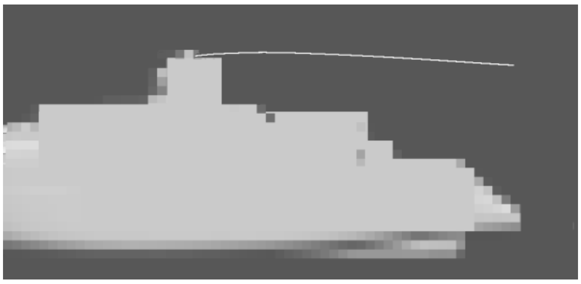


Figure 3.7 - Path of a lighter particle

3.3 Hydrodynamic calculation

The hydrodynamic calculations are very time consuming and need more preparation than the aerodynamic setup. Like in chapter 3.2 the setup, the boundary conditions and the used solver are explained.

3.3.1 Setup

The setup for the hydrodynamic calculation is rather simple since no geometry was varied. With that setup several items could be investigated:

- Laminar flow around hull: Wave pattern
- Turbulent flow around hull: Pressure distribution in the domain especially the wake
- Turbulent flow around hull including outlet flow: Pressure distribution in the domain

This can be achieved by just changing the boundary conditions. For a better convergence of the sensitive iteration a stepwise procedure was developed. First a laminar calculation for t=15s at U=2m/s (air and water) was started which helps to initialize a pressure field in a more stable non-turbulent environment. Now the wanted ship speed of 6m/s can be applied and run for another 15s. After this the turbulence model was switched on. After 5 more seconds an air velocity of 4m/s was applied at the exhaust outlet for another 5s.

In summary:

•	U_{water}= 2 m/s	U_{air}= 0 m/s	t=15s	turbulence model = off
•	U_{water}= 6 m/s	U_{air}= 0 m/s	t=15s	turbulence model = off
•	U_{water}= 6 m/s	U_{air}= 0 m/s	t=1s	turbulence model = on
•	U_{water}= 6 m/s	U_{air}= 4 m/s	t=9s	turbulence model = on

The setup was realized with the openFoam® solver interFoam. A turbulent calculation setup with an almost identical grid by starCCM+ (For time reasons without the exhaust outlets) was run and compared to the interFoam results. Also comparison studies with the shipFoam solver were started but not finished during the work due to the very long run times of the solver.

3.3.2 Boundary Conditions

The boundary conditions were achieved by iterative trails and base on the "Wigley Hull" tutorial by Eric Paterson [18]. They can be seen in Table 2 and are similar to the values in Table 1.

Table 2 - Boundary conditions for the towing tank

	Alpha	Velocity	Pressure	k	ω	nut
Inlet air	FixedValue 0	fixedValue	Buoyant-Pressure 1	Fixed-Value	Fixed-Value	FixedValue
Inlet water	FixedValue 1	fixedValue	Buoyant-Pressure 1000	Fixed-Value	Fixed-Value	FixedValue
Outlet	ZeroGradient	Zero-Gradient	ZeroGradient	Zero-Gradient	Inlet-Outlet	Zero-Gradient
Sides	Symmetry plane	Symmetry plane	Sym. plane	Sym. plane	Sym. plane	Sym. plane
L. Wall	ZeroGradient	fixedValue	ZeroGradient	kqRWall-Function	omegaWall Function	nutWall Function
Top	inletOutlet	Pressure-InletOutlet-Velocity	Total-Pressure	Inlet-Outlet	slip	InletOutlet
Ship	ZeroGradient	fixedValue	Buoyant-Pressure 0	kqRWall-Function	omegaWall Function	nutWall Function

3.3.3 Solver

To calculate resistance parameters openFoam® offers various solvers that incorporate the VOF method and are thereby able to calculate free surface water flows and body forces. For seakeeping calculations moving mesh solvers that handle 6 Degrees of Freedom (DoF) are available.

InterFoam is a transient solver for two phase flows using the compression scheme of the openFoam® VOF method. For the ship hydrodynamics it requires a very good mesh or otherwise it is very unstable. It needs very low time stepping when the grid gets finer to calculate the pressure fields.

InterDymFoam incorporates the same properties as the interFoam solver but allows moving meshes as well with 6 DoF. This allows sinkage and trim calculations.

Because of the unstable behavior of interFoam *shipFoam* was developed by M. Couwenberg [33]. It is a solver for fixed or free trim calculations of floating vessels and based on interDymFoam. ShipFoam is especially designed for ship hydrodynamic problems and can perform the same tasks as interFoam and interDymFoam. A specialty of the solver is the shipDict file where the mass, center of gravity, moment of inertia and other ship properties can be defined. Hull motions, center of gravity shifts and force balances are written out for every calculation into text files. It also includes a feature to force velocities and accelerations on the body or hold certain degrees of freedom fixed.

4 Quality aspects

For now it has only been said, which calculations were run during this thesis. In this chapter the arrangements to ensure a good calculation quality besides the already performed aspects are presented.

4.1 Setup quality assessment

To guarantee a good quality CFD also in later used models, the setup and the grid condition was investigated before and during the calculations. No analytical solution is available for these specific computations so that verification cannot be done in this thesis, but it is possible to choose verified or validated models performing best for a similar problem to reduce the error this way.

Industry standard and the chosen approach is the Finite Volume Method with a two-equation RANS model. For all calculations this setup was considered best, while keeping the computational and time expenses in reasonable limits. DES and LES are more accurate and many studies exist that describe their advantages over RANS methods, but they are not feasible yet in the fast changing and money limited ship design industry.

Two-equation RANS methods give good predictions of most ship industry flow phenomena and are nowadays verified in many studies. Especially in ship hydrodynamics conferences like Gothenburg2000 [31] or Tokyo2005 [34] using the Wigley hull, the 5415 combatant vessel or the KVLCC2 crude oil carrier these methods were validated and proved their strong performance.

As the turbulence model a k-ωSST was chosen for both calculated problems. In an extensive study by Bardina, et al. [16] it was considered the best overall performing turbulence model and is especially well operating for aerodynamic flow situations with high quality results. The results for hydrodynamic problems are also reliable although the gap to RSM or LES is larger [35]. An important problem for the k-ωSST model can be that the y+ value is out of scope (Reasonable values would be 30 < y+ < 300 [26]). This leads usually to diverging solutions and is thereby easy to detect. Unluckily is it not simple to adjust the value with snappyHexMesh. Apart from this the k-ωSST model is very stable and robust compared to other two equation models, which was also shown during this thesis in short comparison studies.

Convergence of a solution was considered when the residuals for ω, k, U and p were smaller than 10^-3 as recommended in [26]. This means that the variations in the solution between iterations is small and does not change significantly anymore. OpenFoam® allows easy handling and plotting of the residuals. Another convergence criterion was the rate of change of the force coefficients. By adding a few lines into the solver, forces on the body (usually lift and drag) and their normalized coefficients can be written out for each iteration and be plotted. Convergence problems can also occur due to unsuitable numerical schemes. The schemes need to be fitted to each problem individually, e. g. if high skewness was a problem like in the aerodynamic calculation, upwind schemes were avoided and skewness correctors were applied. An introduction and description to several schemes and their properties was handed over to the yard in the handbook. Generally all lower order schemes are more inaccurate, but higher order schemes tend to be more unstable. Usually the first tests were started with upwind and then switched to linear (for aerodynamics) and van Leer (for hydrodynamics) once the solution converged with the first order scheme. The presented results were performed with the higher order schemes.

4.2 Grid convergence study

As previously mentioned, complex grids created with sHM tend to incorporate certain failures that affect the outcome of the calculations. Since the results could not be compared to an analytical solution or a DNS result, other quality measurements had to be performed. For this several computations with different grid sizes but same setups were performed during this thesis.

This grid convergence study or grid refinement study is a method to detect the ordered discretization error in CFD computations. A detailed procedure of this can be seen on the NASA website grc.nasa.gov [36]. The idea is that the spatial discretization error goes towards zero, when the grid is refined. The error is coupled to the order of the numerical scheme that is used and can be estimated with various methods e.g. in Stern, et al. [37] which advise a minimum of three grids.

The aerodynamic setup was calculated with four different grids doubling the size each step since the error was still too large with the third mesh. The grid refinement ratio was roughly 2. Essentially this is because sHM works automatically and often places cells a different way when more cells are available. This means that not every cell was split in half in one refinement step, but the overall cell quantity was doubled. The settings of sHM were not changed though, but the quantity of the blockMesh cells so that a constant and even refinement can be assumed. In Table 3 values are displayed taken from those calculations for the various grid sizes. The values for U_x and p were printed from a single cell upstream of the ship were a stable flow could be anticipated [36] .

Table 3 - Grid convergence study results

	Coarse #4	Medium #3	Fine #2	Finest #1
Cell amount	~500.000	~1.000.000	~2.000.000	~4.000.000
U_X	- 3.9816	- 4.0971	- 4.1084	- 4.1086
P	0.311	0.289	0.284	0.283
C_D	- 6.7761	- 6.5615	- 6.242	- 6.233
C_L	8.748	8.539	8.437	8.436

One can see in Table 3 that the values converge towards a solution the finer the mesh gets. This relationship shows nicely the background and expected outcome of this study. The convergence towards a fixed value is the cause for an iterative process and can be seen in Figure 4.1.

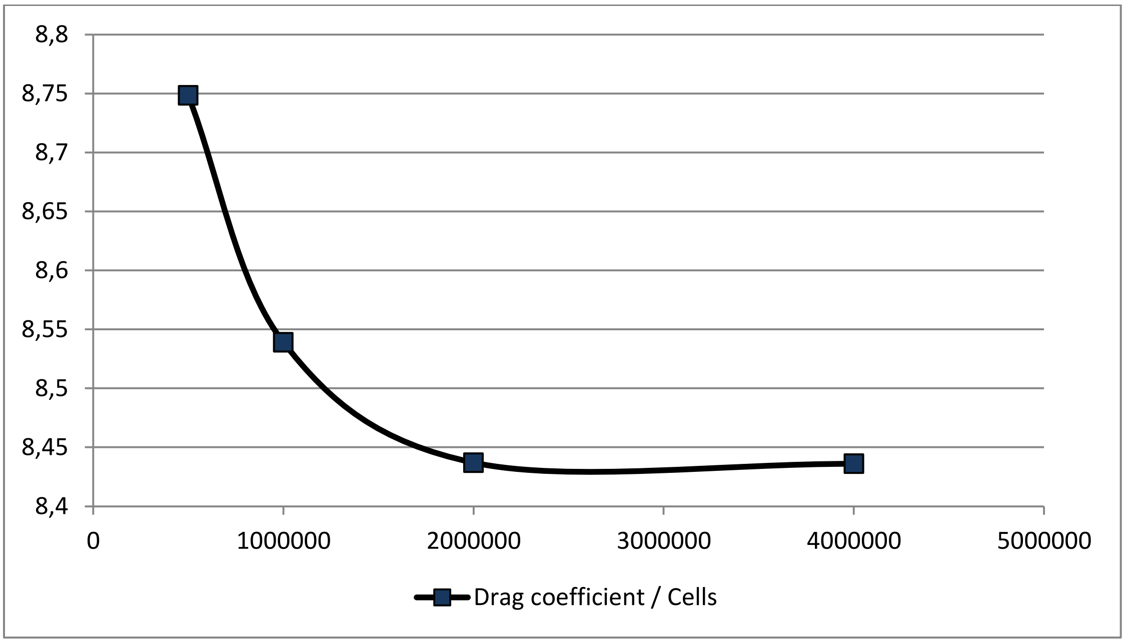


Figure 4.1 - Convergence of the drag coefficient with rising cell numbers

Using the Richardson extrapolation [38] described on the NASA website [36] one can get a higher-order estimate of the flow field values and with this reduce the error due to the coarser grids. Table 4 shows the outcome of such a calculation with a safety factor of 1.25. Interesting are the Grid Convergence Indices (GCI) and a ratio that indicates the asymptotic range of the solution. They should be either as small as possible - for the GCI - or as close to one as possible - for the ratio.

Table 4 shows that an acceptable small error can be assumed in the finest grid. The only parameter with a GCI value over 1% was the pressure, a critical value for many openFoam® solvers. Its residuals need to be checked carefully during the runs. This knowledge was now transferred into the calculations and the cause for using grids larger than three million cells in the computations. Still this guarantees only a low discretization error and does not say anything about the overall error of the computation.

Table 4 - Grid convergence calculations

	U_X	P	C_D	C_L
Richardson Extrapolation	**- 4.108604**	**0.28275**	**6.2327**	**8.43599**
GCI #1 / #2	**0.011%**	**11.04%**	**0.524%**	**0.015%**
GCI #2 / #3	0.62%	55.02%	18.55%	1.497%
GCI #3 / #4	6.36%	237.9%	11.89%	3.03%
Ratio #12 / #23	**0.999951**	**1.003534**	**1.00144**	**1.00012**
Ratio #23 / #34	5.505123	1.156349	55.2753	50.3605

5 Results

In this chapter the results of the two main calculations the aerodynamic and the hydrodynamic calculations are presented. Due to the secrecy policy of the yard this section has more a descriptive character than comparing exact values, i.e. only quantitative comparisons will be presented.

5.1 Validation of the aerodynamic computations

As has already been said there are many sources of errors in a CFD calculation. According to Larsson/Raven [11] these errors can come from an inadequate representation of the mathematical model by the CFD code or from a not exact modeling of the physical reality. The former is checked by verification of the code, the latter by validation of the results with experimental data.

In Appendix A comparisons of several computation results are displayed. Two aspects were of interest during these calculations. First the performance of openFoam® was investigated compared to the wind tunnel results and second different funnel geometries were compared for the yard. The detailed values of the setup can be found in chapter 3.2.1. The performance aspects and also the different displayed parts are:

- Exhaust flow with zero angle of attack (Also pipe geometry comparison)
- Exhaust flow with a small apparent wind angle (Also pipe geometry comparison)
- Exhaust flow with a larger apparent wind angle
- Boundary layer development testing
- Recirculation area testing

Table 5 shows the performed CFD calculations to validate, if the model that was provided to the yard makes reasonable conclusions about the flow possible. Only the three highlighted (black) variants are displayed as examples for their groups in the appendix together with the wind tunnel results since they show already the general behavior of all computations for that particular field of research (zero angle, small angle, etc.). The comparison of most CFD results showed a satisfactory or better agreement with the pictures from the wind tunnel tests. No reasonable difference was found between the normal and the heat included pipe flow scenario, so the use of the simpler model is recommended for standard wind tunnel like experiments. The Appendix A shows only results from MYacht. The similar AYacht was tested shortly and only as output results for the yard. It is not included in the appendix section but it showed analogical behavior. A full comparison of the AYacht was not performed due to time limitations.

Table 5 - Aerodynamic validation cases

App.Angle/Variant	#1(low pipes)	#2 (medium)	#3 (high)	#4 (high)
0° (True 0°)	o	o	o	**o**
17° (True 30°)	**+**	+	o	o
51° (True 60°)	x	x	**-**	x

+ = good, o = satisfactory, - = unsatisfactory, x = not calculated due to time reasons

The overall performance of the first two groups, i. e. no and small angles of attack, was good (Figure_apx B.1, Figure_apx B.2, Figure_apx B.4, Figure_apx B.5). A problem of these groups was that the calculations only show the streamlines of the air flow. They are close to the results, but the influence of gravity on the heavier oil particles used in the wind tunnel was noticeable by a wider spreading of the wake in z direction. This makes it also difficult to judge the CFD results, when the flow gets very slow. You can assume that the exhaust is not carried any further, but the calculated streamlines go on. The inserting of heavy particles into these critical flow situations with injectorSolidParticleFoam can be corrective. The Figure_apx B.3 and the Figure_apx B.6 show clearly the possibilities openFoam® offers the yard in comparing different variants. The influence of different funnel geometries has a clearly visible effect on the flow over the aft decks of the vessel.

Convergence is harder to achieve when the angle of attack went above 30° (called “larger angles” in the Appendix). To achieve a converged solution low relaxation factors and low order schemes needed to be used so that the quality of the computation dropped significantly. The highly rotational flow combined with the skewed mesh could be a reason for this behavior and result in an insufficient prediction of the flow field. For time reasons no deeper investigation of this circumstance was done and also of no other variants than #3 were performed with success. Even though one late example calculation of variant #3 showed that starting with a laminar calculation first and applying a stepwise procedure like described in the setup chapter (See chapter 3.2.1) can produce much better convergence with higher quality. Each calculation takes much more time then, so it is preferred to calculate without a low initial velocity and to switch turbulence models on from the beginning. An automated coupling of the convergence behavior and the setup of openFoam would be useful for these applications. When the solution converged the flow was predicted too high in z-direction. It could be the same problem of the low velocity carrying the light air away in the CFD simulation (Figure_apx B.7) but not the heavier smoke in the experiment (Figure_apx B.8). The view with only the streamlines in the post processing could be misleading here.

Another important and interesting behavior to investigate was the development of the boundary layer over the ship (Figure_apx B.9 & Figure_apx B.10). Trends on how the flow of particles in the air will develop around the superstructure can be seen from this. Here the same results as in the other groups could be seen. Good results for small or no angles and sometimes convergence problems for larger angles with the k-omega SST model were noticed. A general problem that can occur can be seen in Figure_apx B.11. The recirculation area clearly visible in Figure_apx B.12 was not fully resolved by simpleFoam. The resolution might have been not high enough for this complex occurrence or it was simply not captured with this way of post processing. For deeper investigation of this problem was unluckily no time left as well.

5.2 Validation of the hydrodynamic computations

To assure convergence of the hydrodynamic calculations with the sensitive multiphase solvers a stepwise procedure needed to be done. This fitted into the setup since no variation on the geometry but only on the boundary conditions was needed. The biggest problem was a highly increasing Courant Number and a following crash of the solving process. This could either be a grid quality problem or the problem of a too high time stepping. The time stepping varied between 0.001s and 0.02s.

When the Courant number went too high, the time step was lowered and the simulation was started new. Adaptive time stepping did not give the wanted results and normally just delayed the crash. Lower time stepping is needed when the turbulence model was turned on as well. Sometimes the pressure did not converge, especially when dynamic mesh moving was active, i.e. the ship was free to sinkage and trim, and interFoam (or interDymFoam) crashed with the same high Courant number problem. Table 1 shows that a large lowering of the time step could prolong the calculation also in this case, but the solution took so long that it is not practicable to work with.

Table 6 - Convergence problems interFoam and interDymFoam

Time step	Failure after seconds	Number of Iterations
0.002	0.076	38
0.001	0.043	43
0.0001	0.0056	56

Among other things this was a cause for the development of the shipFoam solver. Especially for sinkage and trim calculations better converging solutions were achieved than with interDymFoam. There are no results to compare yet with the interFoam solver for the hydrodynamic setup since the code structure of shipFoam does not allow it yet to be run in parallel. The process was started but not finished at this point.

Reviewing the results in Appendix C one can see that interFoam produces good output from a ship hydrodynamic point of view compared to other CFD programs. The wave pattern is very alike compared to starCCM+ and reasonable compared to similar HSVA results which cannot be printed for yard policy reasons. OpenFoam predicts a little higher wave heights then the commercial solver, but both are over predicting compared to HSVA (Figure_apx C.1, Figure_apx C.3, Figure_apx C.15 & Figure_apx C.16). Especially the wake wave is too high when computed with the turbulence model of openFoam®. The little bit disturbed pattern in Figure_apx C.3 is believed to come from the short time between the switching to turbulence and the picture. This should be longer than 5s. A more steady state than was reached in Figure_apx C.4. But former studies of the yard and also the wake wave in Figure_apx C.4 show that usually more time is needed for a full steady state of the computations. Again time limitations restricted this study.

When looking only at the streamlines around the bow one can see that they are very similar and the velocities were predicted the same from both programs (Figure_apx C.5 & Figure_apx C.6). They can be a good help when optimizing the flow around underwater structures of the ship. The same holds for the description of the streamlines around the exhaust outlet (Figure_apx C.13). When the exhaust flow from the underwater outlet was analyzed, Figure_apx C.14 proves that interFoam provides moderate output. It displays three time steps during the development of an air outflow. Figure_apx C.4 shows nicely how the air flow from the exhaust affects the wave making of the ship. This will most likely not be this high magnitude in reality since these were unpractical test values but it proves that openFoam® can handle this input as well.

The absolute pressure was calculated as expected and very similar with both solvers. The difference in the annotation comes from different starting points - openFoam® starts at 0 Pa and starCCM+ at 10^5 Pa. This is no problem when incompressible flow is assumed. The pressure patterns visible in all figures (Figure_apx C.7 - Figure_apx C.12) were almost identical. The stagnation point and the low pressure areas on convex parts of the hull were resolved. A minor difference of 2000 pa was noticed between the solvers. From the wave height comparison with HSVA results it can be seen that openFoam is then over predicting the pressure field.

6 Discussion

During this thesis different openFoam® tutorials for the Fr. Lürssen yard were created mainly to solve aerodynamic and hydrodynamic flow situations around mega yachts. In detail this was a streamline analysis around the superstructure and a pressure field study on the ship hull. On the one hand did the yard really need those calculations as a judgment base, on the other hand were they a good foundation to gain expertise in the field of openFoam® for the yard and to show, what this tool can do today. Furthermore mockups to analyze heat distributions, particle injection and dynamic mesh movement (sinkage & trim) were established but not as in depth tested as the former tutorials. Several topics that indicated their relevance during the work were investigated. In particular these were the software sHM, the different openFoam® solver, numerical schemes and parallel computing. All research results were given to the yard including a guideline and an instruction manual for the tutorials. This was done in line with a study about the integration of open source software (OSS) on a medium-size yard (~1000 employees), which will be summarized in this section. For this reason openFoam® was also compared to the commercial software StarCCM+ by CD-adapco, but only regarding the specific study field of hydrodynamic ship flows.

But the thesis has another use for the yard beside the performed computations. It was also meant as acceleration and strengthening of Lürssen's work with openFoam®. On the one hand the engineers get familiar with the topic much quicker, which can avoid an expensive training course; on the other hand new cases can be treated fast, effective and cheap with the standard models. Controlled changes to the models are quite easily possible which makes openFoam® a good and solid tool, especially when running repetitive or similar tests. These changes could regard the geometry, different flow situations or velocities. Based on the tutorials also other computations are feasible. A propeller could be included to the hydrodynamic calculations to analyze their influence on the wake. Also pipe flow simulations could be an interesting work area, e. g. to simulate the exhaust flow already in the exhaust gas system. The work with openFoam® improves the overall comprehension for CFD calculations as well since it penalizes failures or a wrong setup more severely than most programs. This forced way of learning has a positive effect on the understanding of other tools and their specific strength and weaknesses. In addition to that, openFoam® offers itself several possibilities to explore further fields in CFD analysis. There are specialized solvers on multiphase simulations, chemical reactions, flame and engine simulation or the settling of solid matter in fluids just to name a few computations that could be interesting for a yard. Apart from this more experience is needed the further the yard engineer develops solutions away from the openFoam® tutorials. But normally there will not be too much valuable time spent with the introduction of a program, since solid results are always needed as fast as possible on a modern shipyard. This leads to the question of which level of knowledge is necessary to start using open source software. A basic requirement is, as in every CFD program, good to very good interpretation skills of fluid dynamic processes to judge the results in the right way. Often a solution looks the right way, but has minor errors that are noticed right away by experienced fluid engineers. Also important is of course basic knowledge of CFD software and the handling of Linux based systems. Most companies use Microsoft® Windows® systems and the difference in work is not that simple to cope with for IT (Information Technology) inexperienced personal. The comprehension of programming languages especially of C++ is advisable but not essential. The openFoam® source code is open to use and restructure for everyone, but might be not intuitive to understand, again for engineers without an IT background.

6.1 CFD methods on a shipyard

The main aspect of choosing CFD on a shipyard is to decrease or replace laborious and expensive experiments. Free software has here of course an extra advantage compared to commercial ones. During a simulation the response to the needs of the design and construction is more flexible than during an experiment. If the model changes after an experimental stage they can only be taken care of with large additional expenses. Normally the two main tests in the early design stages of a ship are the same in most ship yards - the aerodynamic wind tunnel testing and the towing tank tests. This is especially easy to realize with the openFoam® case folder structure. Furthermore is it easier to adjust and to reconstruct the setup parameter than in an experiment which is done usually externally by test facilities.

Compared to the popular rule of thumb, CFD methods are usually more precise. The engineer can cover more areas with computer simulations, which are usually not realizable with simple formulas e.g. the design of high performance rudders or similar extreme applications [39]. Rule of thumbs are too limited and introduce a high modeling error. On the other hand it is often difficult for employees on yards to integrate structural changes from a time point of view. The famous "We have always done it that way" mentality is mostly not a sign of stubbornness but that it cannot be managed internally any other way. That this hinders the use of CFD or other new methods is not uncommon in shipyards like in most other industrial offices.

As it is with many other innovative techniques, this attitude can produce problems for an internationally acting company since CFD can help to reduce costs and to increase the product quality. To maintain the quality advance of the industrial nations against the cheaper producing countries engineers need to be able to work flexibly and with the newest technologies. By missing out technological innovations a company increases the risk of reacting too late to new rules or trends. This is especially a danger for high technology yards and should be avoided. OpenFoam® offers here an excellent chance to gain experience in a new development without taking high investment risks.

As a third step in this analysis the theoretical fields of usage for CFD applications on a yard and due to its high development status also openFoam®´s possible purpose areas will be listed. One of the economically most important goals of naval architects on a yard is the minimization of the ships resistance, especially the wave resistance and the viscous resistance. For the former well established potential flow solver like nuShallo by the HSVA or shipFlow® by FlowTech® are used, which outmatch the openFoam® potential flow solver "potentialFoam" due to their intense marine focus [13]. Regarding the viscous calculations this study has shown that openFoam® shows comparable performance already today in ship hydrodynamic calculations.

Together with the viscous resistance the pressure/velocity distribution in the flow field and on the hull can be calculated. Here came into notice when comparing to StarCCM+ that the same result quality can be done with open source software for ship hydrodynamic calculations.

From this you can tell that for resistance computations around appendages these or similar results can be expected. Also cavitation studies have been performed successfully in openFoam® on NACA sections or propellers, e. g. by Bensow [40]. The practical relevance on a shipyard is doubtful for now, since the appendages are usually designed by the manufacturers and you need the time consuming LES method to get reasonable results.

Aerodynamic calculations as well as windage estimations can be done successfully with most RANS using CFD methods and good outcome can be expected also with openFoam® as it was shown during this work. When the flow around complex geometries is analyzed good care should be given to the modeling and high resolution should be applied to fully resolve all phenomena. Particle flows are usually not necessary to achieve good results, but can be included. To represent moving ship structures dynamic meshes are applicable to most modern solvers. With them and advanced boundary conditions floating and seakeeping calculations are possible and allow a more realistic approximation of the ships behavior. OpenFoam® has the theoretical requirements to solve moving mesh problems. In reality the current solvers have problems with complex grids which are needed in floating ship calculations.

6.2 Comparison to a commercial solver

To get a better understanding of how openFoam® differs to commercial software and thereby to understand what the specific problems of the work with open source tools can be some direct pros and cons are listed in Table 7 to compare StarCCM+ and openFoam®. The most important arguments are explained also on the next page.

Table 7 - Comparison of openFoam and StarCCM+

Aspect	OpenFoam®	StarCCM+
Costs	+ Free of charge	- Expensive
Installation	o Problems when not using Ubuntu	+ Easy for Linux and Windows
Handling	- No GUI	+ Graphical pre-processing
Meshing	o Both methods are very powerful and include latest meshing techniques. Still there can be problems with both of them.	o See left
Pre-processing	- Many functions need to be included by the user	+ Everything in one tool
Different tasks	- To check if setup was right during the run is difficult	+ more flexible
Repetitive tasks	+ Fast to use, if the setup is not changed	o Saving of setup for other ships not as simple as in openFoam®
Post-processing	- Paraview unstable	+ Online and in one tool + Concentrations easy to visualize
Results	o Comparable quality, but larger effort to achieve them	o Very stable and fast solver, reasonable physical behavior
Support	- Support	+ Software help and customer support
Speed	+ Parallel computing + Parallel meshing	- Parallel work includes extra costs
Extra functions	o Not explained and hard to find	+ More functions
Expandability	o Large due to open source code	o Large due to programming interface
Solver	o Many solver for special cases	o Everything in one tool + Insertion of particles easier
Schemes	+ Many numerical schemes to choose from	- Just a few schemes
BC´s	o Many BC´s with own programming	o Many BC´s with own programming
Learning	- Complex setups need too much time to work yourself in	+ Intuitive program in combination with the help function
Others	- Single cell difficult to address	+ Cell remedy for single cells
		+ Online access to field functions

The most obvious difference is the missing GUI, which seems to hinder the intuitive use and the learning of openFoam®. The whole setup is controlled via text files that contain the parameters that change the setup. These parameters have mostly a preset value and often no explanation what the parameter is used for. The tutorials on which the work of most openFoam® beginners is based on, include also only the necessary text files and not the full band of possibilities. This makes it difficult to fathom the capacity of the tool as well as the fact that it consists of many different small applications rather than one combined one.

A following problem evolves from the missing online support which is a big disadvantage for a complex CFD application where bugs in the program and setup problems are difficult to distinguish. A handbook with the explanation of some standard CFD problems is the only help for the program. Online forums like cfd-online.com [18] help to reduce complexity but are not practical for every problem. StarCCM+ offers, as almost all commercial tools, a practicable GUI where each point is explained by a help utility and the parameters are arranged to ease up the learning and pre-processing steps. More advanced questions are fastly answered by an online support team.

The commercial solver itself seemed to be a lot more stable programmed. While a stepwise increase in velocity and also first iterations without turbulence where necessary in openFoam® to get a converging solution for the 6m/s hydrodynamic calculation, the starCCM+ solver did the same task in one run without problems and with a likewise result (see Appendix C).

A main drawback of openFoam® in post-processing is that a check of the temporary results (e. g. pressure field) is only possible in previously defined time steps. These are written into the case folder for each step and produce a large amount of data. If only one value was typed incorrectly, the user usually sees his mistake after the whole run. This can result in useless loops. Post-processing is done in openFoam® with the paraview based application paraFoam (In this thesis version 3.8.1). In commercial tools the data writing is handled the same way but you can also visualize online the actual time step in the GUI. This enables the user to follow the development of the calculations closely but to keep the amount of data in a reasonable range.

The largest advantage for any economically acting company compared to commercial solvers is certainly that openFoam® is cost free, i.e. needs no license to be paid for. As a result it is very simple to parallelize computations with openFoam® by running it on the several cores of a single computer. Commercial software usually needs as many licenses as the computation is split into. With 6 to 8 cores already on normal desktops nowadays this can lead to a major cost issue. And also the performance of the MPI (Message Passing Interface) parallel computing is very good with openFoam®. In a short study with both simpleFoam and interFoam it could be shown that no or only little time and information is lost due to domain splitting or value transfer between the domains (called processors in openFoam®) of a parallel computation. If a tutorial was set right for a specific problem openFoam® can be a very powerful and fast tool. StarCCM+ can be unpractical sometimes for repetitive calculations, openFoam® enables the user to set up and run a second similar computation in only a few hours. After the aerodynamic setup for MYacht was completed, AYacht took less than 3 hours to setup. Most of the time was spend for creating, adjusting and meshing the .stl file, while the restructuring of the boundary conditions and needed dictionaries is done very quickly. The calculations itself got reasonable results with sometimes little unrealistic physical with openFoam® which is believed come from the time limitations of the thesis and thereby not a complete handling of all possible methods.

The benefit of the excellent expandability of openFoam®, since anyone with good C++ and CFD skills can change and rebuild the program completely user-defined, was caught up by now by most commercial solvers with many modification functions and programming interfaces incorporated in the software.

6.3 Sustainability aspects of the work

The exertion of CFD software, especially in openFoam®, implies also many options for a sustainable improvement of the systems. To decrease the needed engine power and with that the fuel consumption and exhaust production, the optimization of the ship hull is one of the most effective methods. Important for this is the calculation of the wave pattern or height which admittedly can be done as standard practice on most shipyards with potential flow solvers as it has been said already. The improved resolution and illustration of the ships aft body and wake and a possible optimization of the propeller inflow with a RANS solver is not that common and could lead to higher ship efficiency. The same holds for the design and positioning of appendages (e. g. the rudder, the stabilizers, etc.), even though this is usually done by the producing companies and not by the yard.

Imaginable are also cost reductions due to an optimization of the flow in the exhaust system line and with that better adjustment options for the engine and other involved devices. The flow phenomena of technical equipment in general are only slowly covered with CFD methods and many possibilities for quality improvements are thinkable. When using openFoam® to calculate heat transfer phenomena in detail, e. g. on the funnel or other heat pipes, insulating material could be reduced since a clearer temperature distribution is known by the engineers. To simulate the HVAC properties of a standard room onboard can increase the energy efficiency as well and avoid for an example that heat sensitive devices are damaged, dangerous blind areas in which gases can accumulate occur or simply too much energy for cooling and heating is needed.

Energy savings resulting from a flow optimized arrangement of the superstructure, e. g. using the aerodynamic model tutorial, is also very easy to achieve. The windage or air resistance component of the overall resistance of a ship is indeed much smaller than hull resistance but, combined with the better conditions for people on board due to lower exhaust gas concentrations, an interesting field for yard engineers to look into. Less effort in deck cleaning due to lower particle concentrations on board will also contribute to a less polluted environment.

A last but crucial factor to use CFD like in all model tests is the avoidance of "unpleasant surprises" in full scale. Material and energy intensive rework can be prevented by good simulations beforehand that allow a more detailed preview of the future performance of the vessel. The three main advantages of CFD simulations can be seen again in Figure 6.1.

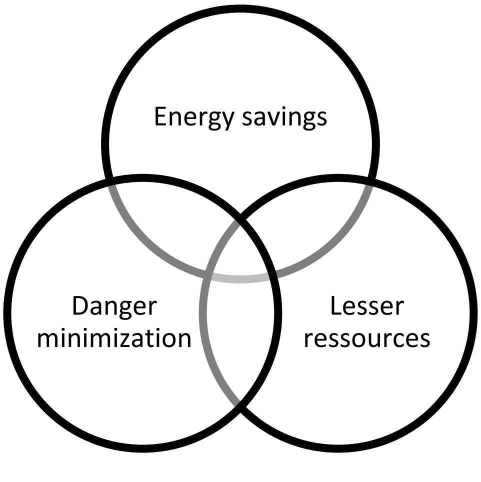


Figure 6.1 - Sustainable aspects of openFoam usage

6.4 Conclusions of the thesis

The positive aspects of openFoam® named in the last section do not come for free or, like often in engineering, without drawbacks. The main problem and the cause for the so far lower spreading of openFoam® in the ship building industry lies in the low support and the unintuitive handling of the tool. On the one hand is an expensive openFoam® training contrary to the free of charge advantage of the application against commercial software, on the other hand is it hard to learn due to the many unexplained setup possibilities and parameters. But still it is tested by many yards at the moment due to its high possibilities and low usage costs.

A GUI would facilitate the work with CFD software for beginners on a yard and makes it easier to legitimate the time consuming computations due to quickly produced results. To be more attractive for ship yards openFoam® needs a simplified handling in which good working parameters are fixed to the program. The large variety of options might have a big value for the academic area. In the daily routine on a ship yard, where reliable and fast information is the basic requirement for the use of any simulation software, a simplified GUI supported version of the qualitative premium tool, comparable to most commercial software, would have a big use.

If the good reproducibility due to the copy and paste case structure stays as well, openFoam® might have good chances to solve the repetitive tasks that shipyards often have to deal with. Interfaces to other applications are not excessively available and are hindered furthermore by the compatibility only to the Linux system. Most yards would need to include the OSS to their standard Microsoft® Windows® environment, which might be a problem. Much time can be spent setting up a new Linux system if no experience in this field is available at the yard. The overall demand of CFD is in addition not as high for now in the ship building industry as it could be and as its possibilities offer. From the view of the 1990s it is said to be too slow and the trust in the quality of the calculations is not very high. But this thinking is turning in the industry in the recent years.

OpenFoam® especially turns slowly as well towards the maritime industry. For now there are not many specialized applications for ship building. In particular the special interest group - ship hydrodynamics (SIG) does pioneer work in this field. The openFoam® extend project provides the platform for exchange. However the SIG is very open and thereby unstructured in the way of work. Selective efforts to research on one topic e. g. a well working RANS wake calculation, is done in academics but mostly between the universities own CFD codes. The openFoam® workshops are here a possibility for better exchange.

Jasak and Rusche [41] have listed where openFoam® works for maritime applications at the moment, but mostly on a for companies not feasible university level. The same holds for the impressive contributions of Eric Paterson [42]. A more practical approach was tried by the engineers of STX France [43] with an optimization study performed completely with openFoam®. Their good results showed also the large practical benefit that openFoam® can offer to the industry.

Another contribution comes, as has already been mentioned, from Couwenberg [33] implementing the shipFoam solver particularly for ship applications. With this tool 6DoF, pressure and force calculations are theoretically possible. But during this thesis it was noticed that work is still needed here. Together with R. Moolenaar the Author has worked on a shipFoam version for openFoam® 1.7.1.

Special applications that each covers a small part of the ship analysis in a dependable way could be a step into the right direction. This would be easier with a better coordination of development plans for openFoam® so that many developers could work into the same direction. Especially the leading openFoam® universities could have a key role in this process. Since a direct support would disagree to the open source mind better communication platforms than forums, in which you can only search for similar errors or hope that someone answers to the problem, might be needed. A better listing and sorted collection of questions and answers to special topics in openFoam® could ease the entrance to the software for newcomers, and by that overcome the restraint of the ship building and other industries to use open source CFD software.

7 Future work

A part of the future work on openFoam® in the maritime sector could be the programming of a working viscous solver for pressure and resistance calculations on the ship hull. The simultaneous implementation of a stable 6DoF solver is necessary to calculate sinkage and trim at the same time. The shipFoam solver has all the requirements to fulfill this but needs further investigations to further reduce instable behavior and to make the solving process faster than it is.

Regarding the yard, new models and add-ons to the existing models are imaginable. Depending on the use the application area can or should be changed or increased. An extended simulation of the exhaust system could give a better view onto the flow behavior at the funnel. For the underwater analysis the adding of appendages and their effect on the openFoam® solving process needs to be tested. The correct positioning of the propellers and rudders is of great interest for the yard and should be realizable with the present models.

For the hydrodynamic as well as for the aerodynamic calculations pure air was assumed for the exhaust since that introduces just a small error and works well with the current openFoam® solvers. The program is not ready yet to calculate multiphase flows in such complex environments and with only small time for the setup and the calculation allowed. It could be a work for the future to set up a standard model for multiphase gas flow problems in the ship industry. But first the openFoam® environment needs some changes to allow this.

The actual multiphase solvers in openFoam® are programmed for a specific study field but might be suitable when modified. The dynamic mesh handling in openFoam® also needs further examination. InterDymFoam and shipFoam are not yet able to solve the behavior of every floating structure. The "sinkage & trim" - tutorial works but demands too much adjustment in the pre-processing stage to be useful for the yard yet. The functionality offers possibilities for the future but needs thereby further testing.

Another difficulty during the work was a missing standard procedure to run the ship CFD calculations like there is for the towing tank tests with the ITTC rules. A previously normed cell size, boundary layer thickness or mesh quality grade would result in an easier validation of the outcome of any CFD calculation with its model tests like there is for the few standard validation test case like the Wigley case or the model 5415.

List of References

1. **Andersson, B., et al., et al.** *Computational Fluid Dynamics for Chemical Engineers.* Sixth Edition. Gothenburg : s.n., 2010.

2. **Kulkarni, P.R. und Singh, S.N., Seshadri, V.** Parametric studies of exhaust smoke-superstructure interaction on a naval ship using CFD. *Computers & Fluids.* 2006, 36, S. 794-816.

3. **Moctar, O., Gatchell, S. und Bertram, V.** *Aerodynamic Flow Simulations around Ship Superstructures.* Hamburg : HSVA, 2001.

4. **Ferziger, Joel H. und Peric, Milovan.** *Computational Methods for Fluid Dynamics.* Berlin Heidelberg : Springer, 2010.

5. **Versteeg, H. und Malalasekra, W.** *An Introduction to Computational Fluid Dynamics: The Finite Volume Method.* Upper Saddle River, New Jersey : Prentice Hall, 2007.

6. **Blazek, J.** *Computational Fluid Dynamics: Principles and Applications.* Oxford : Elsevier Science, 2001. ISBN-13: 978-0080430096.

7. **Geller, W.** *Thermodynamic fuer Maschinenbauer - Grundlagen fuer die Praxis.* Berlin : Springer, 2006.

8. **Batchelor, G. K.** *An Introduction to Fluid Dynamics.* Cambridge : Cambridge University Press, 2000. ISBN 0521663962.

9. **Schlichting, H., et al., et al.** *Boundary-Layer Theory.* [Übers.] K. Mayes. 8th. Berlin : Springer, 2000.

10. **Craft, T. J.** Computational Fluid Dynamics and Turbulence Mechanics. [Online] 20. April 2011. http://cfd.mace.manchester.ac.uk/twiki/pub/Main/TimCraftNotes_All_Access/cfd1-pt1.pdf.

11. **Larsson, L. und Raven, H.** *Ship Resistance and Flow.* Gothenburg and Wageningen : s.n., 2004.

12. **Jasak, Hrvoje.** *Error analysis and estimation for the Finite volume method with applications to Fluid Flows.* Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine. London : s.n., 1996. PhD Thesis.

13. **Axfors, B., et al., et al.** *The ARC 45 - A Downwind Performance Cruiser.* Gothenburg : Chalmers University, 2010.

14. **Shih, T.H., et al., et al.** A new k-epsilon eddy-viscosity model for high reynolds numer turbulent flows - model development and validation. *Computers fluids.* 1995, Bd. 24, S. 227-232.

15. **Yakhot, V., et al., et al.** Development of turbulence models for shear flows by a double expansion technique. *Physics of Fluids.* 1992, Bd. Vol. 4, No. 7, S. 1510-1520.

16. **Bardina, J.E., Huang, P.G., Coakley, T.J.** *Turbulence Modeling Validation, Testing, and Development.* s.l. : NASA Technical Memorandum 110446, 1997.

17. **Wilcox, D. C.** *Turbulence Modeling for CFD.* La Canada CA : DCW Industries, Inc., 1998.

18. **GNU Free Licence.** CFD Online . [Online] MediaWiki, 2011. www.cfd-online.com.

19. **Menter, F.R.** Zonal Two Equation k-ω Turbulence Models for Aerodynamic Flows. *American Institute of Aeronautics and Astronautics Paper.* 1993, 1993-2906.

20. —. Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications. *American Institute of Aeronautics and Astronautics Journal.* 1994, Vol. 32, S. 269-289.

21. *A unified approach to the design and application of bounded high-order covection schemes.* **Waterson, N. P und Deconinck, H.** Atlanta : Pineridge Press, 1995. 9th Int. Conf. on Numerical Methods in Laminar and Turbulent Flow.

22. **Patankar, S. V.** *Numerical Heat Transfer and Fluid Flow.* London : Taylor & Francis, 1980. 978-0891165224.

23. **Hirt, C.W. und Nichols, B.D.** Volume of Fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics.* 1981, 39, S. 201-225.

24. **Youngs, D. L.** Time dependent multi - material flow with large fluid distortion. [Buchverf.] K. W. Morton. [Hrsg.] M. J. Baines. *Numerical Methods for Fluid Dynamics.* New York : Academic Press, 1982, S. 273.

25. **Rusche, H.** *Computational Fluid Dynamics of Dispersed two-phase Flow at High Phase Fractions.* London : Imperial College of Science, 2002.

26. **Ansys Fluent.** FLUENT 6.3 Documentation . [Online] Fluent Inc., 20. September 2006. http://my.fit.edu/itresources/manuals/fluent6.3/help/.

27. **Năstase, A.** *Computation of supersonic flow over flying configurations.* s.l. : Elsevier, 2008. 0080449573, 9780080449579.

28. **Free Software Foundation.** *OpenFOAM - User Guide.* Boston : s.n., 2008.

29. **Fehlmann, Andre.** *Parameterstudie zur automatischen Gittergenerierung der OpenFOAM Anwendung snappyHexMesh.* Bremen : s.n., 2010.

30. **Nordin, Niklas.** Contrib_snapEdge. [Online] openFoamWiki, 10. February 2011. http://openfoamwiki.net/index.php/Contrib_snapEdge.

31. **Larsson, L., Stern, F. und Bertram, V.** Benchmarking of Computational Fluid Dynamics for Ship Flows: The Gothenburg 2000 Workshop. *Journal of Ship Research.* 2003, Bd. 1, 47.

32. **Persson, A.** *Description and implementation of particle injection in OpenFOAM.* Gothenburg : Chalmers University of Technology, 2010.

33. **Couwenberg, M.** Sourgeforge - openFoam extend. [Online] 2010. http://openfoam-extend.svn.sourceforge.net/viewvc/openfoam-extend/trunk/Breeder_1.6/shipHydrodynamicIG/.

34. *Proceedings of the CFD Workshop Tokyo 2005.* **Hino, T.** Tokyo, Japan : s.n., 2005.

35. *Comparison of explicit algebraic stress models and second-order turbulence closures for steady flows around ships.* **Deng, G. B. und Visonneau, M.** Barcelona : Ecole Centrale de Nantes, 1999. MARNET-CFD First Workshop.

36. *NPARC Alliance CFD Verification and Validation Website.* [Online] 27. August 2007. http://www.grc.nasa.gov/WWW/wind/valid/tutorial/spatconv.html.

37. **Stern, F., et al., et al.** Comprehensive Approach to Verification and Validation of CFD Simulations - Part1. *Journal of Fluids Engineering.* December 2001, 123, S. 793-802.

38. *The deffered approach to the Limit. Part I. Single lattice. Part II. Interpenetrating lattice.* **Richardson, L. F. und Gaunt, J. A.** 1927, Philosophical Transactions of the Royal Society of London, Bde. Series A, Containing Papers of a Mathematical or Physical Character, S. 299-361.

39. **Blount, D. L. und Dawson, D.** Rudder Design for High-Performance Boats. *Professional Boat Builder.* Aug/Sept, 2002.

40. *Simulating Cavitating Flows with LES in openFoam.* **Bensow, R. E. und Bark, G.** Lisbon, Portugal : s.n., 2010. V European Conference on Computational Fluid Dynamics.

41. **Jasak, H. und Rusche, H.** Marine Hydrodynamics Solver in openFoam. [Online] 2008. http://powerlab.fsb.hr/ped/kturbo/OpenFOAM/slides/OSCFDC_4Dec2008.pdf.

42. *Simulation Of Wakes, Wave Impact Loads, and Seakeeping Using OpenFoam.* **Paterson, E., Smith, K. und Ford, S.** Montreal, Canada : s.n., 2009. 4th OpenFOAM® Workshop.

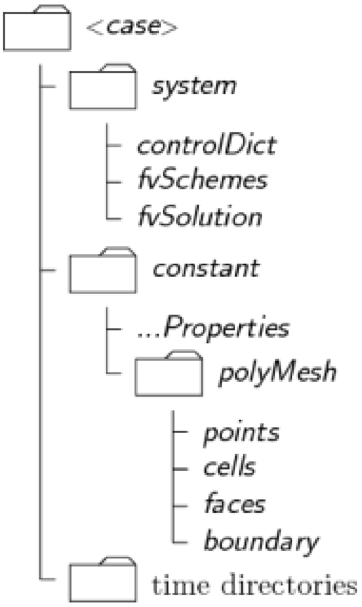
43. **Cordier, S, et al., et al.** *Application of OpenFOAM® To Hull Form Optimisation At STX France.* s.l. : RINA.

# Appendix A	Short introduction to OpenFoam®

An openFoam case is not controlled with an interactive user interface, but with text files (see Figure_apx A.1), so called dictionaries, and the Linux console / terminal. For a calculation three folders need to be created / copied:

- /system
- /constant
- /0

/system contains all information about the solving process
/constant contains all information about the mesh, turbulence models and other constants
/0 or /0.org contains the boundary conditions - the starting values for each field parameter



Figure_apx A.1 - OpenFoam® case folder structure [28]

OpenFoam® Handling

In the following section the most important dictionaries and input files are explained shortly. A dictionary in openFoam® contains all information that the program has to know. For an example all data to control the application blockMesh is stored in the blockMeshDict text file.

There is no online support, but some help can be found at:

http://www.openfoam.com/docs/user/index.php

With many problems help can be found also in online forums, like:

http://www.cfd-online.com/Forums/openfoam

Important openFoam® commands (Execute in the Linux terminal in the current case folder)

- Utilities: http://www.openfoam.com/docs/user/standard-utilities.php#x14-900003.6
- Solvers: http://www.openfoam.com/docs/user/standard-solvers.php

blockMesh	- Domain for the computations
checkMesh	- Check of the current mesh
decomposePar	- Dividing the domain into defined parts for parallel computing
foamJob -s -p COMMAND	- Parallel computing of the command "COMMAND"
reconstructPar	- Rebuilding of the domain (mesh, field values, etc)
reconstructParMesh	- Rebuilding of the generated mesh
	e. g. *"reconstructParMesh -mergeTol 1e-06 -latestTime"*
snappyHexMesh	- Grid generation of the .stl (File: system/snappyHexMeshDict)
transformPoints	- Treating of mesh points
	e. g. Turning of the mesh: transformPoints -yawPitchRoll "(0 0 0)"
	Scaling of the mesh: transformPoints -scale "(1 1 1)"
surfaceCheck	- Check of the .stl file
setFields	- Pre-processing of the phase distribution α and the field values
yPlusRAS	- Check of the y+ value

The most important folders

In /constant all the physical properties are defined. The most common are:

- transportProperties (Definition of the phase constants rho, nu, etc)
- turbulenceProperties (Choice of turbulence models: laminar, RASModel, LESModel)
- RASProperties (Choice of RANS models: z. B. kEpsilon)
- LESProperties (Choice of LES models)
- g (Definition of the gravitational constant)
- environmentalProperties (g in oF 1.6)
- dynamicMeshDict (Definition of the dynamic meshing - z. B. staticFVMesh for static meshes)
- thermophysicalProperties (For heat transport)
- /polyMesh/blockMeshDict (Definition of the block mesh and the patches/boundaries)

In /system the entire solver controls are defined. The most common are:

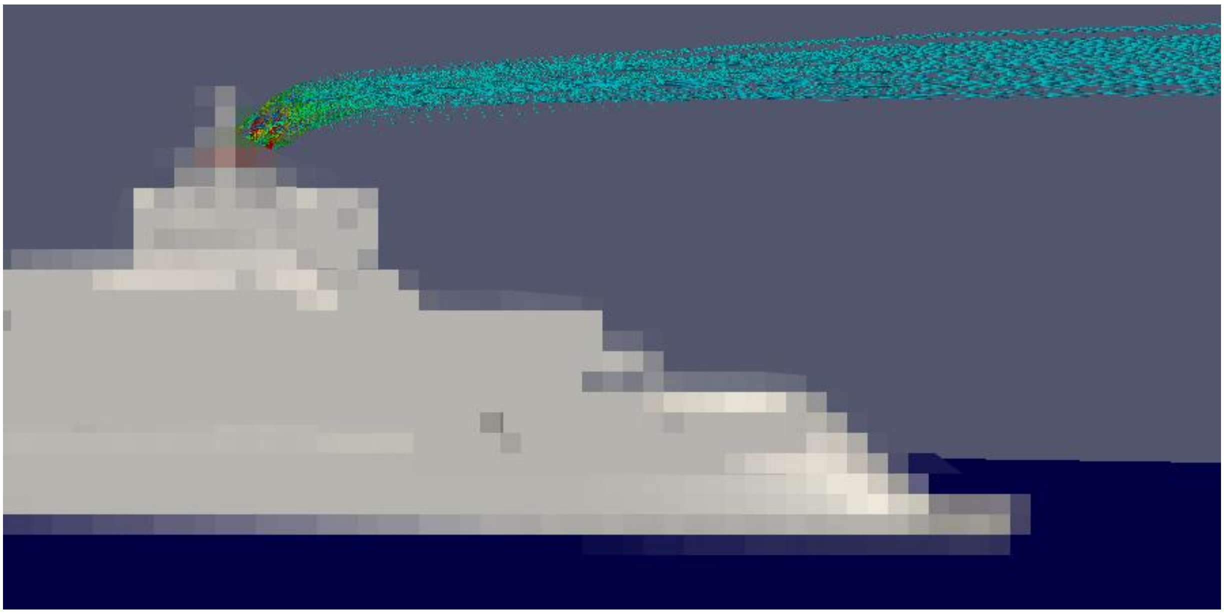
- controlDict (Definition of the solver, time step, writing, extra functions like force output)
- decomposeParDict (Splitting of the domain)
- fvSolution (Definition of solution algorithms, relaxation factors)
- fvSchemes (Definition of the schemes used for each field value)
- setFieldsDict (Phase interface definition before the solving process)
- snappyHexMeshDict (Definition of the meshing of the .stl geometry file)

In/0 the boundary conditions for U, p, α, ε, etc. is defined

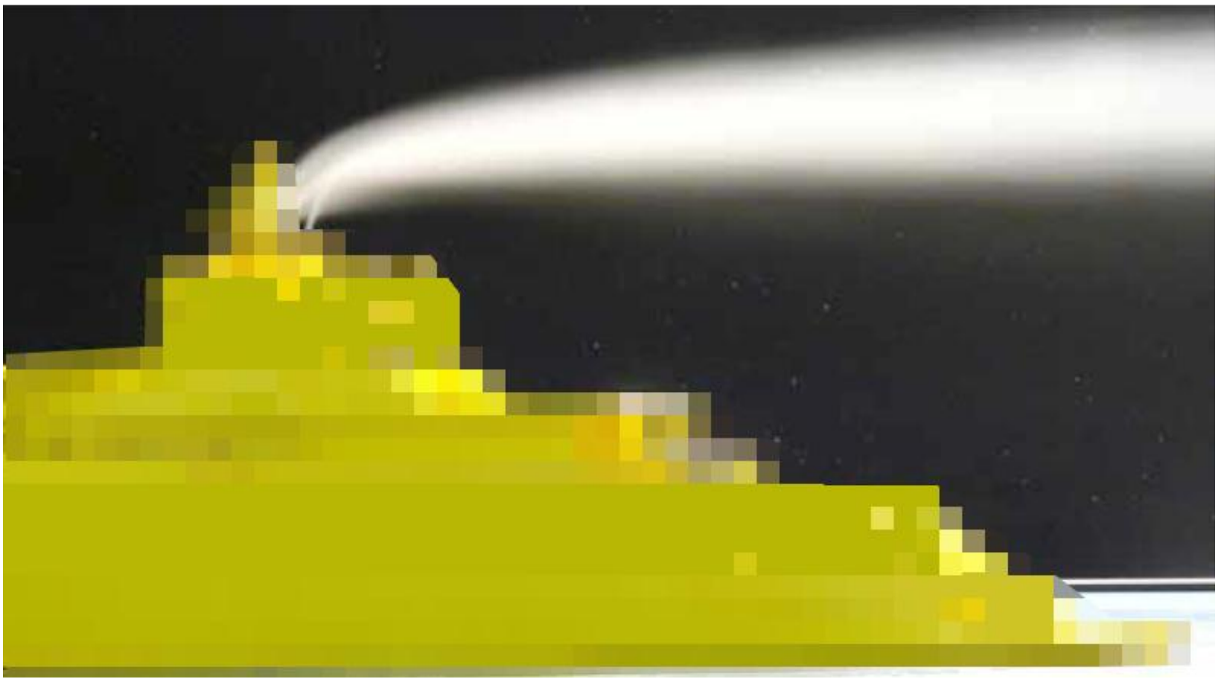
Depending on the writing time starting from the /0 directory the field values are stored in similar folders during the solving process (e. g. /0.1, /0.2, etc.)

Appendix B Aerodynamic validation pictures (MYacht)

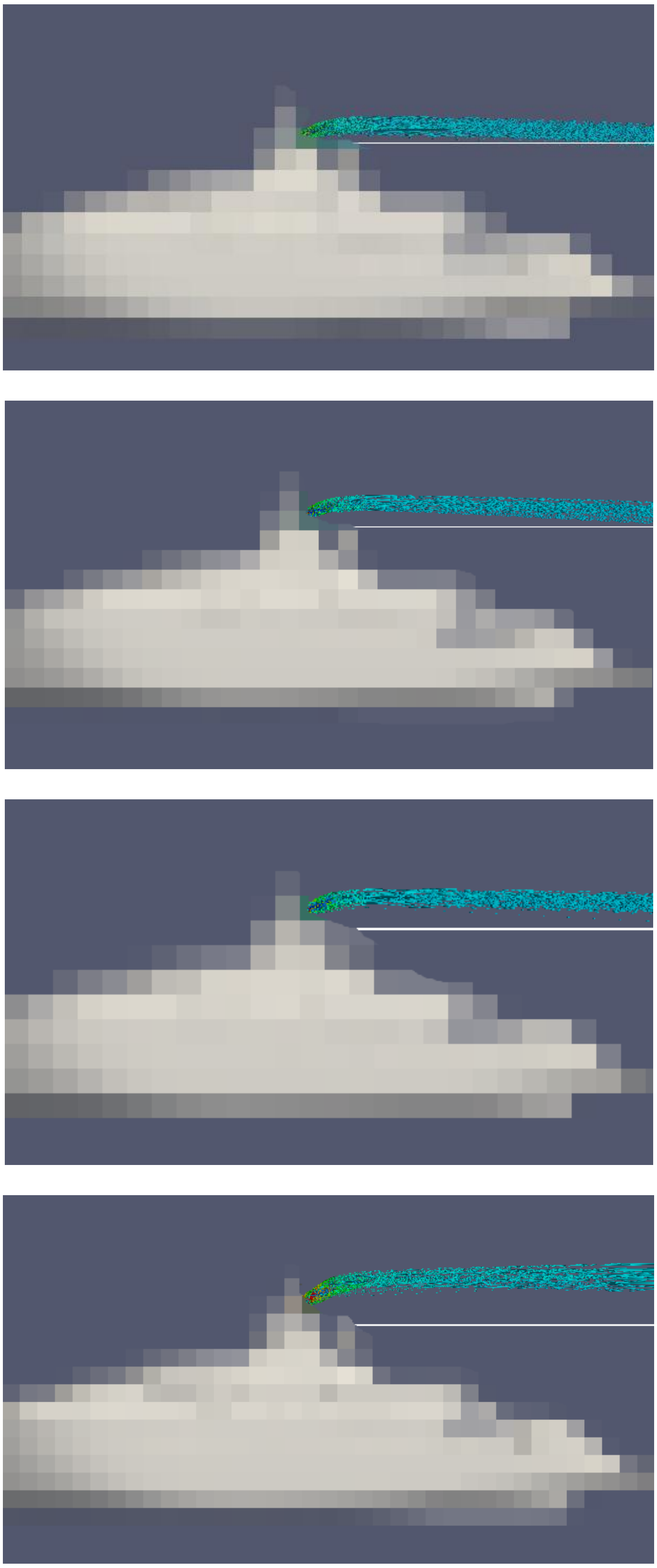
Pipe flow without a wind angle



Figure_apx B.1 - CFD variant #4, 0°

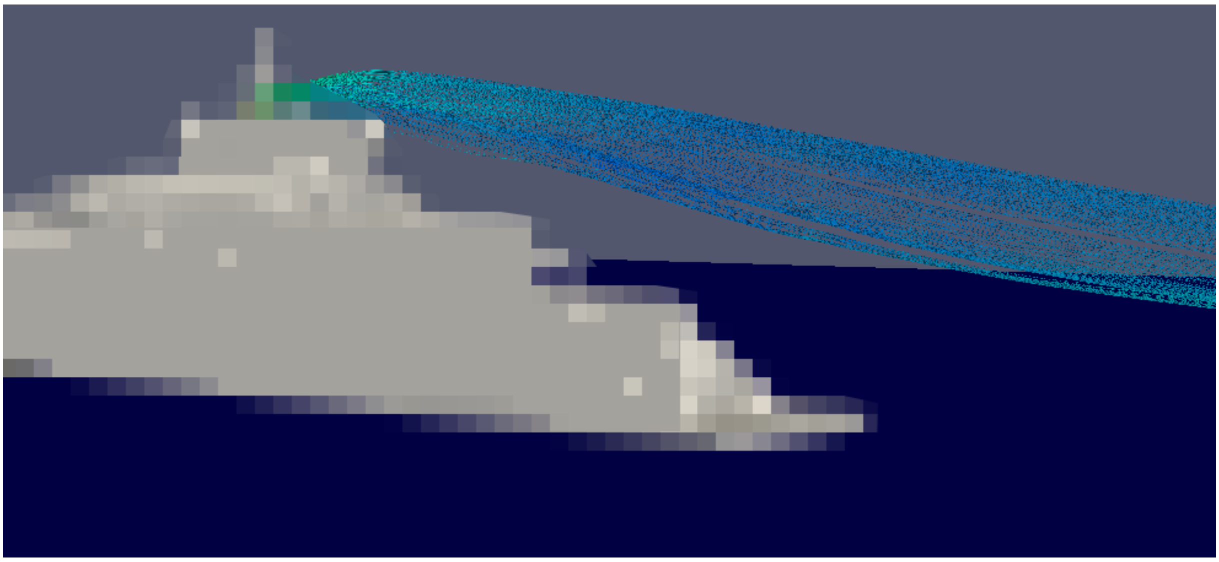


Figure_apx B.2 - Wind tunnel variant #4, 0°

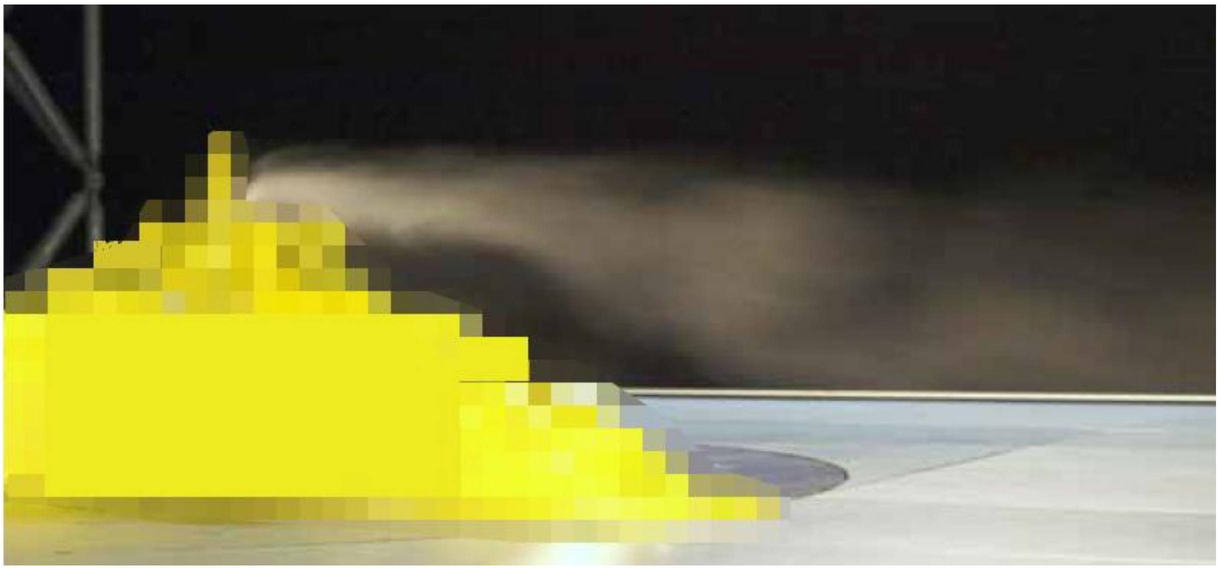


Figure_apx B.3 – Influence of different outlet geometries 0° (Variants #1[top] - #4[bottom])

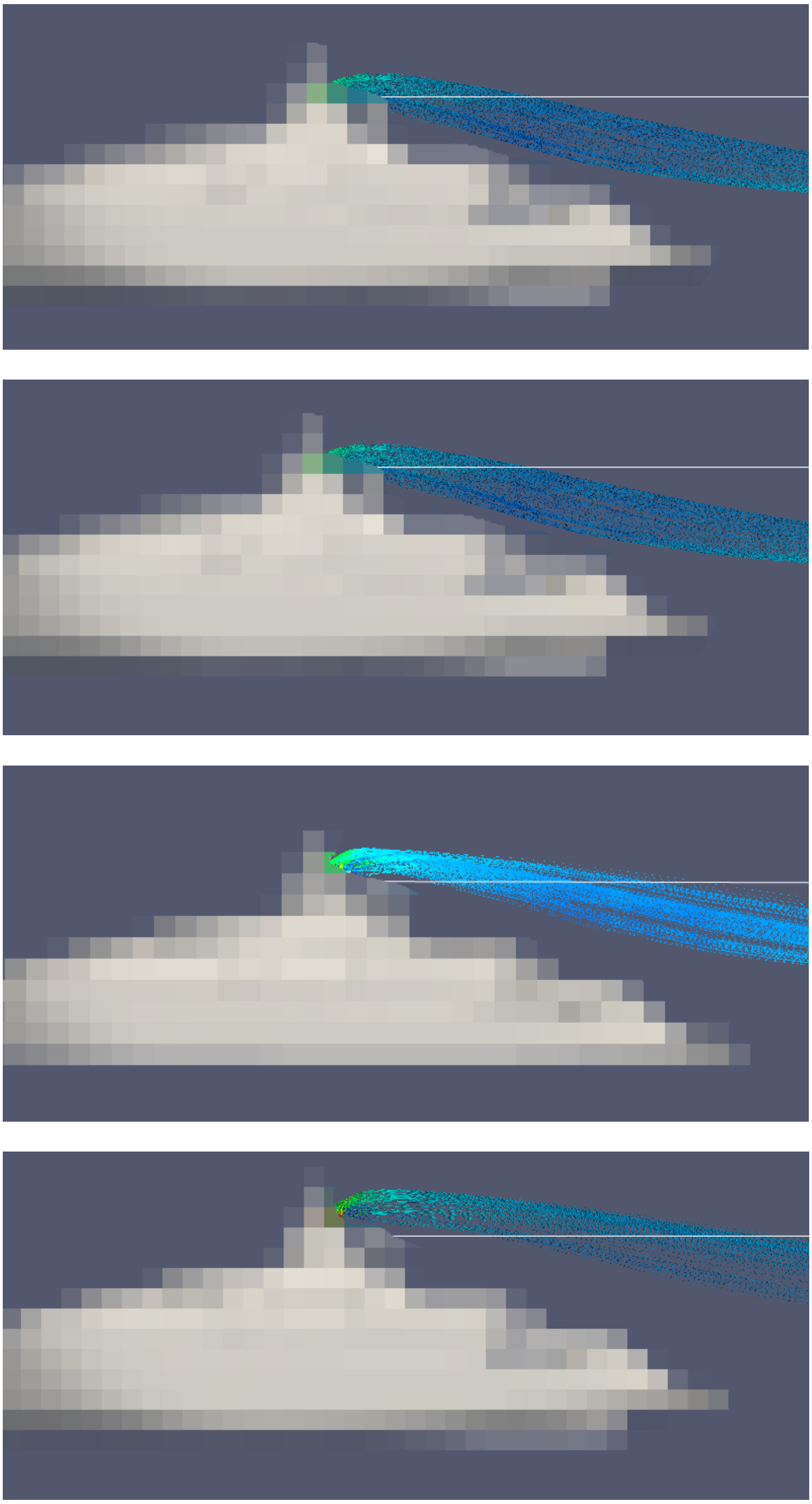
Pipe flow with a small wind angle



Figure_apx B.4 - CFD variant #1, 17°

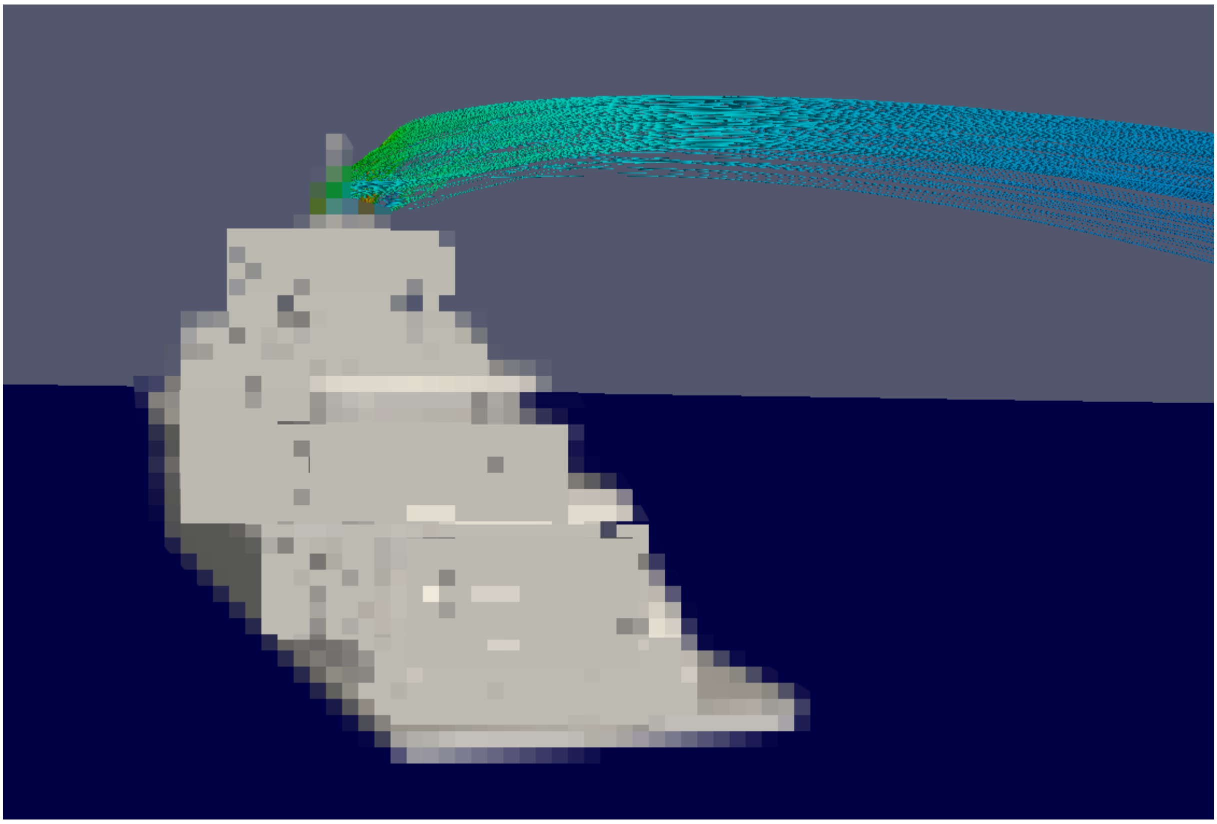


Figure_apx B.5 - Wind tunnel variant #1, 17°

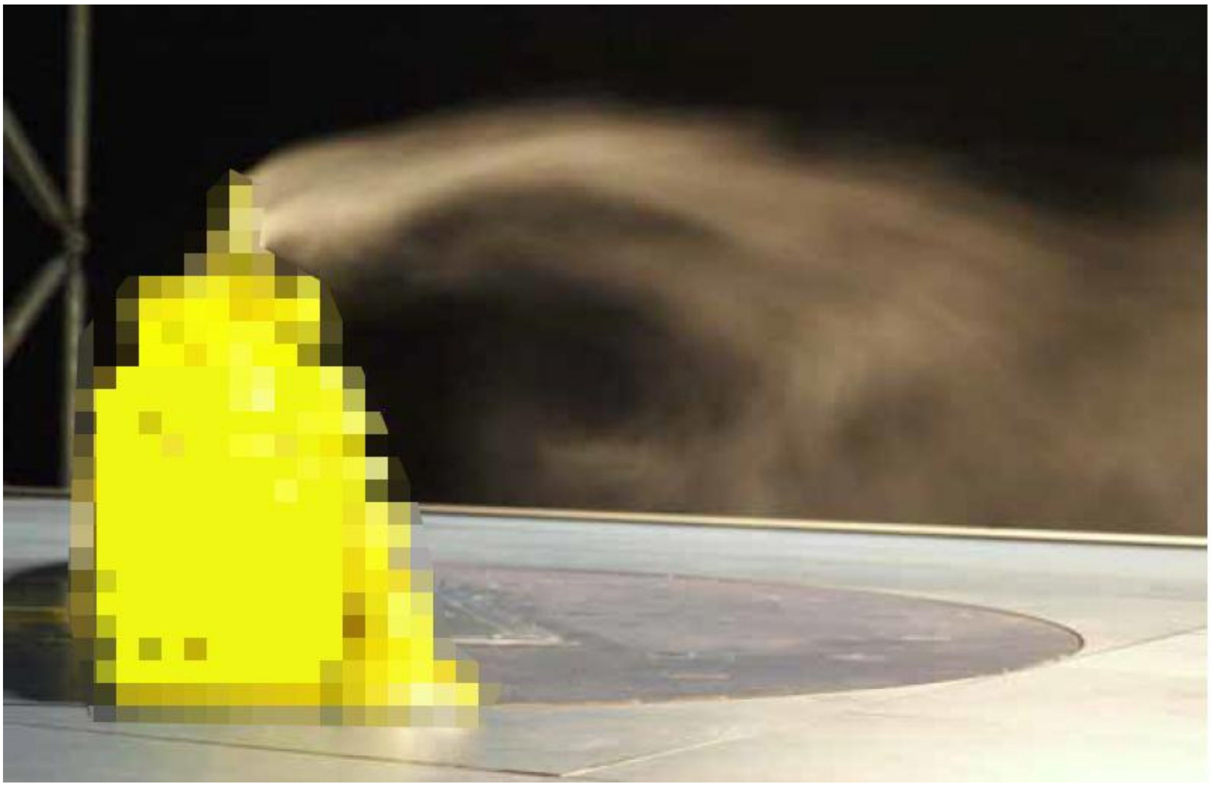


Figure_apx B.6 – Influence of different outlet geometries 17° (Variants #1[top] - #4[bottom])

Pipe flow with a large wind angle

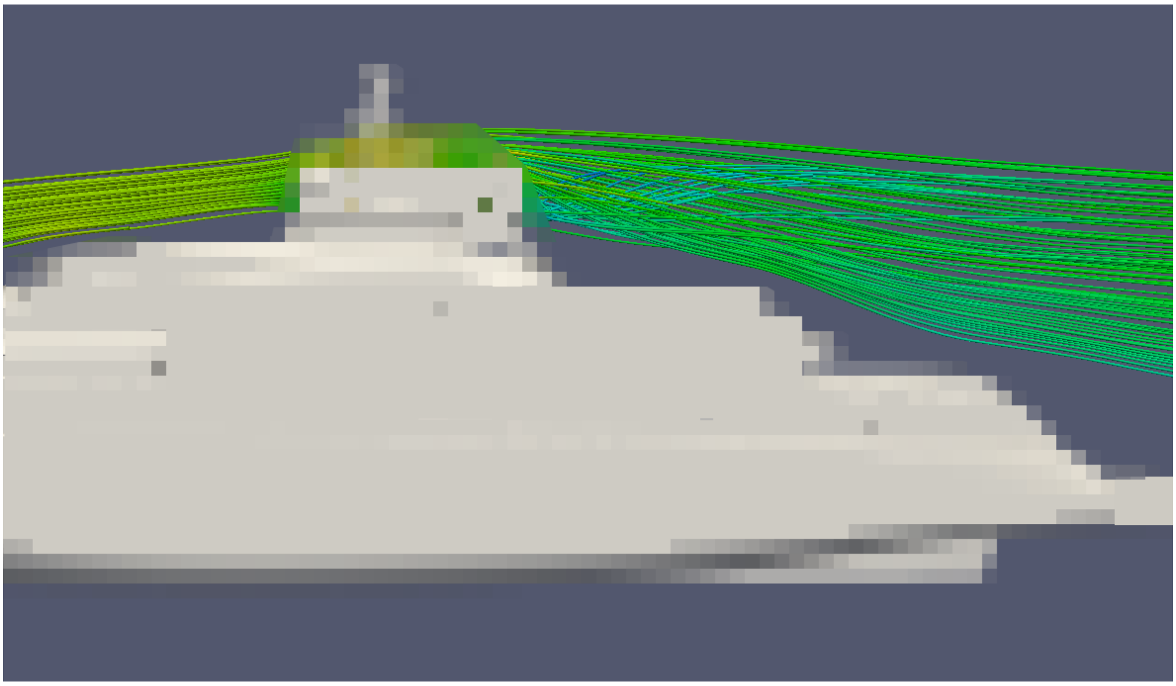


Figure_apx B.7 - CFD variant #3, 51°

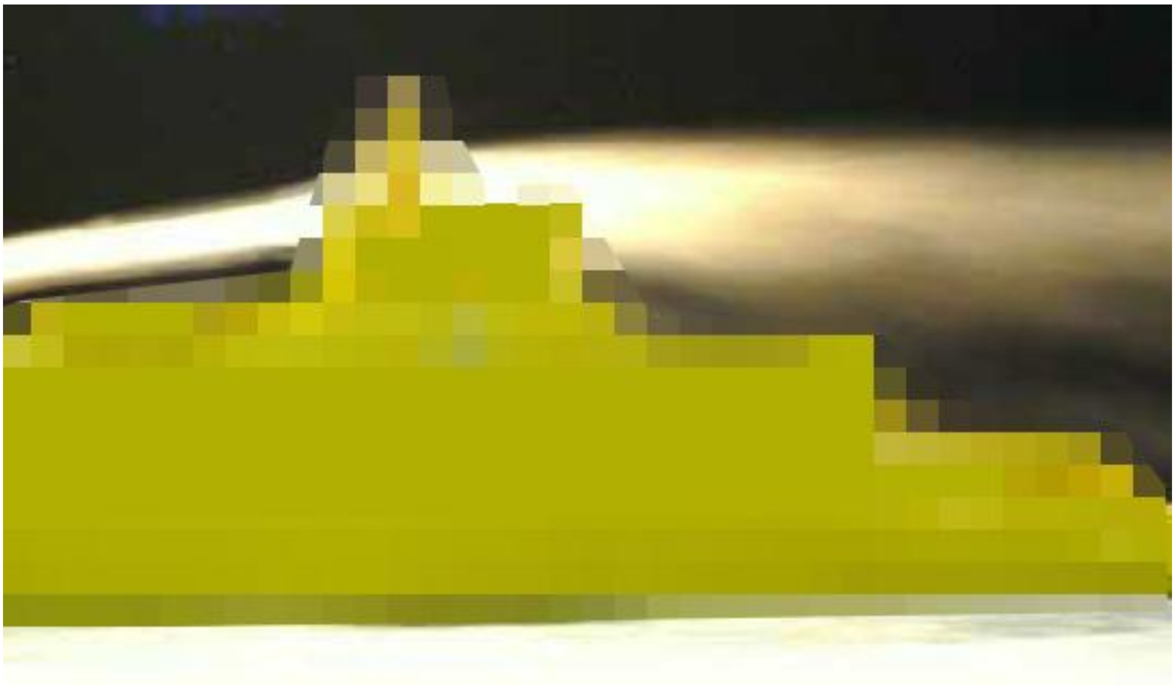


Figure_apx B.8 - Wind tunnel variant #3, 51°

No pipes enabled, wind flow side view

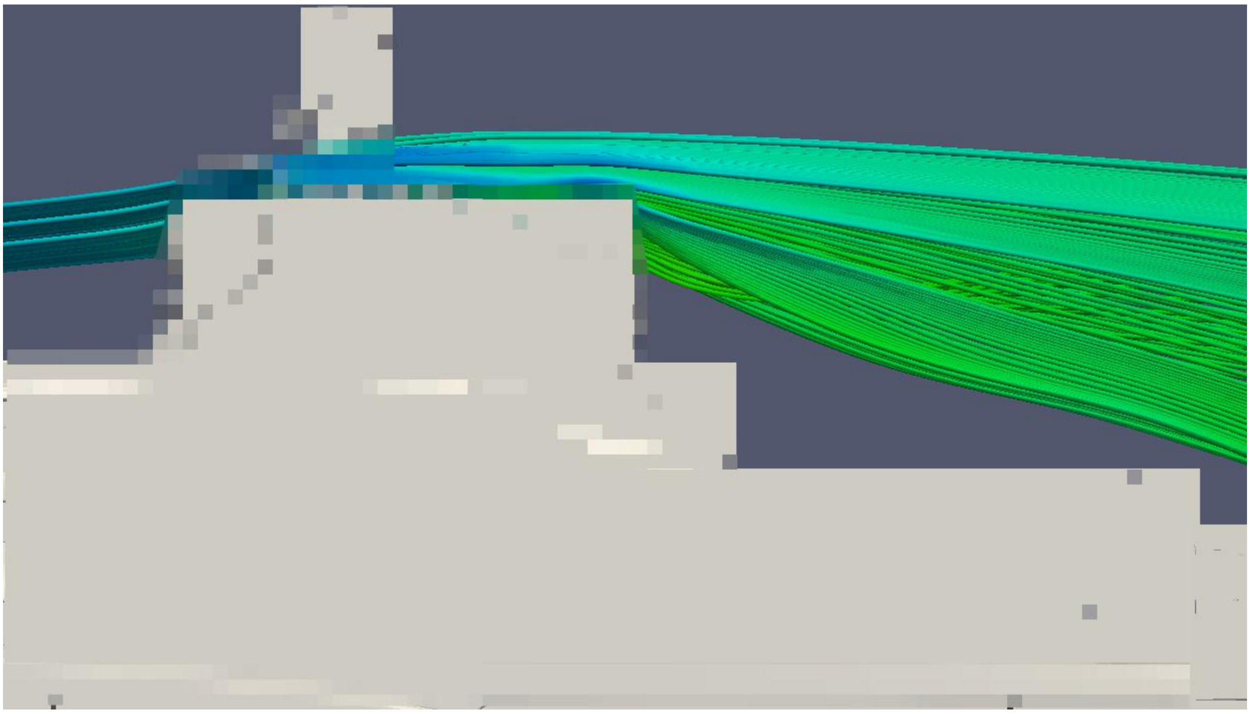


Figure_apx B.9 - CFD variant #4, 0°

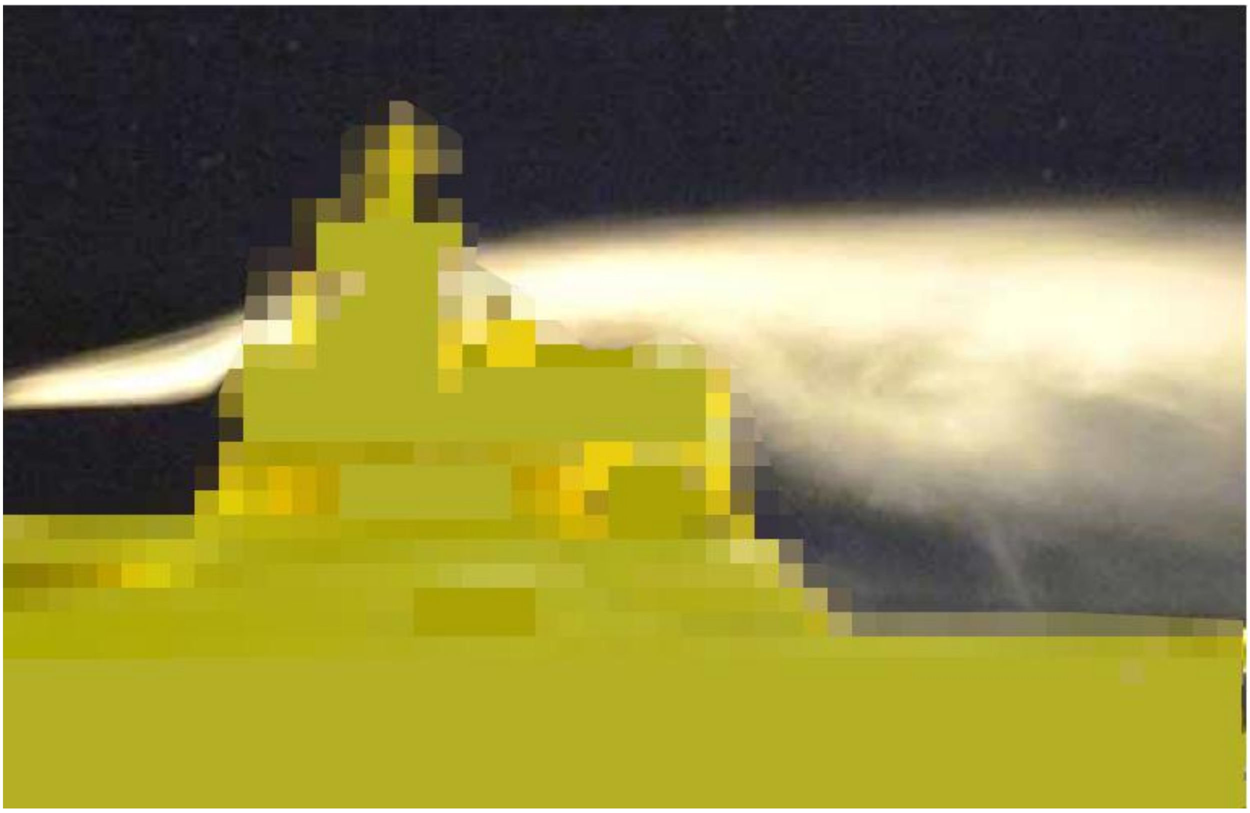


Figure_apx B.10 - Wind tunnel variant #4, 0°

No pipes enabled - closer side view of recirculation zone



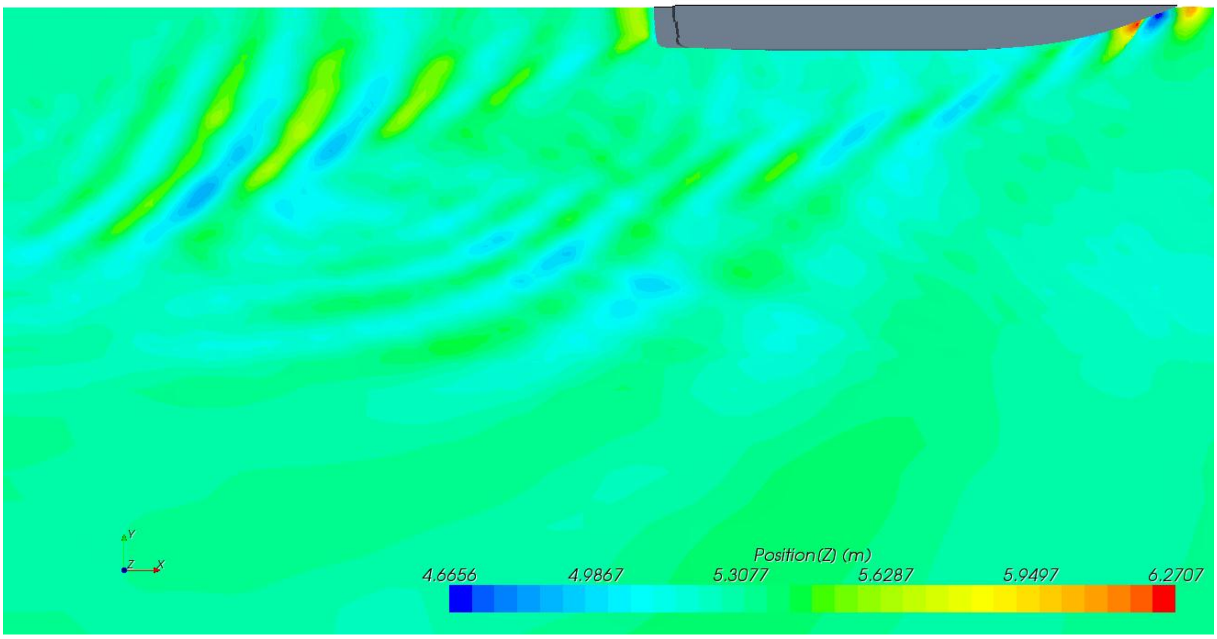
Figure_apx B.11 - CFD variant #4, 0°



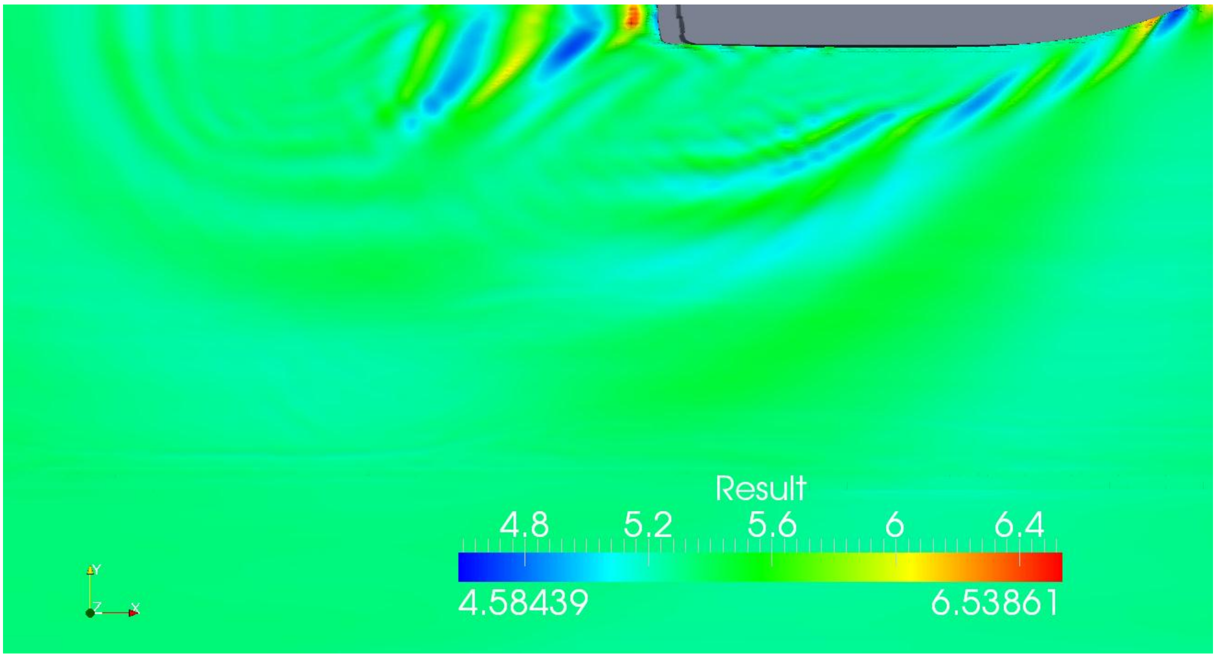
Figure_apx B.12 - Wind tunnel variant #4, 0°

Appendix C Hydrodynamic comparison pictures (Cyacht)

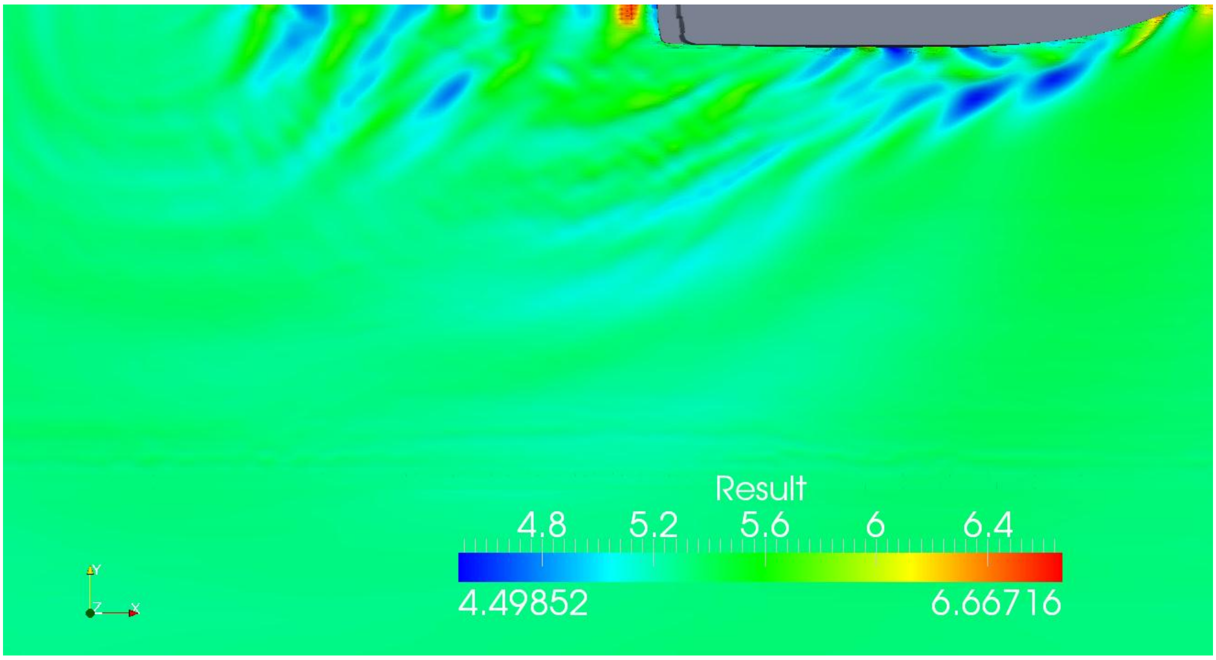
Wave pattern



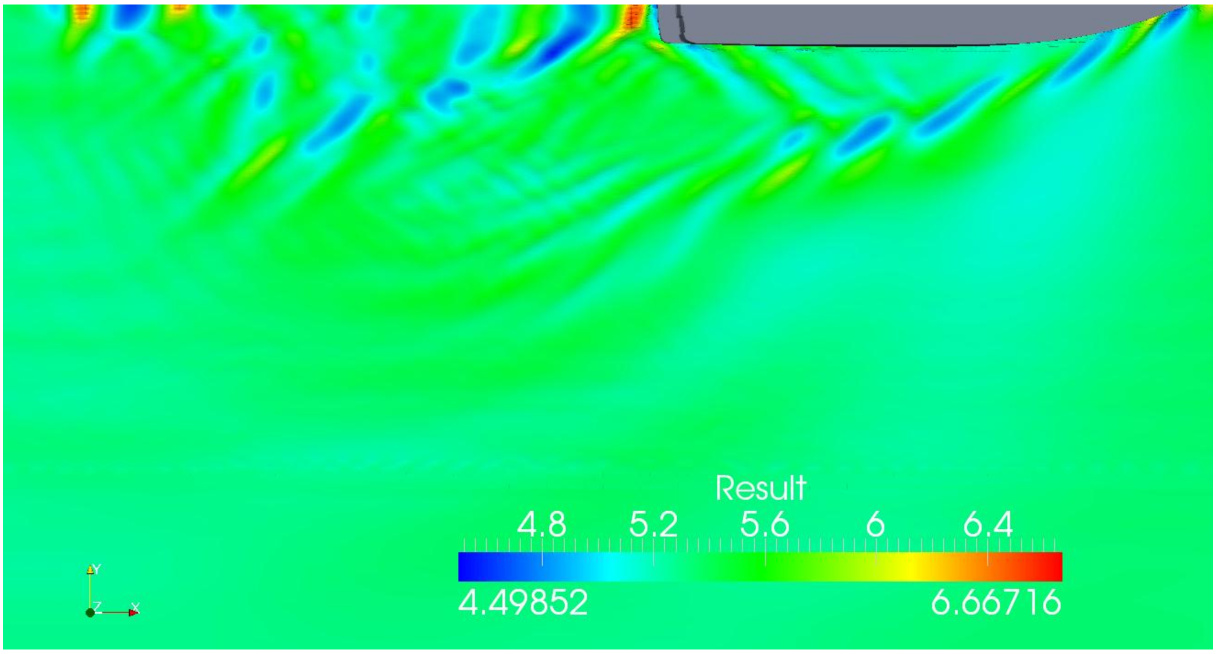
Figure_apx C.1- Wave pattern starCCM+ [m], 6 m/s



Figure_apx C.2 - Wave pattern interFoam [m], 6m/s, laminar, 30s

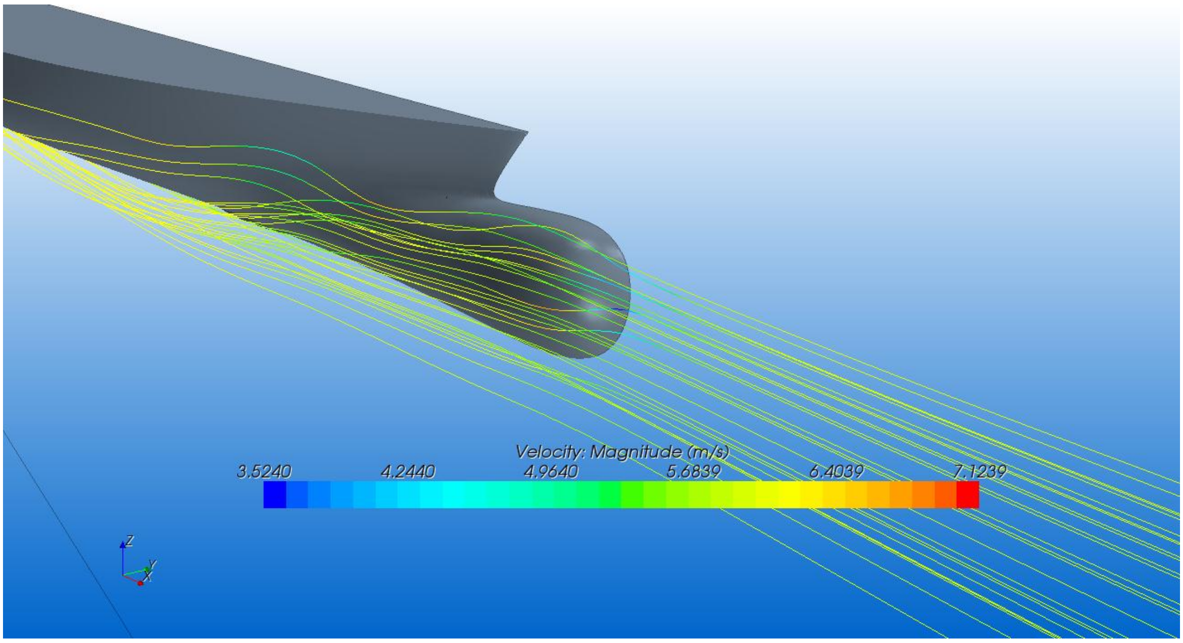


Figure_apx C.3 - Wave pattern interFoam [m], 6m/s, turb, 35s

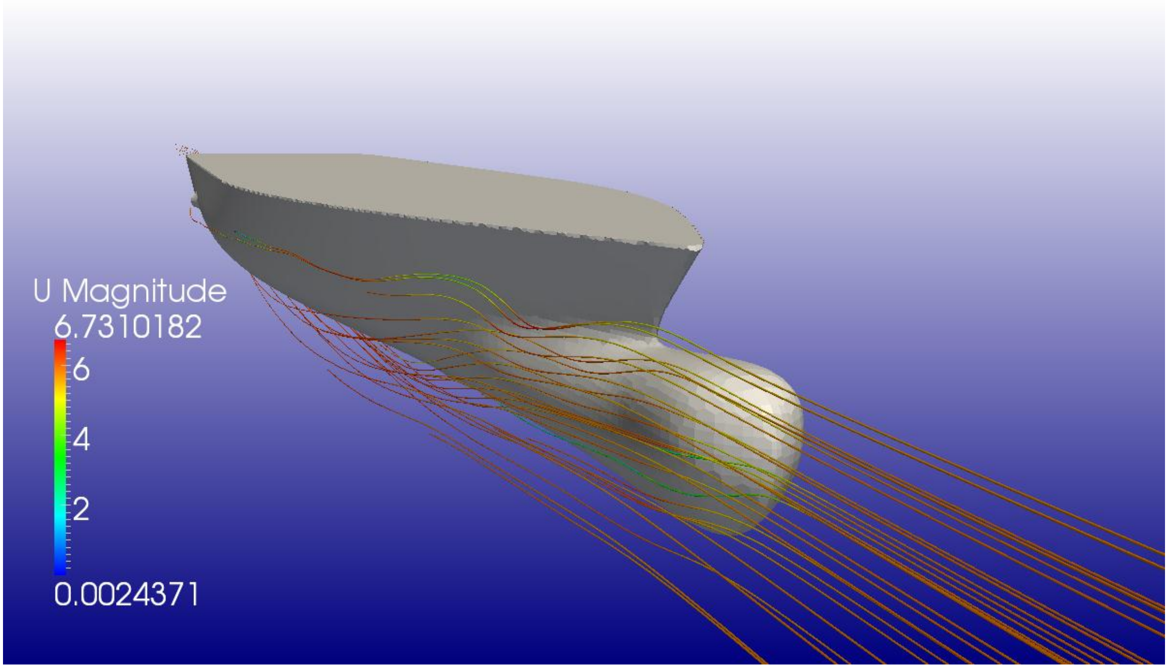


Figure_apx C.4 - Wavepattern with influence of outlet, interFoam [m], 6m/s, 40s

Streamlines bow turbulent

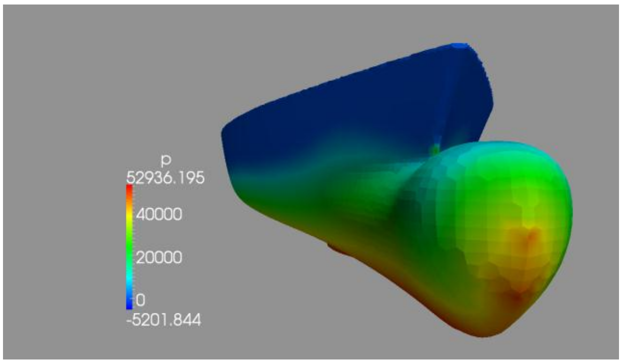


Figure_apx C.5 - Streamlines at bow starCCM+ [m/s]

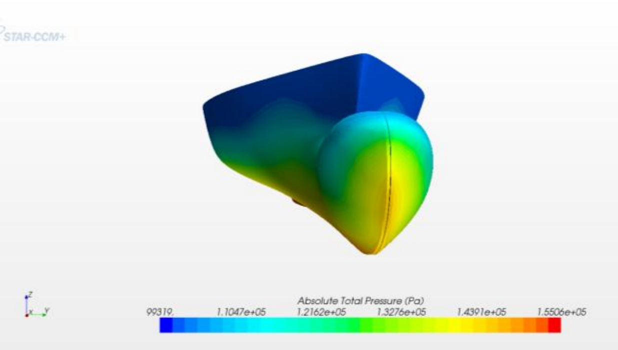


Figure_apx C.6 - Streamlines at bow interFoam [m/s]

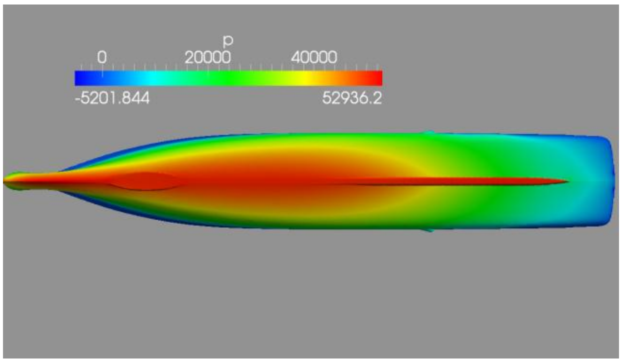
Pressure distribution on bow



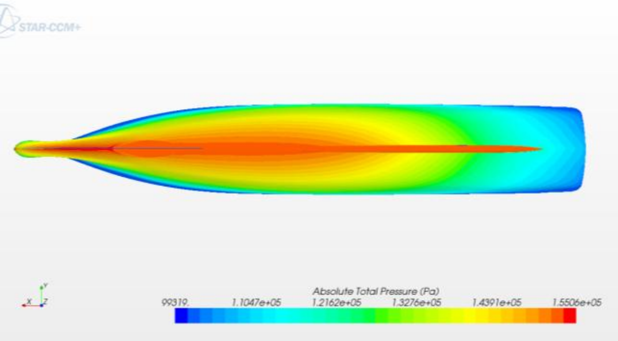
Figure_apx C.7 - Absolut pressure bow [pa], interFoam



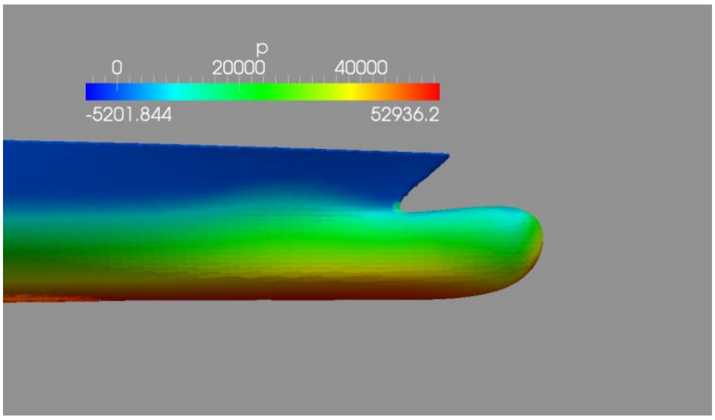
Figure_apx C.8 - Absolut pressure bow [pa], starCCM+



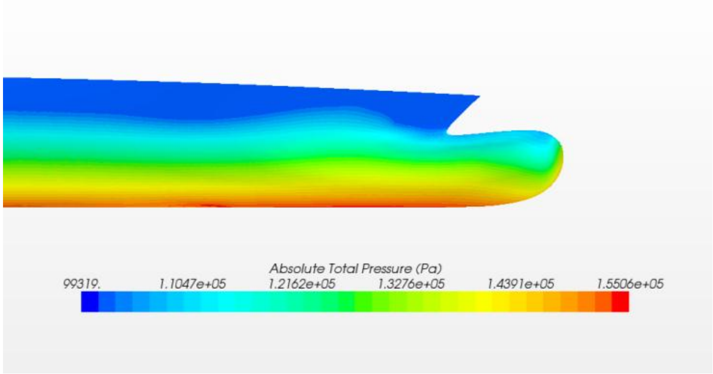
Figure_apx C.9 - Absolut pressure hull [pa], interFoam



Figure_apx C.10 - Absolut pressure hull [pa], starCCM+

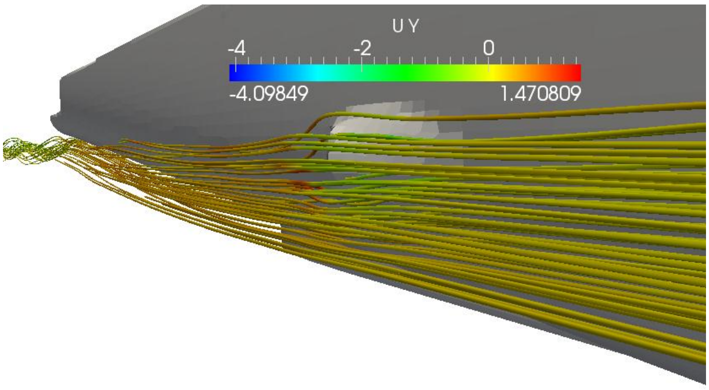


Figure_apx C.11 - Absolute pressure side [pa], interFoam



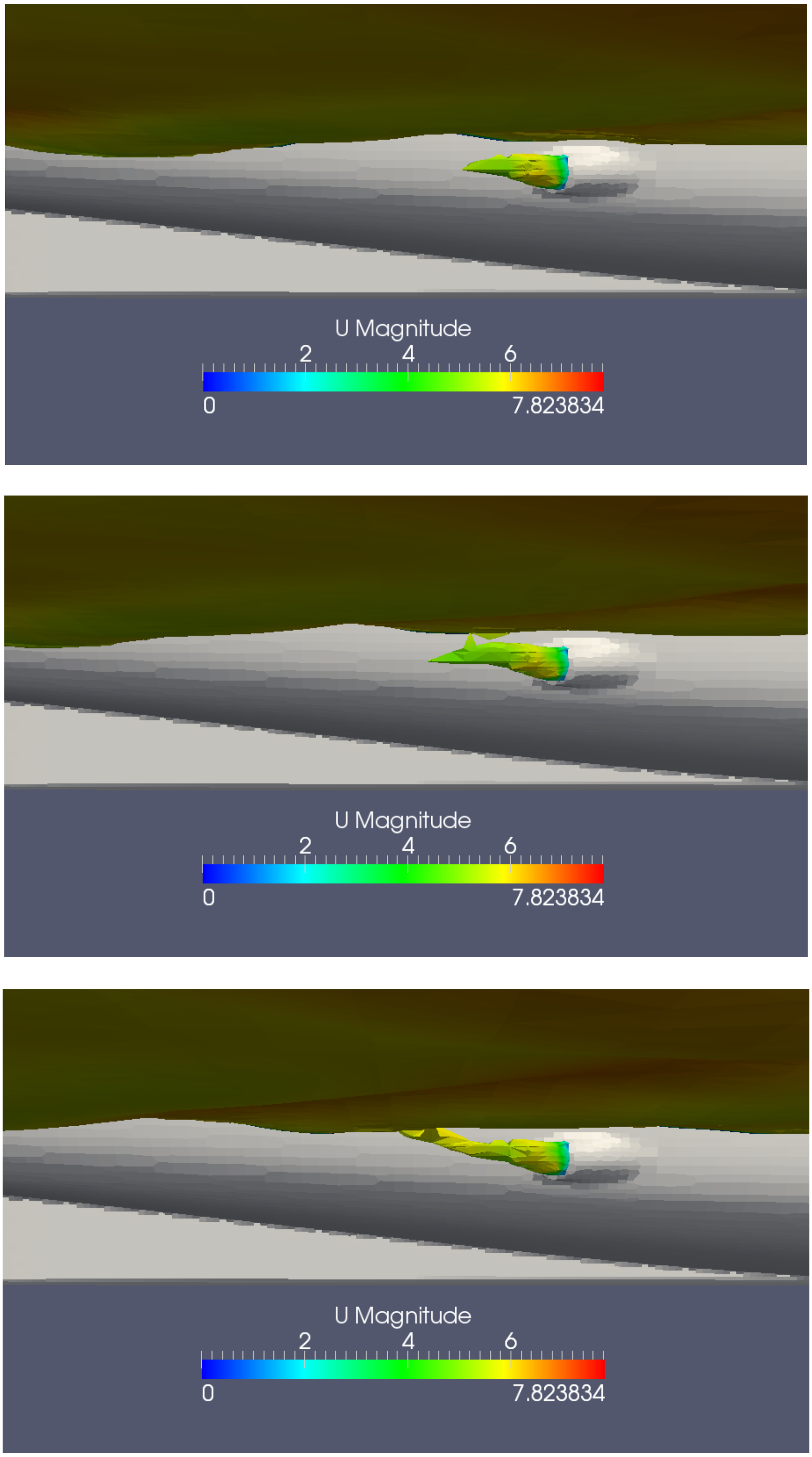
Figure_apx C.12 - Absolut pressure side [pa], starCCM+

Pressure distribution on underwater outlet turb



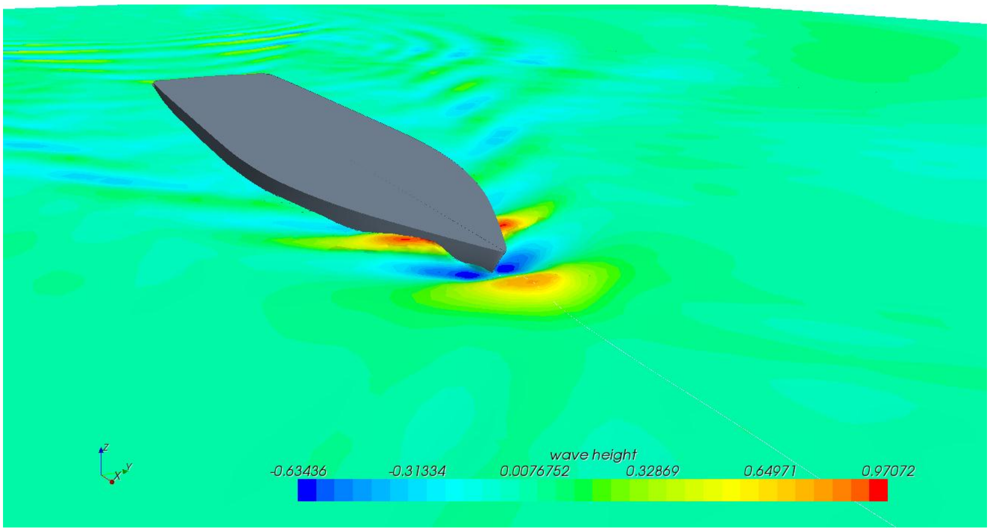
Figure_apx C.13 - Streamlines around exhaust outlet [m/s]

Alpha on underwater outlet

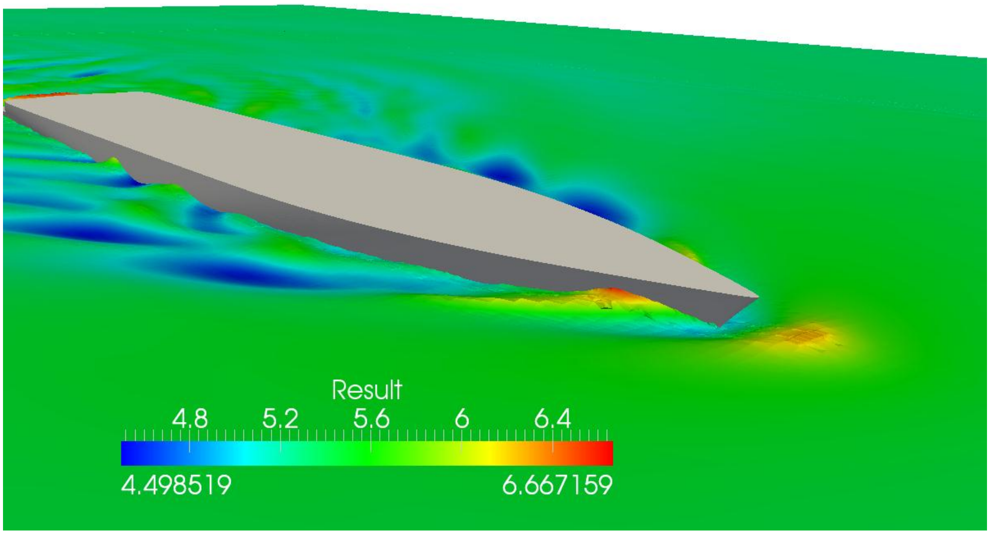


Figure_apx C.14 - Air outflow from the underwater outlet, 4 m/s

Bow wave sideview



Figure_apx C.15 - Wave height elevation [m], starCCM+, 6m/s



Figure_apx C.16 - Wave height elevation [m], interFoam, 6m/s