Tuomas Turunen

Analysis of Multi-Propeller Marine Applications by Means of Computational Fluid Dynamics

School of Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 5.5.2014

Thesis supervisor:

Prof. Timo Siikonen

Thesis advisor:

M.Sc. Timo Rauti

Aalto University
School of Engineering

AALTO-YLIOPISTO
INSINÖÖRITIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Tuomas Turunen

Työn nimi: Monipotkuristen meritekniikan laitteiden analysointi laskennallisen virtausmekaniikan keinoin

Päivämäärä: 5.5.2014 Kieli: Englanti Sivumäärä:8+77

Sovelletun mekaniikan laitos

Professuuri: Virtausmekaniikka Koodi: ENE-39

Valvoja: Prof. Timo Siikonen

Ohjaaja: DI Timo Rauti

Työssä tarkastellaan laskentamenetelmiä, joiden avulla voidaan analysoida useampipotkurisia meritekniikan laitteita. Tavoitteena on luoda laskentatyökalu, jolla voidaan parantaa olemassa olevia laitteita ja arvioida uusien konseptien suorituskykyä ja siten suunnitella hyötysuhteeltaan nykyistä parempia tuotteita. Ensiksi käydään läpi tällä hetkellä saatavilla olevia laskentamenetelmiä ja arvioidaan niiden soveltuvuutta kirjallisuustutkimuksen avulla. Potkurin pyöriminen mallinnetaan MRF-menetelmällä sekä pyörivän hilan menetelmällä, jossa hilan eri osat on erotettu niin sanotulla liukuvalla pinnalla. Turbulenssi mallinnetaan RANS-tyyppisellä kaksiyhtälömallilla SST k-ω. Teoria, johon laskentamenetelmät perustuvat, käydään läpi ja mallien toimivuutta tutkitaan vertaamalla laskettuja tuloksia mittauksiin. Vertailutapauksia on kaksi. Ensimmäisessä tapauksessa on yksi potkuri avovesiolosuhteissa ja toisessa tapauksessa yksikkö, johon kuuluu vastakkainpyörivä potkuri (CRP). Laskenta tehdään avoimen lähdekoodin virtaussimulointiohjelmistolla OpenFOAM-2.2.x, jossa sovelletaan esitettyjä laskentamenetelmiä. Kaikki käytetyt menetelmät toimivat yksittäin, mutta edelleen tarvitaan jatkotutkimuksia, jotta saavutettaisiin tavoiteltu laskentatyökalu useampipotkuristen laitteiden analysoimiseen.

Avainsanat: Vastakkain pyörivä potkuri (CRP), Moving Reference Frame (MRF), OpenFOAM, RANS, SST k-ω, laivan potkuri, avovesikäyrä

AALTO UNIVERSITY
SCHOOL OF ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Tuomas Turunen

Title: Analysis of Multi-Propeller Marine Applications by Means of Computational Fluid Dynamics

Date: 5.5.2014 Language: English Number of pages:8+77

Department of Applied Mechanics

Professorship: Fluid Dynamics Code: ENE-39

Supervisor: Prof. Timo Siikonen

Advisor: M.Sc. Timo Rauti

Computational methods applied for the analysis of marine applications with more than one propellers are studied. The goal is to establish a computational tool that allows an improvement of existing products and the evaluation of new concepts so that new products with an improved efficiency can be designed. First, an overall view of current possibilities is given and their capabilities are evaluated by summarizing results from literature. The moving reference frame (MRF) method and moving meshes based on the sliding grid method are used to account for the effects due to propeller rotation. Turbulence is modelled with the two-equation RANS-model SST k-ω. The theory behind these methods is presented and their performance is evaluated by comparing computational results to measured data. Two test cases are used: The first one is a single propeller in open-water conditions and the other one is a contra-rotating propeller (CRP) unit in a towing tank. The software adopted is the open-source CFD toolkit OpenFOAM-2.2.x that provides the required methods. All methods tested are found to work well, although reaching the ultimate goal of analysing multi-propeller marine applications still requires further studies.

Keywords: Contra-Rotating Propeller (CRP), Moving Reference Frame (MRF), OpenFOAM, RANS, SST k-ω, Marine propeller, Open-water curve.

Preface

This Master's Thesis project has been something I would not have expected when I first accepted the position. I have had the change to learn to know three new cities in Sweden and one in Finland. I have also seen the way of working of an international world-class organisation and a new university other than my own. I was taken as a part of the team and I am sure the contacts that have been formed, both personal and professional, will last after this project.

Out of the multitude of people that have somehow been involved with this work, I owe the most gratitude to Johan Lundberg from the Rolls-Royce team in Kristinehamn, Sweden, and professor Rickard Bensow from the Chalmers university of Technology in Gothenburg, Sweden. I have found our discussions very interesting.

Furthermore, I want to thank two people who have greatly influenced not only this work but my entire professional path so far. The first one is my professor Timo Siikonen who lured me into the world of fluid mechanics by offering me my first academic summer job in the summer of 2010 and has kept me busy with all kinds of projects since then. The other person is Mikko Auvinen who has always had time to walk me through all possible problems during the years and inspired with his own example. Furthermore, I am grateful for the support from all other people who have worked with me at the university.

The final and the most important person I want to mention here is my beloved fiancée Niina who has stood by me all of these years. My life would not be what it is without you.

Otaniemi, 9.3.2014

Tuomas A. I. Turunen

Contents

Symbols

$\mathbf{a}$	Vector
a_P	Diagonal coefficient of matrix A divided by cell volume V
A_P	Diagonal coefficient of matrix A
A_N	Off-diagonal coefficient of matrix A
C_μ	Turbulence model coefficient
C_p	Pressure coefficient
Co	Courant number
D	Propeller diameter
$\hat{e}$	Unit vector
E	Coefficient in the log-law
$\mathbf{F}$	Force
G	Turbulence generation
I	Turbulence intensity
J	Advance coefficint
k	Turbulence kinetic energy
K_Q	Torque coefficient
K_T	Thrust coefficient
m	Mass
n	Number of propeller revolutions per second
p	Pressure
p_0	Ambient pressure
$\mathbf{q}$	Source term
Q	Torque
$\mathbf{r}$	Position vector
R	Reynols stress tensor
Re	Reynolds number
S	Strain rate tensor
$\mathbf{S}_f$	Face area vector
T	Thrust
u^+	Dimensionless velocity
u_τ	Friction velocity
V	Volume
$\mathbf{U}$	Velocity
w	Weight factor
y^+	Dimensionless wall distance

α	Diffusion coefficient
$\beta^\star$	Turbulence model coefficient
β_1	Turbulence model coefficient
δ_{ij}	Kroenecker delta
η	Efficiency
γ	Blending factor
κ	Von-Karman constant, condition number
λ	Lambda viscosity, eigenvalue of a matrix
μ	Dynamic viscosity
ν	Kinematic viscosity
ω	Rotation vector, Specific dissipation of turbulence kinetic energy
ϕ	General variable
ρ	Density
σ	Cavitation number
τ	Stress tensor

Operators

$\nabla \mathbf{A}$	Gradient of vector $\mathbf{A}$
$\nabla \cdot \mathbf{A}$	Divergence of vector $\mathbf{A}$
$\nabla \times \mathbf{A}$	Curl of vector $\mathbf{A}$
$\mathbf{A} \cdot \mathbf{B}$	Inner product of vectors $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{AB}$	Outer product of vectors $\mathbf{A}$ and $\mathbf{B}$
$\frac{\mathrm{D}}{\mathrm{D}t}$	Total derivative in time
$\frac{\partial}{\partial t}$	Partial derivative in time
$\sum_i$	Sum over index i

Abbreviations

AMI	Arbitrary Mesh Interface
CFD	Computational fluid dynamics
CRP	Contra-rotating propeller
GAMG	Geometric-algebraic multi-grid
ITTC	International Towing Tank Conference
`MRF`	Moving Reference Frame
PBiCG	Preconditioned Biconjugate Gradient
RANS	Reynolds Averged Navier-Stokes
R&D	Research and Development
RRHRC	Rolls-Royce Hydrodynamic Research Center

1 Background

1.1 Motivation

The motivation for this study is the wish to introduce modern tools into the analysis of fluid dynamics problems related to products of Rolls-Royce Marine. A special interest is in the analysis of multi-propeller devices such as the current *Contaz* product that is a thruster unit with two propellers rotating in opposite directions (contra-rotating propeller, `CRP`) manufactured by Rolls-Royce Oy Ab in Rauma, Finland.

Figure 1: *A contra-rotating propeller (CRP) unit manufactured by Rolls-Royce.*

One benefit of the contra-rotating concept in comparison to a single propeller is the fact that it reduces the loading on each single blade improving efficiency and cavitation performance. In particular, this helps in situations where the propeller size is limited for some reason (like shallow water) but still a given thrust is needed that would otherwise lead to a too highly loaded single propeller.

Another benefit of the CRP-concept comes from the fact that the second propeller reduces the tangential velocity component produced by the first propeller. Reducing the swirl directly contributes to efficiency, since the rotational movement of water behind the ship produces no thrust and is thus a pure loss [1, Ch.10.7].

The current concept is certainly not the hydrodynamically best possible solution and it could be improved. Furthermore, there is a need to evaluate the performance of new concepts such as a pulling CRP unit to see, if they allow further improvements in efficiency and, at the same time, evaluate if they are otherwise feasible. Currently there are no large-scale pulling CRP units in the market and it might be that the steering moments become too large for the overall costs to be reasonable. To be able to improve the current products and to evaluate the feasibility of new concepts, a tool for predicting the hydrodynamic performance of multi-propeller devices is needed. Establishing such a tool is the main goal of this work.

Historically, problems in propeller hydrodynamics have been approached through so called potential methods that are based on several simplifications. A discussion of such methods is provided for example in Reference [2]. At present, a standard analysis by means of computational fluid dynamics (CFD) includes full Navier-Stokes equations which is also pursued in this work. There is a range of software available for the task, both commercial codes with licensing costs and open-source codes. Since the computations are expected to become quite large and need parallel

processing, the open source CFD software package, `OpenFOAM`, was chosen for this work in order to avoid licensing costs. `OpenFOAM` is licensed and distributed by the OpenFOAM Foundation and developed by OpenCFD Ltd. [3].

This work consists of a general discussion of propeller analysis in Chapter 1.2 which is followed by an overview on relevant computational and experimental methods based on literature in Chapters 1.3 and 1.4. After the overview, the computational methods used in this work are presented in more detail in Chapters 2 and 3 and the rest of the work documets the performed CFD-analyses. An outline of the computations will be presented in Chapter 1.5.

1.2 Propeller Performance

The simplest way to analyze propeller performance is to measure the thrust (T) produced by the propeller and the torque (Q) used to drive the propeller. Efficiency can be defined as the ratio of the acquired power in thrust production to the power used to drive the propeller.

In the marine context, thrust and torque are usually given as non-dimensional measures as a function of a non-dimensional speed called the *advance coefficient*. The definitions of thrust- and torque coefficients, K_T and K_Q, and the advance coefficient, J, are given as

$$K_T = \frac{T}{\rho n^2 D^4} \tag{1}$$

$$K_Q = \frac{Q}{\rho n^2 D^5} \tag{2}$$

$$J = \frac{U}{nD}. \tag{3}$$

Above, propeller diameter is denoted as D, n is the number of propeller revolutions in one second, ρ is the water density and U is the advance speed, for example the speed of a ship. The coefficients are found by applying dimensional analysis and assuming that free surfaces have no effect on the propeller performance. Under such conditions the coefficients are theoretically identical for all geometrically similar blade forms.

The definition of efficiency can be written as

$$\eta = \frac{T\,U}{2\,\pi\,n\,Q} \tag{4}$$

$$= \frac{K_T}{K_Q}\,\frac{J}{2\,\pi}$$

The thrust and torque coefficients (K_T and K_Q) being similar for *all* cases with an identical blade shape is not entirely true. There are several factors that affect propeller performance even with a constant advance coefficient J. One is turbulence

around the propeller which is, to a large extent, a function of the Reynolds number (Re)

$$Re = \frac{UD}{\nu} \tag{5}$$

where ν is the kinematic viscosity of water. With increasing propeller dimensions the Reynolds number increases which will generally lead to a higher turbulence level. Above sufficiently high Reynolds numbers when most of the blade is surrounded by a turbulent boundary layer, the effect is probably not very significant. The difference between a model scale and a full scale propeller may, however, be significant. Also the turbulence of the incoming flow plays a role. Since propeller computations and measurements are conducted in model scale with a considerably lower Reynolds number than those of real world applications in their operating conditions, Reynolds number effects lead to an uncertainty the computations. The uncertainty can be decreased by running simulations or making experiments with full scale geometries. Both approaches are, however, expensive.

Another factor affecting force prediction is cavitation. It depends on the surrounding pressure level and propeller loading. A non-dimensional number related to cavitation, the *cavitation number*, is defined as

$$\sigma = \frac{p_0 - p_s}{\frac{1}{2}\rho U^2} \tag{6}$$

where p_0 is the static pressure on the shaft center line and p_s is the vapor pressure at the ambient temperature. Cavitation has a relatively small influence on propeller performance even in quite large extents. At a point called thrust breakdown, however, when cavitation increases above a certain level, it has a considerable negative effect on thrust [4]. Thus, at moderate values of the advance coefficient, cavitation is not expected to impose considerable uncertainties into the computations.

1.3 CFD Methods

The flow around contra-rotating propellers is always turbulent and unsteady due to the interaction between propellers. There are ways to reduce an unsteady problem into a steady one and thus save computational time. However, at the same time simplifications are made and the feasibility of these simplifications depends on their ability to depict the phenomena of interest [5]. In the following, an overview of possible CFD-methods that might be used for tasks described above is given with a literature review. The methods are discussed on a general level and are not specifically related to `OpenFOAM`.

According to [6] there are basically three ways to treat the propeller rotations when the propeller is geometrically resolved: the *sliding plane*, the *Moving Reference Frame (`MRF`)* and the *mixing-plane* approaches. The most computationally demanding out of these but also the most accurate is the sliding plane approach. Propeller movement is included by actually moving different parts of the mesh. Since some parts rotate and some parts remain unchanged the mesh will, in general, have discontinuities. In order to handle the communication over the discontinuity, an

interface is needed. The method is, by definition, time-dependent and as such also computationally relatively expensive [6].

The moving reference frame (MRF) is a steady-state method in which different zones are given different rigid body motions (such as a rotational speed) without actually moving the mesh. The otherwise time-dependent flow is turned into steady-state by writing the governing equations as observed from a moving or, in other words, relative coordinate system that follows the body movement. In such a coordinate system there clearly seems to be no movement. A thorough description of the MRF method in OpenFOAM will be given in Chapter (2.3). Since the geometry remains fixed throughout the simulation the method is also known as the *fixed rotor*-approach. The MRF approach can be used when the transient interactions between adjacent zones is weak but not if the interactions need to be resolved [6]. Yet another name for the MRF method is the *quasi-steady* method.

The mixing plane approach also leads to a steady-state solution. The fluid domain is divided into zones and terms are added and modified like in the MRF method. All regions with different motions are treated as independent problems and information is passed between adjacent regions by averaging information in the circumferential direction on zone boundaries. The averaging removes any oscillations otherwise present in the solution [6].

In the following three subsections, experiences conserning these three methods, their feasibility and usage, are presented. The applicability of the methods is discussed separately in case of thruster units, CRP units and problems with a non-uniform inflow to the propeller. The literary results are reflected on the current study in order to make a decision on what methods will be applied in the computational part of this work.

1.3.1 Propeller-Structure Interaction

In a study of Sànchez-Caja et al. [5], the mixing-plane- and the MRF methods were compared to unsteady computations with a sliding plane in the case of a ducted single propeller with a rudder. They found out that propeller forces were better predicted with the MRF approach than with the mixing-plane approach. Grid size was approximately $1 \cdot 10^6$ cells in the coarsest and $9 \cdot 10^6$ cells in the finest grid. The MRF method predicted propeller forces that were within one percent of the time-accurate results. However, the mixing plane approach performed better at predicting efficiencies producing results within half a percent of the time accurate results. Also rudder forces were better predicted with the mixing plane approach. They state that, if the simpler models (mixing plane or MRF) are used, it is important to ensure that there is no significant interaction between the mesh regions that are separated by an interface. In particular, there should be no solid walls downstream of the interface that block the flow [5].

The above mentioned implies that the interface should be set far enough from the propellers which is impossible in CRP cases. Thus it does not encourage to use either of the mixing plane or the MRF methods in CRP cases. It also implies that the precense of the thruster unit downstream of the propellers in the case of a pulling

CRP will involve difficulties.

Guo et al. [7] studied the interaction of a single pulling propeller and a pod using the mixing plane and sliding plane approaches with the RANS-solver FLUENT. The mesh size was approximately $3.6 \cdot 10^6$ cells. They concluded that the forces and moments of a single blade vary by about 7% and 6%, respectively, during one propeller revolution. The variations are due to the pod located downstream of the propeller. The total forces and moments on the propeller vary, however, by only 0.6% and 0.7%, respectively. Guo did not validate the results, however, due to the lack of experimental data. He pointed out one problem with the mixing plane. Part of the inbound flow to the strut is averaged on the mixing plane and part is not which leads to physically debatable flow conditions at the strut [7].

Reflecting on the present study, Guo's results imply that the mixing plane will decrease the reliability of forces on the strut at least in the case of a pulling configuration. Sànchez-Caja et al. [5] yielded better rudder forces with the mixing-plane than the `MRF` approach but in a case of a ducted propeller. Thus, the two findings are not in contradiction. Furthermore, the presence of solid structures even on the downstream side of the propeller will affect forces on a single blade which confirms the unsteadiness of all CRP cases where there are always several bodies interacting with each other.

Sànchez-Caja et al. [8] studied a podded propeller construction constisting of a single pulling propeller, pod and a strut using the sliding mesh and mixing plane approaches. They also computed the propeller in open-water conditions. In the latter study they obtained thrust- and torque coefficients as well as efficiencies within 1.5% of results measured at MARIN. Their results, however, predicted by 4.5% too high thrust- and torque coefficients as compared to measurements conducted at VTT. Efficiency was, however, predicted within an error of 0.2%. The differences between the two measurements are discussed in the report and they can at least partially be attributed to different hub geometries, Reynolds numbers and measuring equipment. In their study, one time increment Δt corresponds to 0.625° propeller rotation.

The propeller performance in the mixing plane thruster computation gave the thrust within 8.5% and efficiency within 6.5% of measured values. The reasons for the deviation are also discussed. The grid near the propeller was not fine enough, about one percent is attributed to the difference between the time dependent and quasi steady methods. Less than 2% error is estimated to be due to simplifications in the computational model. The time accurate computation was able to predict the forces on the strut and pod much better than the mixing plane approach. They also found out that the time dependent computation predicts lower pressure peaks than the mixing plane approach. This is addressed to the lack of shed vortices in the steady computation which increases loads.

1.3.2 Contra-Rotating Propellers

Wang et al. [9] computed a contra-rotating propeller in open water conditions. They used a time-accurate RANS-based method with the sliding plane approach and

tested the influence of time-step size and three two-equation turbulence models on the thrust- and torque coefficients. They tested three different time step sizes corresponding to propeller rotations of 0.25°, 0.50° and 1.0° in a time step. All results differ from each other and the accuracy improves with a decreasing time increment. This implies that not even 0.25° is a sufficiently small time step to capture the time-accurate history of the thrust- and torque coefficients without some error from the time discretization. They ran computations with two CRP combinations. In the first set, both propellers had the same number of blades and, in the second one, the blade numbers were different. The effect of time step size was found to be weaker in the case of different blade numbers than when both propellers have the same number of blades. The turbulence models tested were the standard k-ϵ, the RNG k-ϵ and the SST k-ω model. Out of these three models, the SST k-ω model gave the most accurate solutions while the standard k-ϵ model was least accurate. The errors of the predicted time-averaged thrust- and torque coefficients compared to experimental values were of the order of 5%. The prediction of the front propeller was found to be more accurate than the prediction of the aft propeller.

Feng et al. [10] studied the open-water performance of a contra-rotating propeller with FLUENT. They studied the performance of the front propeller in three ways. One steady-state computation was made with a single blade (without aft propeller) so that periodic boundary conditions were imposed on the sides of the blade domain and the `MRF` approach was used. The second computation was conducted with a single complete front propeller without including the aft propeller. The case was computed as a time accurate analysis where the sliding mesh approach was used to model the rotation. In the third computation, two propellers were computed using the sliding mesh approach. The time step was chosen to correspond to a front propeller rotation of 2° per time step. They kept the rotational speed of the aft propeller constant and the speed of the front propeller was varied. In each case the differencies from measured values were less than 2% for K_T and K_Q and less than 4% for efficiency at advance coefficients of $J = 0.9$. At larger advance coefficients ($J > 1$) the errors were of the order of 10% where, however, the accuracy of the experimental values is not clear. At small values of advance coefficient the time-accurate method performed better than the single-blade steady-state computation. However, all computations reproduce the trends correctly[10].

Opposed to the study of Wang et al. [9] who used time steps down to $0.5°/\Delta t$ the study of Feng et al. shows that even a time step of $2°/\Delta t$ can be an acceptable choise at least at moderate values of advance coefficient J. It should also be noted that the differences between results acquired with different time steps in Ref. [9] were not very large, either. According to these two CRP studies, a propeller force prediction in open water conditions within approximately 5 percent could be expected, in particular at moderate values of advance coefficient, J.

1.3.3 Propeller Acting in a Wake Field

Dhinesh [11] computed a single propeller first in open water conditions and then in a ship wake with the STAR CCM+ software. The open water computation was

conducted using the MRF approach which gave K_T and K_Q values within the order of 1% from measured values. No numerical data is given in the report so the error is evaluated visually from graphs. The case with the ship hull included was a time-accurate analysis and was computed using the sliding plane approach. In that case, the force coefficients were predicted within 5% of measured values [11].

Even if the two cases were computed using different methods the difference in errors also implies that a loss in reliability can be expected when moving from open water computations to cases where the propeller is located in a wake. This is quite expected since the wake is, firstly, not well-defined and secondly, more complicated than a uniform flow. In the current study this implies an increased uncertainty in the force prediction in the case of a pushing CRP compared to a pulling propeller set-up.

1.3.4 Conclusions

The computational tool under development is expected to be able to analyze both pushing and pulling propeller concepts. The results of Guo [7] indicate that the mixing plane approach is debatable due to the lack of physicality of the flow leaving the aft propeller and forces on structures. Also Ref. [5] states that using the mixing plane or the MRF methods with structures downstream of solid structures is not recommended. Furthermore, it is shown in [8] that the mixing plane performs quite poorly in predicting propeller loads while, however, Feng [10] states that trends are modelled correctly by all methods which is the first priority in an R&D tool.

The sliding plane is shown to perform well in several studies which was expected, since it contains the least simplifications. Earlier experience both at the Aalto University and at Rolls-Royce imply that at some point the MRF will no more be able to give reasonable results. One of such cases is for example an inclined inflow to the propeller. Thus, both literature and experience encourage to study such a method.

According to results by Wang [9], a complete convergence from the time step point of view is not reached even with a time step corresponding to a propeller rotation of a quarter of a degree. However, Feng [10] used a time step corresponding to two degrees of rotation and still achieved reasonable results. Thus, in this study two degrees of rotation per time step will be used as a starting point for the time step size in time-accurate computations and a time step corresponding to 0.5° propeller rotation will also be used as in order to provide good comparison.

Wang [9] reported the SST k-ω turbulence model to work properly in propeller computations so it will be used as a basis also in this work. According to the literature review it is clear that the sliding plane method must be a part of this study. Both steady methods have their advantages and disadvantages but at least one of them should be included due to their much lower computational costs compared to the time-accurate method. The standard OpenFOAM distribution does not have an implementation of the mixing plane method at present but includes both the MRF and sliding plane approaches so those two methods will be studied in more detail. Also the findings of Refs. [5] and [7] show the MRF as a slightly better option than

the mixing-plane. For future studies, the mixing plane should, however, also be kept in mind.

1.4 Experimental Methods

The traditional method for studying propeller performance is to conduct experiments. They are quite expencive but irreplacable due to the lack of reliable enough computational tools to predict complicated flow phenomena.

Measurements can be made in so called *towing tanks* where the propeller is transported in a pool. Usually the propeller, hub and possible surrounding structures are attached to a frame that is moved relative to water. Propeller forces and torques are measure with a dynamometer.

Another way to measure propeller performance is the *cavitation tunnel*. Contrary to towing tank, in a cavitation tunnel the propeller is fixed in space and the water flows inside the tunnel like air flows in a wind tunnel. The pressure can be adjusted in cavitation tunnels. It can be either increased to prevent cavitation or decreased to accentuate cavitation effects.

A third way to conduct measurements is to perform full-scale tests. Due to obvious reasons, they are expensive and no full-scale data is available for this study. In this work, two measurements are used as a basis for validation of computational results. The first measurements have been conducted in a cavitation tunnel and the second tests in a towing tank. The measurements used are dicussed closer in the following two subsections.

1.4.1 Cavitation Tunnel

The open-water tests were conducted at the Rolls-Royce Hydrodynamic Research Center (RRHRC) in Kristinehamn, Sweden. The facility has a closed section cavitation tunnel (T-32) that can be pressurized or depressurized in order to decrease or increase cavitation effects. Propeller forces and moments are measured with an inhouse dynamometer consisting of stretch slips that are coated with resin to protect them against water.

The tunnel itself is driven by a 250 kW motor. The propeller is driven by another smaller motor located outside of the tunnel in the upstream direction from the propeller. The driving moment is passed to the propeller through a shaft on the upstream side of the propeller. Between the shaft and water there is a non-rotating cover and at the end of the shaft there is the dynamometer for force and moment measurement. Since the motor is in the atmospheric pressure and the propeller in the pressurized tunnel, the forces induced by the pressure difference are taken out computationally by measuring the tunnel pressure at the propeller level and pressure outside the tunnel.

The test section of the tunnel $800\times\ 800\,\mathrm{mm}^2$ and the water velocity in it is measured using a pitot tube. The velocity is kept within $\pm 0.01\,\mathrm{m/s}$ of the nominal value which usually ranges from 3 to 5 m/s. The pitot tube is located upstream of

the propeller outside the tunnel wall boundary layer to represent the free velocity. This velocity is used in calculating the advance coefficients J.

The measurement uncertainties depend on the operating conditions such as water velocity and propeller rotational speed and furthermore on sensors such as the pitot tube and dynamometer. At high values of advance coefficient J, relatively high water velocities can be used which makes the uncertainty related to the inflow smaller. However, at high J-values the measured forces and moments approach zero and thus their relative uncertainty increases. At small water velocities the case is the opposite. The fact that the forces increase at low J-values poses a limit for the tunnel velocity through limitations through to dynamometer. The forces are not allowed to rise above a certain level to prevent damage on the dynamometer. Reaching a given $J = \frac{U}{nD}$ with lower absolute loads on the propeller is possible by decreasing both the water velocity and the rotational speed of the propeller [12]. The propeller used in the measurements is a Rolls-Royce design so there is also reliable surface data available for modeling with a computer.

1.4.2 Towing Tank

Tests for a CRP arrangement have been conducted at SVAtech GmbH Potsdam in a towing tank. The set-up includes a thruster and two propellers designed by Rolls-Royce. Thrust and torque were measured from both propellers through a shaft arrangement located behind the propellers. The dynamometer was the R40 from Kempf & Remmers. Thruster drag was measured with the balance R35X from Kempf & Remmers attached to the frame supporting the model. A sketch of the arrangement is shown in Figure 2. For more information on the towing tank facility, see Reference [13].

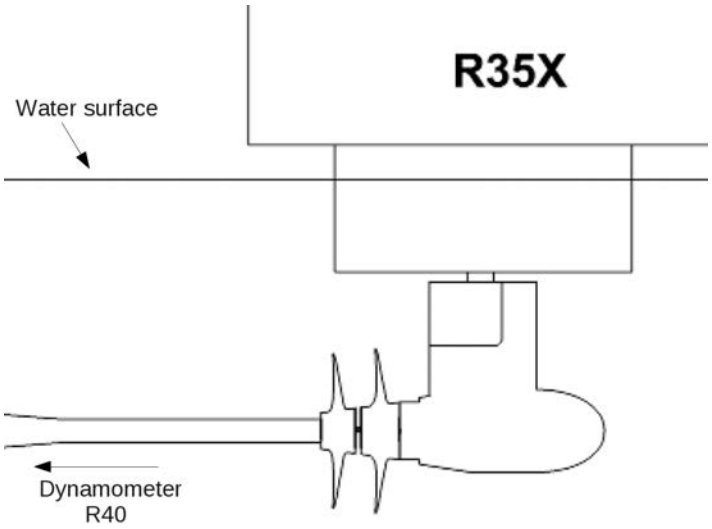


Figure 2: *Sketch of the CRP measurement set-up in the towing tank.*

1.5 Overview of the Computations

As discussed earlier, the methods used for the analysis are the sliding plane method that is time-accurate and the steady-state `MRF` method. There are two sets of measurements that are chosen to be used for validation purposes. One is an open-water case with one propeller and the other is a CRP case consisting of two propellers and a thruster unit.

In order to establish a computational method for the analysis of multi-propeller marine applications, the performance of each of the methods chosen is tested. The computations consist of six steps that are presented in Figure 3. From one case to another, new methods are introduced and the simulation becomes more complicated. At the end, all methods will have been tested and knowledge on difficulties and errors related to them will be gained, which will allow the evaluation of the feasibility of the established computational method. In the following, each step is discussed in more detail.

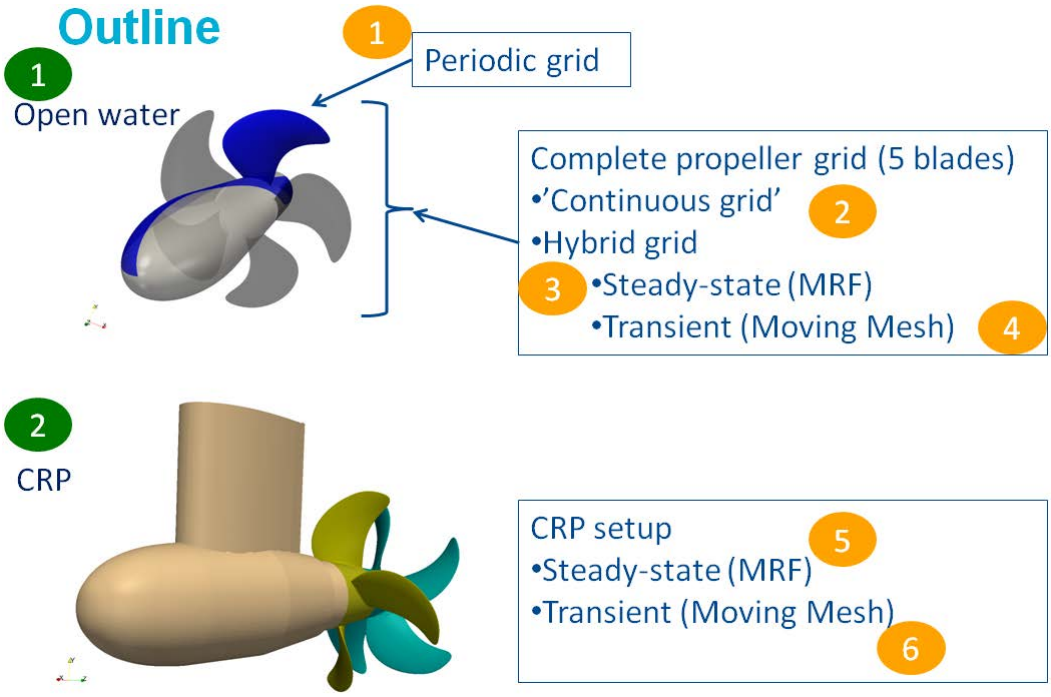


Figure 3: *Outline of the computations.*

The first step is the simplest possible one with open water conditions and only one blade. Thus, the grid has periodic boundaries and, therefore, also the least number of cells of all cases. Yet another simplification is the steady-state approximation with the `MRF` -method. The least number of cells and the lack of time accuracy makes this the computationally most efficient case and it is used as much as possible for different kinds of tests and comparisons. It is used for

- confirming the correctness of the case set-up
- testing different discretizations and other solver settings,

- grid convergence study and
- computation of an open-water curve and error analysis.

The open-water curve and the comparison to the experimental data will give the first idea of errors present in the computations. The grid convergence study ensures that the computational grid is sufficiently fine before proceeding to more complicated simulations and gives additional information on the magnitude of grid based errors.

The second case is identical with the first one with the exception that the complete propeller is included. The mesh is created from the periodic grid by rotating it five times and merging the resulting parts together. Consequently, there is no need for periodic boundaries. No significant differences were observed compared to the first case so no further results will be given conserning step 2.

In the third step, the discontinuity is introduced into the mesh. Otherwise the case will be identical to the previous ones. The new type of mesh will influence the applicability of certain solver settings and, in particular, give new information about errors and robustness of the computations. The main objective is to study the performance of the non-conformal interface between the two mesh regions. In `OpenFOAM`, the interface is called the *Arbitraty Mesh Interface* (AMI) [14].

The fourth step will give the first experiences with the time-accurate solver and moving meshes. It is the same case as the one computed in the third step but a different solver, `pimpleDyMFoam`, is used. The main focus will be on the effects due to temporal discretizations. Different parameters affecting the time accurate method are discussed and their effect on the accuracy of the computations will be studied. By the end of the fourth step all methods will have been tested in the open water case and knowledge gained on how they perform and how they are used.

In the fifth step the CRP-case will be built up and computed as a steady-state problem. The settings used in the open water case are applied on the more complicated case. The main focus is on the performance of the steady-state approximation applied on a clearly unsteady case.

The final step is a time-accurate analysis of the CRP case. It will give knowledge of computation times and possible new errors. The open water time-accurate computation has theoretically no time-dependent nature so also the performance of the time discretizations is evaluated again in a truly unsteady case.

After completing the six steps, there will be a computational set-up that is capable of analysing multi-propeller applications which was the original goal. Furthermore, there is knowledge about the errors, robustness and required computational times of the method used.

2 Governing Equations

Fluid flow must fulfill the same conservation laws as any mechanical system. Usually three laws are applied and they are the conservation of mass, momentum and energy. The difference to, for example, rigid body dynamics is that it is difficult or sometimes even impossible to track a specific set of particles that the conservation laws apply to

(*Lagrangian systems*). A common way to come around this problem is to write the equations for a fixed volume in space that does not move (*Eulerian system*). The conservation law then connects the time rate of change of the conserved variable within the volume and the *fluxes* through the volume boundaries. The total time derivative ($\frac{\mathrm{D}}{\mathrm{D}t}$) is expressed as

$$\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + \mathbf{U} \cdot \nabla = \frac{\partial}{\partial t} + U_i \frac{\partial}{\partial x_i} \tag{7}$$

where ($\frac{\partial}{\partial t}$) denotes the partial derivative with respect to time, $\mathbf{U}$ is the fluid velocity vector and ∇ is the `nabla`-operator including spacial derivatives.

Water is to a very good accuracy an incompressible fluid. It means that the effect of pressure on water density is very weak. There can still be a connection between for example temperature and density. In this work also temperature is, however, in all cases constant in space and time, so consequently the water density will be constant in all cases. This simplifies the solution routines as will be seen later. In the following sections the equations that incorporate the conservation laws of mass and momentum are first presented in their general forms and then simplifications are made based on the incompressibility assumption.

2.1 Conservation of Mass

The conservation of mass in a general case can be written as [15, 2.3]

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = \frac{\partial \rho}{\partial t} + \frac{\partial \rho U_i}{\partial x_i} = 0 \tag{8}$$

where ρ is density and $\mathbf{U}$ is the fluid velocity vector. Setting $\rho = constant$ simplifies the continuity equation to the form in which it is used throughout this work:

$$\nabla \cdot \mathbf{U} = \frac{\partial U_j}{\partial x_j} = 0 \qquad . \tag{9}$$

2.2 Conservation of Momentum

The conservation of momentum is fundamentally a statement of Newtons second law

$$\mathbf{F} = m \frac{\mathrm{D}\mathbf{U}}{\mathrm{D}t} \tag{10}$$

where $\mathbf{F}$ is a force vector and m is the mass of some set of particles. The forces include *surface forces* such as pressure- and viscous forces and *body forces* such as gravity. Being a vector equation in a three dimensional space the conservation of momentum introduces three new scalar equations.

The total derivative of velocity in Eq. (10) includes the partial time derivative and the momentum flux according to Eq. (7). Thus the conservation of momentum

per unit volume can be written

$$\rho\frac{\partial(\mathbf{U})}{\partial t} + \rho\nabla\cdot(\mathbf{U}\mathbf{U}) = -\nabla p + \nabla\cdot\tau + \rho\,\mathbf{q} \tag{11}$$

$$\rho\frac{\partial(U_i)}{\partial t} + \rho\frac{\partial(U_iU_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho\, q_i$$

The convection term, $\rho\nabla\cdot(\mathbf{U}\mathbf{U})$, has a total of nine terms corresponding to three convective terms in each of the three equations. Even if the term with the divergence of velocity is zero in an incompressible case it must be included in the equations when an iterative procedure is used for solving a steady-state case. The reason is that mass conservation is not guaranteed during the solution process.

For a *Newtonian fluid* such as water the shear stress tensor can be written in terms of velocity gradients and viscosity [15, 2.4.3]

$$\tau = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \delta_{ij}\lambda \overbrace{(\nabla\cdot\mathbf{U})}^{=0,\text{ by continuity}} \tag{12}$$

where μ is the dynamic viscosity and λ is the so called λ-viscosity that will vanish since the divergence of velocity vanishes for incompressible flows. Dividing further by density gives

$$\tau/\rho = \nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{13}$$

By defining the *strain rate tensor*

$$S = \frac{1}{2}\left(\nabla\mathbf{U} + (\nabla\mathbf{U})^T\right)$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{14}$$

the shear stress tensor can be expressed in a shorter form

$$\tau/\rho = 2\,\nu S \tag{15}$$

If the density is constant the equation can be divided by ρ. By further substituting the shear stress in Eq. (15) into the momentum equation, Eq. (11), we get the momentum equation for an incompressible newtonian fluid.

$$\frac{\partial\mathbf{U}}{\partial t} + \nabla\cdot(\mathbf{U}\mathbf{U}) = -\nabla(p/\rho) + \nabla\cdot(2\,\nu S) + \mathbf{q} \tag{16}$$

$$\frac{\partial U_i}{\partial t} + \frac{\partial}{\partial x_j}(U_iU_j) = \frac{-\partial(p/\rho)}{\partial x_j} + \frac{\partial(2\,\nu S_{ij})}{\partial x_j} + q_i$$

The equation is in the differential form and incorporates the conservation of momentum for every point in space.

2.3 Moving Reference Frame

A time-dependent case with rotating geometries can be reduced into a steady problem by changing the reference frame to follow the rotational movement. The method has been presented already in 1985 by Holmes and Tong [16] and applied for example by Siikonen [17]. In `OpenFOAM` the `MRF` method is implemented so that absolute velocities are solved with equations expressed in the relative coordinate system. To derive the equations, first some helpful definitions are introduced and then Eq. (16) is expressed in the relative coordinate system.

The rotational movement can be expressed by a rotation vector ω. The tangential velocity caused by the rotation at any point given by the displacement from origin $\mathbf{r}$ can be expressed as the cross product

$$\mathbf{U}_t = \omega \times \mathbf{r} \tag{17}$$

The time derivative of any vector $\mathbf{a}$ is experienced in a different way by an observer in the accelerating (rotating) coordinate system than by an observer in the inertial frame of reference. The relation between the time derivatives in the case of rotational movement is given as [18]

$$\left.\frac{\mathrm{D}\mathbf{a}}{\mathrm{Dt}}\right|_I = \left.\frac{\mathrm{D}\mathbf{a}}{\mathrm{Dt}}\right|_R + \omega \times \mathbf{a} \quad . \tag{18}$$

Setting $\mathbf{a} = \mathbf{r}$ gives the relationship between the absolute and relative velocities.

$$\mathbf{U}_I = \left.\frac{\mathrm{D}\mathbf{r}}{\mathrm{Dt}}\right|_I = \left.\frac{\mathrm{D}\mathbf{r}}{\mathrm{Dt}}\right|_R + \omega \times \mathbf{r} = \mathbf{U}_R + \omega \times \mathbf{r} \tag{19}$$

Furthermore, setting $\mathbf{a} = \mathbf{U}_I$, using Eq. (19) to express velocity in the relative frame of reference and noting that $\left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{Dt}}\right|_R = \mathbf{U}_R$ gives an expression for the total derivative of velocity [19]

$$\begin{aligned} \left.\frac{\mathrm{D}\mathbf{U}_I}{\mathrm{Dt}}\right|_I &= \left.\frac{\mathrm{D}\mathbf{U}_I}{\mathrm{Dt}}\right|_R + \omega \times \mathbf{U}_I = \\ \left.\frac{\mathrm{D}(\mathbf{U}_R + \omega \times \mathbf{r})}{\mathrm{Dt}}\right|_R &+ \omega \times (\mathbf{U}_R + \omega \times \mathbf{r}) = \\ \frac{\mathrm{D}\mathbf{U}_R}{\mathrm{Dt}} + \frac{\mathrm{D}\omega}{\mathrm{Dt}} \times \mathbf{r} &+ 2\,\omega \times \mathbf{U}_R + \omega \times \omega \times \mathbf{r} \quad . \end{aligned} \tag{20}$$

The momentum equation, Eq. (16), is rewritten

$$\frac{\mathrm{D}\mathbf{U}_I}{\mathrm{D}t} = -\nabla(p/\rho) + \nabla \cdot (2\,\nu S) + \mathbf{q}$$

Now it is emphasized that so far the equation has been expressed in the *inertial* coordinate system in terms of the *absolute* velocity. In the following, the terms in the momentum equation will be expressed in the *relative* coordinate system. The

total derivative is already given in Eq. (19). Pressure is a scalar quantity and thus independent of coordinate system. The diffusion term needs to be treated separately. Before starting manipulating it a helpful relation is derived. Let us define a cylindrical coordinate system (r,θ,z) whose z-axis is aligned with the ω vector and $\hat{e}$ refers to a unit vector in that system. Direction $\hat{e}_r$ points radially out from the z-axis and $\hat{e}_\theta$ is perpendicular to both $\hat{e}_r$ and $\hat{e}_z$. In such a system the rotational movement can be simply expressed as

$$\omega \times \mathbf{r} = \omega\, r\, \hat{e}_\theta \tag{21}$$

and the ∇-operator is defined as

$$\nabla = \hat{e}_r \frac{\partial}{\partial r} + \hat{e}_\theta \frac{1}{r} \frac{\partial}{\partial \theta} + \hat{e}_z \frac{\partial}{\partial z} \qquad . \tag{22}$$

A term that will arise in the derivation of the diffusion term is given as

$$\nabla(\omega \times \mathbf{r}) = \nabla(\omega\, \mathbf{r}\, \hat{e}_\theta) =$$

$$\hat{e}_r \left[\omega \hat{e}_\theta + \omega r \underbrace{\frac{\partial \hat{e}_\theta}{\partial r}}_{=0}\right] + \hat{e}_\theta\, \omega \underbrace{\frac{\partial \hat{e}_\theta}{\partial \theta}}_{=-\hat{e}_r} = \omega \hat{e}_r \hat{e}_\theta - \omega \hat{e}_\theta \hat{e}_r \qquad = \tag{23}$$

$$\begin{pmatrix} 0 & \omega \hat{e}_r \hat{e}_\theta & 0 \\ -\omega \hat{e}_\theta \hat{e}_r & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad .$$

As can be seen, the gradient of the rotational velocity is an antisymmetric matrix.

Expressing the diffusion term in the relative coordinate system yields

$$\begin{aligned}
\nabla \cdot (2\,\nu S) = \nabla \cdot \left[\nu \left(\nabla \mathbf{U}_I + (\nabla \mathbf{U}_I)^T\right)\right] = \\
\nabla \cdot \left[\nu \left(\nabla(\mathbf{U}_R + \omega \times \mathbf{r}) + (\nabla(\mathbf{U}_R + \omega \times \mathbf{r}))^T\right)\right] = \\
\nabla \cdot \left[\nu \left(\nabla \mathbf{U}_R + \nabla(\omega \times \mathbf{r}) + (\nabla \mathbf{U}_R + \nabla(\omega \times \mathbf{r}))^T\right)\right] = \\
\nabla \cdot \left[\nu \left(\nabla \mathbf{U}_R + \nabla(\omega \times \mathbf{r}) + (\nabla \mathbf{U}_R)^T + (\nabla(\omega \times \mathbf{r}))^T\right)\right] = \\
\nabla \cdot \left[\nu \left(\nabla \mathbf{U}_R + (\nabla \mathbf{U}_R)^T\right) + \nu \left(\underbrace{\nabla(\omega \times \mathbf{r}) + (\nabla(\omega \times \mathbf{r}))^T}_{=0\,,\,\text{since}\,\nabla(\omega \times \mathbf{r})\,\text{is antisymmetric}}\right)\right] = \\
\nabla \cdot \left[\nu \left(\nabla \mathbf{U}_R + (\nabla \mathbf{U}_R)^T\right)\right])
\end{aligned} \tag{24}$$

The manipulation above shows that the diffusion term can be expressed by either of the velocities, the *absolute* or the *relative* velocity. Considering the source term $\mathbf{q}$

independent of the frame of reference the momentum equation can now be expressed in the relative coordinate system

$$\frac{\mathrm{D}\mathbf{U}_R}{\mathrm{D}t} = -\nabla(p/\rho) + \nabla\cdot\left[\nu\left(\nabla\mathbf{U}_R + (\nabla\mathbf{U}_R)^T\right)\right] + \mathbf{q} \tag{25}$$

$$\underbrace{-\frac{\mathrm{D}\omega}{\mathrm{Dt}}\times\mathbf{r} - 2\,\omega\times\mathbf{U}_R - \omega\times\omega\times\mathbf{r}}_{\text{Additional terms}}$$

The total derivative of $\mathbf{U}_R$ can still be further expanded and partially expressed with absolute velocities. The reason for doing so becomes apparent later.

$$\frac{\mathrm{D}\mathbf{U}_R}{\mathrm{D}t} = \frac{\partial\mathbf{U}_R}{\partial t} + \nabla\cdot(\mathbf{U}_R\mathbf{U}_R) = \frac{\partial\mathbf{U}_R}{\partial t} + \nabla\cdot(\mathbf{U}_R\mathbf{U}_I - \mathbf{U}_R\,\omega\times\mathbf{r}) = \tag{26}$$

$$\frac{\partial\mathbf{U}_R}{\partial t} + \nabla\cdot(\mathbf{U}_R\mathbf{U}_I) - \underbrace{\nabla\cdot\mathbf{U}_R}_{=0}(\omega\times\mathbf{r}) - \underbrace{\mathbf{U}_R\cdot\nabla(\omega\times\mathbf{r})}_{=U_{R,r}\,\omega\,\hat{e}_\theta - U_{R,\theta}\,\omega\,\hat{e}_r = \omega\times\mathbf{r}} =$$

$$\frac{\partial\mathbf{U}_I}{\partial t} + \frac{\partial(\omega\times\mathbf{r})}{\partial t} + \nabla\cdot(\mathbf{U}_R\mathbf{U}_I) - \omega\times\mathbf{U}_\mathbf{R})$$

Substituting Eq. (26) to Eq. (25) and assuming that ω is constant in time we get

$$\frac{\partial\mathbf{U}_I}{\partial t} + \nabla\cdot(\mathbf{U}_R\mathbf{U}_I) = -\nabla(p/\rho) + \nabla\cdot\left[\nu\left(\nabla\mathbf{U}_R + (\nabla\mathbf{U}_R)^T\right)\right] + \mathbf{q} +$$

$$\underbrace{\omega\times\mathbf{U}_R - \omega\times\omega\times\mathbf{r}}_{=\omega\times\mathbf{U}_I} \tag{27}$$

$$\frac{\partial\mathbf{U}_I}{\partial t} + \nabla\cdot(\mathbf{U}_R\mathbf{U}_I) = -\nabla(p/\rho) + \nabla\cdot\left[\nu\left(\nabla\mathbf{U}_I + (\nabla\mathbf{U}_I)^T\right)\right] + \mathbf{q} - \omega\times\mathbf{U}_I$$

Comparison with the original momentum equation, Eq. (16), shows that there are only two changes in the momentum equation when it is expressed in the moving reference frame. The first difference is in the convection term. The change can be interpreted so that the convective velocity is the relative velocity but the velocity being solved for is the original absolute velocity. The second change is on the right-hand-side of the equation. There is an additional source term $\omega\times\mathbf{U}_I$ that comes from the coordinate transformation.

If a method for solving the Navier-Stokes equations in the inertial coordinate system exists it can be used for `MRF` computations by accounting for these two changes in the momentum equation. In practice, the new acceleration term can be included as a part of the original source $\mathbf{q}$ and is thus very easy to implement. The convective flux, which will be introduced in Chapter 3.2, must be computed in a different way and the continuity equation needs no changes whatsoever.

2.4 Turbulence

The governing equations given earlier in Eqs. (9) and (16) describe both laminar and turbulent flows. Solving turbulent flows with the basic equations is called `Direct Numerical Simulation (DNS)` and is a valid and accurate method for turbulence prediction. It is, however, too expensive for engineering purposes. The main reason for that is the ratio of largest to the smallest turbulent eddies being too large in practical flows. The smallest eddies require a very fine mesh and the largest eddies require the computational domain to be large enough so that the overall number of cells is very large. To overcome the costs different ways to model, not fully compute, turbulence have been developed [20].

In this work the Reynolds Averaged Navier-Stokes (*RANS*)-equations are used. It means that the governing equations are averaged in time and the effect of turbulence on the average flow field is modelled. For the temporal averaging velocity and pressure are divided into a temporally constant part and a fluctuating part.

$$\mathbf{U} = \overline{\mathbf{U}} + u' \tag{28}$$

$$p = \overline{p} + p' \tag{29}$$

The time average of velocity $\mathbf{U}$ is defined as

$$\overline{\mathbf{U}(\mathbf{x})} = \lim_{T\to\infty} \frac{1}{T} \int_0^T \mathbf{U}(\mathbf{x}, t)\, dt \tag{30}$$

where the averaging interval T must be larger than the temporal fluctuations but smaller than temporal variations of interest. Another way to interpret the average flow field called *ensemble averaging* is to take an average of solutions obtained by solving the same flow N times, which is described as [20, Ch. 9.4.1]

$$\overline{\mathbf{U}(\mathbf{x}, t)} = \lim_{N\to\infty} \frac{1}{N} \sum_1^N \mathbf{U}(\mathbf{x}, t) \tag{31}$$

The continuity equation being linear in velocity $\mathbf{U}$ remains unchanged in the averaging process. The case is the same for all other terms in the momentum equation except for the unlinear convection term. When averaged in time new terms containing turbulent velocity fluctuations arise. The time averaged momentum equation is written as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) = -\nabla(p/\rho) + \nabla \cdot (2\,\nu S) + \mathbf{q} + \nabla \cdot R \tag{32}$$

where the only difference to the original momentum equation is the term R, which is a second-order tensor including nine terms. Written in index notation the tensor and its divergence read

$$R_{ij} = -\overline{u_i' u_j'} \tag{33}$$

$$(\nabla \cdot R)_i = \frac{-\partial \overline{u_i' u_j'}}{\partial x_j} \quad . \tag{34}$$

Since R is, by definition, symmetric only six new terms arise due to turbulence [20, Ch. 9.4.1].

The new term is problematic, since it contains velocity fluctuations which are not known if only time averaged equations are used. A fundamental assumption behind the RANS-models is that the new term R is modelled as a diffusive term and thus it is often called the *Reynolds stress tensor*. This also explaines the notation R. According to the Boussinesq-hypothesis, Reynolds stresses are given as

$$R_{ij} = -\overline{u_i' u_j'} = 2\,\nu_T S_{ij} - \frac{2}{3} k \delta_{ij} \tag{35}$$

which implies that turbulence can be modelled through two terms: The turbulent viscosity, ν_T, and the turbulence kinetic energy, k. Comparing to the molecular diffusion term it is seen that the turbulent viscosity works exactly as the molecular viscosity. The turbulence kinetic energy is defined as

$$k = \frac{1}{2}\overline{u_k' u_k'} \tag{36}$$

and is, as implied by its name, the kinetic energy of the turbulent velocity fluctuations per unit mass. The last term in Eq. (35) ensures that k is always correctly reproduced from the definition of R.

2.5 Turbulence Modelling: SST $k-\omega$ RANS-Model

The SST k-ω-model is a two-equation RANS model. It includes two new equations from which the effect of turbulence can be modelled. One equation is written for the turbulence kinetic energy, k. The second equation is written for the specific dissipation rate, ω, of the turbulence kinetic energy k and the turbulent viscosity is defined as

$$\nu_T = \frac{k}{\omega} \tag{37}$$

but is computed according to the Bradshaw assumption presented later. The turbulence equations are presented in more detail in the following.

2.5.1 k- and ω - Equations

The model equation for the turbulent kinetic energy k is given as

$$\begin{aligned} \frac{\partial k}{\partial t} + \nabla\cdot(Uk) - (\nabla\cdot U)k \;&-\; \nabla\cdot(D_{k,eff}\nabla k) \\ &=\; \min\{G\,,\, c_1\beta^* k\omega\} - \beta^*\omega k \end{aligned} \tag{38}$$

and the model equation for the specific dissipation, ω, reads

$$\begin{aligned} \frac{\partial \omega}{\partial t} + \nabla\cdot(U\omega) - (\nabla\cdot U)\omega \;&-\; \nabla\cdot(D_{\omega,eff}\nabla\omega) \\ &=\; \gamma(F_1)G/\nu_T - \beta(F_1)\omega^2 - (F_1-1)CD_{k,\omega} \quad . \end{aligned} \tag{39}$$

The expressions for the parameters and filtering functions in Eqs. (38) and (39) will be given later.The third terms on the left-hand-sides of Eqs. (38) and (39) should vanish according to the conservation of mass but are still required in iterative steady-state computations. The term may influence the iteration process but should not have any effect on the final result. Turbulent kinetic energy production G in Eq. (38) is given as

$$G = 2\nu_T |S|^2 \tag{40}$$

2.5.2 Model Parameters

The diffusion coefficients $D_{k,eff}$ and $D_{\omega,eff}$ are defined in Eq. (41).

$$\begin{aligned} D_{k,eff} &= \alpha_k(F_1)\nu_T + \nu \\ D_{\omega,eff} &= \alpha_\omega(F_1)\nu_T + \nu \end{aligned} \tag{41}$$

The value of the coefficients $\gamma(F_1)$, $\alpha_k(F_1)$, $\alpha_\omega(F_1)$ and $\beta(F_1)$ are filtered between the model coefficients by function F_1 according to Eq. (43).

$$(\gamma\ \alpha_k\ \alpha_\omega\ \beta)^T = F_1\,(\gamma\ \alpha_k\ \alpha_\omega\ \beta)_1^T + (1 - F_1)\,(\gamma\ \alpha_k\ \alpha_\omega\ \beta)_2^T \tag{42}$$

with the following values

$\gamma_1 = 0.5532$	$\alpha_{k1} = 0.85034$	$\alpha_{\omega 1} = 0.5$	$\beta_1 = 0.075$
$\gamma_2 = 0.4403$	$\alpha_{k2} = 1.0$	$\alpha_{\omega 2} = 0.85616$	$\beta_2 = 0.0828$

Coefficient β^* has the constant value of $\beta^* = 0.09$ and $c_1 = 10$. The switching function which governs the choice between the ω- and the ε-equations is

$$F_1 = \tanh\left(\Gamma^4\right) \tag{43}$$

where

$$\Gamma = \min\left\{\min\left[\max\left(\frac{\sqrt{k}}{\beta^*\omega y}; \frac{500\nu}{\omega y^2}\right); \frac{4\alpha_{\omega 2}k}{CD_{k\omega}y^2}\right]; 10.0\right\} \tag{44}$$

Term $CD_{k\omega}$ in Eq. (44) is defined as

$$CD_{k\omega} = \frac{2\alpha_{\omega 2}}{\omega}\left(\nabla k \cdot \nabla\omega\right) \tag{45}$$

and is limited to the lower limit of $CD_{k\omega} \geq 1 \cdot 10^{-10}$. The eddy viscosity ν_T is calculated according to the Bradshaw modification as

$$\nu_T = \frac{a_1 k}{\max\left(a_1\omega; \beta_1 F_2 \frac{1}{\sqrt{2}}\ |S|\right)} \tag{46}$$

where $a_1 = 0.31$. The term F_2 is a switching function dependent on wall distanse. It is defined as

$$F_2 = \tanh\left(\Gamma_2^2\right) \tag{47}$$

where

$$\Gamma_2 = \min\left[\max\left(\frac{2\sqrt{k}}{\beta^*\omega y}; \frac{500\nu}{\omega y^2}\right); 100\right] \tag{48}$$

2.5.3 Boundary Treatment

The behaviour of turbulence near solid walls differs from the free stream conditions. This is easy to understand if one thinks of turbulence as eddies. The viscinity of a wall must affect the behaviour of eddies and, on the other hand, eddies change flow behaviour compared to laminar flow.

Turbulent boundary layers have been researched over the years and it is possible (and mandatory) to introduce some of the results into a CFD-computation. The domain where the viscous effects in the wall region affect the flow is called a *boundary layer* and can be divided into three parts. The division is made according to the following non-dimensional quantities: a dimensionless velocity (u^+) and a dimensionless distance from the wall (y^+). The non-dimensionalisation is made as follows

$$u^+ = \frac{\mathbf{U}}{u_\tau} \tag{49}$$

$$y^+ = \frac{y u_\tau}{\nu} \tag{50}$$

where the *friction velocity*, u_τ, is defined as

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad . \tag{51}$$

Friction velocity is a velocity scale based on the friction force on the wall. It has originally been derived by conducting a dimensional analysis of the velocity profile in the lower parts of a viscous boundary layer [15].

A turbulent boundary layer is shown in Figure 4. The *overlap layer* or *log-layer* in the middle of the boundary layer obeys a logarithmic relationship to the distance from wall, which has been theoretically derived and confirmed experimentally. It extends approximately from $y^+ = 35$ to $y^+ = 350$. The velocity profile in the log-layer is given as

$$u^+ = \frac{1}{\kappa} \ln(E y^+) \tag{52}$$

where $\kappa = 0.41$, called von Kàrmàn constant, and $E = 9.8$ are experimental constants. The flow in the log-layer is dominated by both viscous effects and turbulent eddies.

The part closest to the surface ($y^+ \leq 5$) is dominated by viscous effects only and is called a *viscous sublayer*. Within the viscous sublayer, the dimensionless velocity, u^+, is directly proportional to the dimensionless distance from wall, y^+, and is given as

$$u^+ = y^+ \quad . \tag{53}$$

Between these two regions there is the so called *buffer layer* where velocity does not obey either of the laws. Spalding has proposed a fitting that matches data in all of the three parts well up to the outermost part of the boundary layer [15]. However, for the computational model only a distinction between the viscous sublayer and

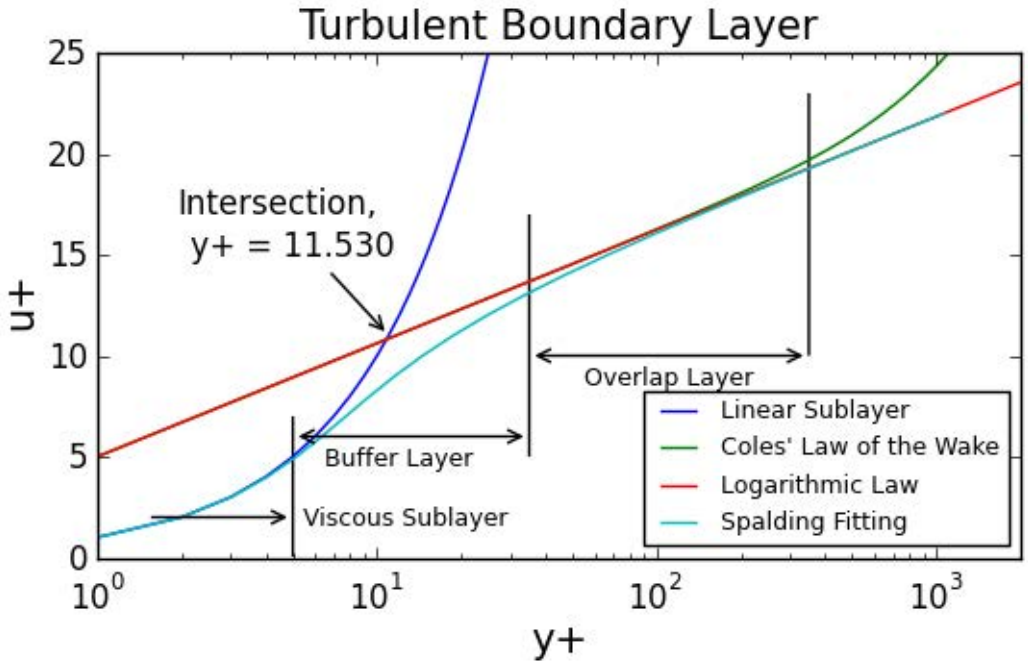


Figure 4: *Turbulent boundary layer.*

the logarithmic layer is made. By setting expressions (52) and (53) equal, the intersection of the logarithmic and linear profiles can be found. Ten Newton's iterations with the initial guess of $y^+ = 11$ gives approximately the boundary between the two layers.

$$y^+_{lam} = \frac{1}{\kappa}\ln(E\, y^+_{lam}) \implies y^+_{lam} \approx 11.530 \tag{54}$$

Under the assumptions of diffusion dominating over convective terms and pressure gradient and the turbulent diffusion dominating over molecular diffusion, k and ω, respectively, must fulfill Eqs. (55) and (56) in the logarithmic layer [21, Ch. 4.6].

$$k_{Log} = \frac{u_\tau^2}{\sqrt{\beta^\star}} \tag{55}$$

$$\begin{aligned}\omega_{Log} &= \frac{u_\tau}{\sqrt{\beta^\star}\kappa y} \\ &= \frac{\sqrt{k}}{C_\mu^{1/4}\kappa y}\end{aligned} \tag{56}$$

Wilcox also shows that ω is proportional to $\omega \propto \frac{1}{y^2}$ in the viscous sublayer:

$$\omega_{Sublayer} = \frac{6\nu}{\beta_1 y^2} \tag{57}$$

Such an asymptotic behaviour may lead to considerable numerical errors if ω is evaluated from the transport equations so, in order to avoid the errors, ω-value

should be explicitly set according to Eq. (57) in 7 to 10 cells closest to wall that should all lie within $y^+ < 2.5$ [21, Ch. 7.2]. Further discussion of the possibilities of modelling turbulence near the walls is given by Hellsten [22].

In `OpenFOAM`, the expressions of ω given for the viscous sublayer and logarithmic layer are averaged in the first cell

$$\omega = \sqrt{\omega_{Sublayer}^2 + \omega_{Log}^2} \tag{58}$$

If one considers how the two terms behave as a function of the distance from wall, y, the sublayer expression (57) being proportional to

$$\omega_{Sublayer} \propto \frac{1}{y^2}$$

dominates near the wall and the expression for the logarithmic layer, Eq. (56), being proportional to

$$\omega_{Log} \propto \frac{1}{y}$$

will dominate at a greater distance from wall.

In addition to specific turbulent dissipation ω, also the turbulent viscosity is treated in a special way near walls. If y^+ is above the point where the linear profile of the viscous sublayer and the logarithmic layer cross, turbulent viscosity is evaluated according to the logaritmic layer relations by introducing an *effective* kinematic viscosity defined as

$$\tau_w = \rho\,\nu_{eff}\frac{\partial \mathbf{U}}{\partial y} \approx \rho\,\nu_{eff}\,\frac{\mathbf{U}_P}{y_P}$$

$$\nu_{eff} \approx (\tau_w/\rho)\,\frac{y_P}{\mathbf{U}_P} \tag{59}$$

where the index P refers to the first cell center from wall. Substituting the logarithmic velocity profile, Eq. (52), and the definition of friction velocity, Eq. (51)

$$u^+ = \frac{\mathbf{U}}{u_\tau} = \frac{1}{\kappa}\ln(Ey^+)$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}$$

$$\implies \tau_w = \frac{\rho\,u_\tau\,\kappa\,\mathbf{U}}{\ln(Ey^+)} \tag{60}$$

into Eq. (59) yields the expression for the effective viscosity.

$$\nu_{eff} = \nu\,\overbrace{\frac{\kappa}{\ln(Ey^+)}}^{u_\tau}\overbrace{u_\tau\,\frac{1}{\nu}\,y_P}^{y^+} = \nu\,\frac{\kappa\,y^+}{\ln(Ey^+)} \tag{61}$$

Below the intersection point, turbulent viscosity is neglected and the effective viscosity is the molecular viscosity.

$$\nu_{eff} = \nu \text{ , if } y^+ < y^+_{lam} \tag{62}$$

With the treatments for ω and ν_T, no direct treatment is applied for the turbulent kinetic energy k. The transport equation for k, Eq. (38), is modified near walls. The production term G is evaluated from

$$G_{\text{near wall}} = \nu_{eff} \frac{|\nabla \mathbf{U} \cdot \mathbf{n}| C_\mu^{1/4} \sqrt{k}}{\kappa y} \tag{63}$$

which affects the value of k near boundaries.

3 Numerical Methods

The governing equations described in Sect. 2 are partial differential equations. They are also non-linear and have continuous solutions. Analytical solutions to the equation are nowadays only known for rather simple cases and usually numerical methods are used to solve cases of engineering interest. For computers the equations must be brought into an algebraic form. Furthermore, the equations are linearized. The outcome is a set of algebraic linear equations that is solved iteratively.

In order to turn the continuous equations into several discrete algebraic equations, the governing equations are formulated for discrete points in space and time. There are several ways to achieve this. A traditional way is the *finite difference* method that is used for solving variables at some discrete points and approximating derivatives with the Taylor series expansion. Another way is to divide the physical domain into finite blocks inside which a shape function is used to approximate the variables. Variables are solved for in nodal points on the finite block. The most common way in computational fluid dynamics (CFD) is the finite-volume approach. It means that the governing equations are integrated over a control volume that is also called a *cell*. Nowadays the so called co-located method is used. It means that variables are solved in the center of each cell. In this chapter some basic principles used in the formulation of the algebraic equations are shown with the help of a general transport equation, Eq. (65), for a general variable ϕ. The discretization of the computational domain will be explained in Chapter 3.1 and the discretization of the equations in Chapter 3.2.

3.1 Discretization in Space

The physical domain is divided into control volumes that must fulfill some requirements such as they are never negative and they must completely fill the solution domain. An example of two cells is shown in Figure 5.

All values are solved in the cell centers (points P and N). The terminology used in `OpenFOAM` is used here which will make the derivation of the numerical schemes

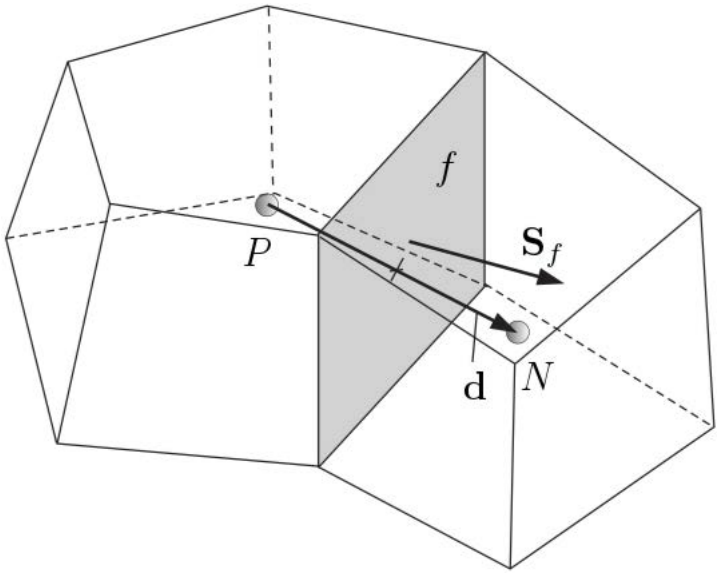


Figure 5: *Cell in* `OpenFOAM`.

more understandable. Cells are connected by faces. Each face is owned by one adjacent cell and the other cell is called the *neighboring* cell. The face f has an area $|\mathbf{S}_f|$ and a unit normal vector $\mathbf{n}$ pointing towards the neighbor. Thus a surface area vector is defined as

$$\mathbf{S}_f = |\mathbf{S}_f|\mathbf{n}$$

The vector $\mathbf{d}$ points from the center of the owner cell to the center of the neighbor cell

$$\mathbf{d} = \overline{PN} \tag{64}$$

The vector **Pf** points from the center of the owner cell to the center of face f and the vector **Nf** from the center of the neighbor cell to the center of face f. The volume of the owner cell is defined as V_P.

The cells are not ordered in any way and they are connected to each other through the faces only, not by any special indexing system. Such a mesh is called unstructured. [23]

3.2 Equation Discretization

A general form of a transport equation is used to present how different terms are transformed from the differential forms into the discrete form.

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{U}\phi) = \nabla \cdot (\alpha \nabla \phi) + \mathbf{q}_\phi \tag{65}$$

In order to discretize the equation according to the finite-volume approach, the equation is integrated over a control volume V to obtain.

$$\overbrace{\int_V \frac{\partial \phi}{\partial t}\, dV}^{\text{time derivative}} + \overbrace{\int_V \nabla \cdot (\mathbf{U}\phi)\, dV}^{\text{convection term}} = \overbrace{\int_V \nabla \cdot (\alpha \nabla \phi)\, dV}^{\text{diffusion term}} + \overbrace{\int_V \mathbf{q}_\phi\, dV}^{\text{source term}} \tag{66}$$

The integration of the time derivative and the source term is straightforward. The terms are simply multiplied by the cell volume V_P. There are several ways for approximating the time derivative itself. The simplest one is the *implicit Euler* method where the time derivative is approximated as

$$\int_V \frac{\partial \phi}{\partial t}\, dV \approx \frac{(\phi\, V)^{n+1} - (\phi\, V)^n}{\Delta t} \tag{67}$$

where n is the readily solved time level and $n+1$ the time level in which the variables are being solved. The method is first-order accurate in time but unconditionally stable. The more accurate option is the *backward* method given as

$$\int_V \frac{\partial \phi}{\partial t}\, dV \approx \frac{3(\phi\, V)^{n+1} - 4(\phi\, V)^n + (\phi\, V)^{n-1}}{2\, \Delta\, t}. \tag{68}$$

The backward method is second-order accurate in time with the cost of higher requirements for memory due to the need to store the solution from two previous time levels (n and $n-1$) [20].

The integration of convection and diffusion terms needs a special attention. The Gauss divergence theorem is applied on them which, in its general form, can be written as [18, Ch. 16.4]

$$\int_V \nabla \star \phi dV = \int_S \phi \star \mathbf{dS} \tag{69}$$

where ϕ is a (smooth) tensor field, $\nabla\star$ represents any of the derivatives $\nabla\cdot$ (divergence), $\nabla\times$ (curl) or simply ∇ (gradient). Surface S is closed and encloses the volume V_P and $\mathbf{dS}$ is the surface vector pointing out from the volume V_P. Applying the Gauss theorem on the convection term the outer divergence operation will vanish and the volume integral will be turned into a surface integral

$$\int_V \nabla \cdot (\mathbf{U}\phi)\, dV = \int_S \phi \mathbf{U} \cdot \mathbf{dS} \tag{70}$$

By further approximating the surface integral by a sum over finite discrete parts, $\mathbf{S}_f$, of surface S, the convection term becomes

$$\int_V \nabla \cdot (\mathbf{U}\phi)\, dV = \int_S \phi \mathbf{U} \cdot \mathbf{dS} \approx \sum_f \phi \mathbf{U} \cdot \mathbf{S}_f \tag{71}$$

Similar steps can be taken with the outer divergence operation in the diffusion term. The discrete form of the transport equation is then obtained as

$$V_P \frac{\partial \phi}{\partial t} + \sum_f \phi \mathbf{U} \cdot \mathbf{S}_f = \sum_f (\alpha \nabla \phi) \cdot \mathbf{S}_f + V_P \mathbf{q}_\phi \tag{72}$$

Since the volume integral is turned into a surface integral the variables $(\mathbf{U}, \alpha, \phi, \nabla\phi)$ need to be expressed on the surfaces, too. Since they are normally expressed in cell centers, a need for interpolation arises. There are several ways to interpolate variables. In the following, three schemes are introduced.

3.2.1 Linear interpolation

Linear Interpolation or *Central Differencing* means that the value ϕ is interpolated onto the face by weighting the two adjacent cell center values by their distances to the face

$$\phi_f \approx w\phi_P + (1 - w)\phi_N \tag{73}$$

so that the weight factor (w) is the ratio of the neighboring cell center distance to the face and to the owner cell center [23]

$$w = fN/PN \tag{74}$$

Central differencing is symmetrical and is used in the discretization of diffusion terms. It is, however, unbounded which means that non-physical new minimum or maximum values may arise due to interpolation. At the worst, this may lead to unstable computation which usually prevents its usage in conjunction with convective terms.

Looking at the truncation error in a one-dimensional non-equally spaced mesh (Figure 6), expressing ϕ in points P and N, and by developing it into a Taylor series

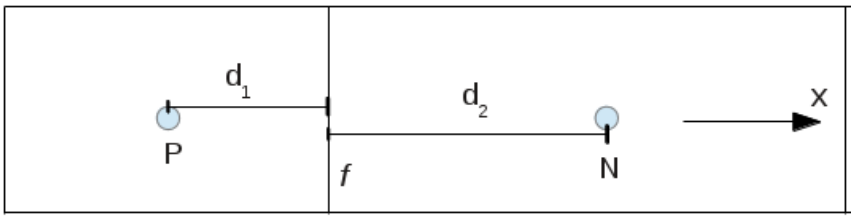


Figure 6: *Set-up for studying truncation errors.*

around the face f yields

$$\phi_P = \phi_f - d_1 \frac{\partial \phi}{\partial x} + \frac{d_1^2}{2} \frac{\partial^2 \phi}{\partial x^2} - \frac{d_1^3}{6} \frac{\partial^3 \phi}{\partial x^3} + \frac{d_1^4}{24} \frac{\partial^4 \phi}{\partial x^4} + \mathcal{O}(d_1^5) \tag{75}$$

$$\phi_N = \phi_f + d_2 \frac{\partial \phi}{\partial x} + \frac{d_2^2}{2} \frac{\partial^2 \phi}{\partial x^2} + \frac{d_2^3}{6} \frac{\partial^3 \phi}{\partial x^3} + \frac{d_2^4}{24} \frac{\partial^4 \phi}{\partial x^4} + \mathcal{O}(d_2^5) \tag{76}$$

Substituting ϕ_P and ϕ_N into Eq. (73) yields

$$\phi_f \approx w\phi_P + (1-w)\phi_N = \phi_f + \underbrace{d_1 d_2 \frac{1}{2}\frac{\partial^2 \phi}{\partial x^2} + \mathcal{O}(d_1^3) + \mathcal{O}(d_2^3)}_{\text{Truncation error}} \tag{77}$$

The leading term in the truncation error is proportional to second spatial derivative of ϕ ($\frac{\partial^2 \phi}{\partial x^2}$) and to $d_1 d_2$ which means that the method is second-order accurate in space. If the cell size is halved the leading term of truncation error reduces to one fourth of the initial magnitude.

3.2.2 Upwind Interpolation

Another simple interpolation scheme is the *Upwind interpolation*. It is used in the discretization of the convection term and can be thought to represent the *transporting* nature of the term. As the name implies it needs information on which side of the face is on the upwind or upstream side. In the convection term, there is always the *convecting* flux (usually surface normal velocity) and the *convected* variable ϕ. It is the convected variable that is interpolated with the upwind scheme. The upwind interpolation of ϕ is written as

$$\phi_f = \begin{cases} \phi_P & , \mathbf{U}_f \cdot \mathbf{S}_f \geq 0 \\ \phi_N & , \mathbf{U}_f \cdot \mathbf{S}_f < 0 \end{cases} \tag{78}$$

Eq. (78) can be written in the same form as Eq. (73). Then the the weight factor is

$$w_{\text{upwind}} = \begin{cases} 1 & , \mathbf{U}_f \cdot \mathbf{S}_f \geq 0 \\ 0 & , \mathbf{U}_f \cdot \mathbf{S}_f < 0 \end{cases} \tag{79}$$

The upwind interpolation is bounded so that it can never produce oscillations [23].

In the case of a positive convective velocity one can use Eq. (75) to study the truncation error of the upwind interpolation.

$$\phi_f \approx \phi_P = \phi_f \underbrace{-d_1 \frac{\partial \phi}{\partial x} + \frac{d_1^2}{2}\frac{\partial^2 \phi}{\partial x^2} + \mathcal{O}(d_1^3)}_{\text{Truncation error}} \tag{80}$$

The truncation error consists of a *first*-order term in addition to a similar *second*-order term that was present in the truncation error of the linear interpolation method. The first-order term is proportional to $\frac{\partial \phi}{\partial x}$ and is thus *damping* or *diffusive* in its nature so that it tends to smooth out any local peaks in the values of ϕ. Being a first-order term it is also larger than the truncation error of the linear interpolation method which makes the upwind method usually too inaccurate to be used for obtaining a reliable final result. Due to its robustness it can be used for starting computations, however.

3.2.3 Blended Interpolation

A combination of central- and upwind interpolations can be formulated as

$$\phi_f = (1-\gamma)(\phi)_{\text{upwind}} + \gamma(\phi)_{\text{CD}} \tag{81}$$

where γ is a blending coefficient [23]. This formulation offers a possibility to improve the accuracy from that of the upwind scheme but still maintain stability.

In the *blended* differencing the value of γ is a constant. There are, however, methods that locally determine γ so that new extreme values are avoided with as high an accuracy as possible. The local value for γ is calculated based on the flow field itself. There are a range of such limiters in `OpenFOAM` such as the `limitedLinear` scheme and many more in literature (see for example Hirsch [24, Ch. 21.3]) such as *minmod* and *van Leer* to mention a few.

The truncation error of the blended scheme consists of those of the upwind and linear schemes. Thus it includes a first-order term that is scaled according to the value of γ which makes the scheme more diffusive compared with the linear interpolation, but also improves stability [23].

3.3 The SIMPLE-algorithm

The Semi-Implicit Method for Pressure Linked Equations (SIMPLE) is a common method for finding a solution to the continuity- and momentum equations in particular in incompressible cases. A traditional description of the algorithm is given by Patankar [25] and the description in this chapter follows the implementation in `OpenFOAM`. The solution algorithm is very similar in both steady-state and time-accurate cases with moving meshes. In a time-dependent case, the algorithm is used inside each time step while in a steady-state solution it can be used as such. The solver used in the steady-state computations is `simpleFoam` and the time-accurate solver is `pimpleDyMFoam`. The name of the time-accurate solver stands for a combination of the `PISO`-algorithm, that is a common method for solving time-accurate problems [26], and of the SIMPLE-algorithm. The code allows the use of dynamic meshes, which is expressed by the part `DyM` in the name. The two codes have some differencies that will be pointed out in the following derivation.

First, the momentum equation is formed. Each term in Eq. (16) is discretized by integrating over a control volume (V_P) and applying the Gauss law, Eq. (69), to transform the volume integrals into surface integrals.

$$\overbrace{\frac{\partial \mathbf{U}}{\partial t}V_P}^{time\ derivative} + \overbrace{\sum_f \phi_f \mathbf{U}_f}^{conv} - \sum_f \nu_f \mathbf{S}_f \left[\overbrace{(\nabla U)_f}^{1.diff} + \overbrace{(\nabla U)_f^T - \frac{1}{3}tr(\nabla U)^T I}^{2.diff} \right] = \mathbf{Q} - \overbrace{\sum_f (\mathbf{S}_f p)}^{\int_{V_P} \nabla p\, dV} \tag{82}$$

where

$$\phi_f = \mathbf{U}_f \cdot \mathbf{S}_f \tag{83}$$

is the volume flux through face f that is obtained at the end of the iteration process or, if no previous loop has been conducted, by a linear interpolation of cell-center velocities. If the `MRF` method is applied the volume flux is expressed in the relative coordinate system as discussed in Chapter 2.3. In the case of a moving mesh, ϕ_f is relative to mesh motion

$$\omega \times \mathbf{r} \cdot \mathbf{S}_f. \tag{84}$$

The velocity gradient tensors are not yet in their discrete form nor is the time derivative for the sake of a clearer presentation. In Eq. (82), only the surface area vector $\mathbf{S}_f$ and the volume flux ϕ_f are, by definition, known on cell faces. The face values of other variables must somehow be approximated from their cell-center values. The effective viscosity ν_{eff} was chosen to be linearly interpolated which reflects the symmetric nature of diffusion.

The calculation of the viscosity term is divided into two parts since the treatment of the velocity gradient tensor differs between them in the code. In the first term, denoted by *1. diff*, a part of the gradient (the orthogonal part) is computed directly from face adjacent cell-center values

$$(\nabla \mathbf{U})_f = \frac{\mathbf{U}_n - \mathbf{U}_o}{\mathbf{d}} \tag{85}$$

This is a good approximation if the vector $\mathbf{d}$ that connects the cell centers is parallel to the surface area vector $\mathbf{S}_f$. If the vectors are not aligned, however, velocity gradients computed in the cell centers are used as a correction term. In order to construct the correction term, the cell-center velocity gradients are computed using the Gauss law

$$\nabla \mathbf{U} = \frac{1}{V_P} \sum_f \mathbf{S}_f \mathbf{U}_f \tag{86}$$

where the velocities are interpolated onto the cell faces according to the linear interpolation method. The velocity gradients in the second part of the diffusion term (*2. diff*) are first computed in cell centers according to Gauss law, Eq. (69), and then linearly interpolated onto the faces. Also the pressure gradient is computed by linearly interpolating the cell-center values onto the cell faces. The time derivative is non-zero only in time accurate computations and can be approximated by using any of the schemes presented in Chapter 3.2. The source term $\mathbf{Q} = \mathbf{q}V_P$ can contain any volume force (per unit mass) such as gravity or electromagnetic forces. If the moving reference frame (MRF) is activated, the additional source term

$$\omega \times \mathbf{U_I} V_P \tag{87}$$

is also included in $\mathbf{Q}$. Eq. (82) stands for force balance in *one* control volume V_P. Writing a similar equation for each cell leads to a set of equations with as many equations as there are cells. The set of equation can be presented as a matrix equation. Splitting the left-hand-side into diagonal and off-diagonal contributions and writing the equation for only one cell gives

$$A_P \mathbf{U}_P + \sum_n A_n \mathbf{U}_n = \mathbf{RHS} \tag{88}$$

where A_P stands for the diagonal term and A_n for the off-diagonal terms in one row of the $(n \times n)$ coefficient matrix A. The diagonal term (A_P) multiplies the velocity U_P that the equation is written for. Accordingly, the off-diagonal terms (A_n) multiply the velocities of the neighboring cells U_n which gives a more understandable meaning to the matrix coefficients: the diagonal term represents the contribution of the cell itself on the new value to be solved, and the off-diagonal terms contain the contributions of the neighboring cells. The fact that usually only up to the *second* neighbours are included in each equation (e.g. in $\nabla\mathbf{U}$-term) explains the fact that the coefficient matrix is quite *sparse* i.e. most of its terms are zero. The term **RHS** contains all explicit terms in Eq. (82) which includes among others the second part of the diffusion term and the pressure gradient as such.

Before solving the momentum equation the matrix equation is under-relaxed in order to increase the diagonal dominance of the coefficient matrix and improve computational stability. If the coefficient matrix is not diagonally dominant

$$A_P < \sum_n A_n$$

the diagonal term is first replaced by the sum of the off-diagonal terms and a corresponding term

$$\sum_n A_n - A_P$$

is added onto the right-hand-side of the equation. The actual under-relaxation means that the diagonal terms A_P are divided by a relaxation factor $0 < \alpha_U \leq 1$. A corresponding term is added to the right-hand-side to keep the equation valid, and thus the momentum equation becomes

$$\frac{A_P}{\alpha_U}\mathbf{U}_P + \sum_n A_n U_n = \mathbf{RHS} + \frac{(1-\alpha_U)}{\alpha_U} A_P \mathbf{U}_P \tag{89}$$

from which a new velocity can be solved.

In order to define the *pressure equation*, new definitions are needed. Firstly, the pressure dependent part is removed from the right-hand-side of the momentum equation (**RHS**). The pressure gradient is not written in its discretized form for the sake of clarity

$$\mathbf{RHS} = \mathbf{rhs} - \nabla p \; V_P \qquad . \tag{90}$$

Furthermore, a so called H-operator is defined as

$$\mathbf{H}(\mathbf{U}) = \frac{\mathbf{rhs} - \sum_n A_n \mathbf{U}_n}{V_P} \tag{91}$$

and the diagonal term is divided by the cell volume V_P.

$$a_P = \frac{A_P}{V_P} \tag{92}$$

Dividing Eq. (88) and substituting definitions (90), (91) and (92) gives

$$a_P \, U_P = \mathbf{H}(\mathbf{U}) - \nabla p. \tag{93}$$

Dropping out the pressure gradient, a velocity-like term and a corresponding volume flux are obtained. They are denoted by an upper index $*$

$$\mathbf{U}^* = \frac{\mathbf{H}(\mathbf{U})}{a_P} \tag{94}$$

$$\phi_f^* = \mathbf{U}_f^* \cdot \mathbf{S}_f \tag{95}$$

The linear interpolation of $\mathbf{U}^*$ is used to obtain the flux ϕ_f^* on each face. If the `MRF` is active ϕ_f^* is expressed in the relative coordinate system by subtracting the term

$$(\omega \times \mathbf{r}) \cdot \mathbf{S}_f.$$

Substituting $\mathbf{U}^*$ back into Eq. (93) and dividing by a_P gives an expression for velocity

$$\mathbf{U}_P = \mathbf{U}^* - \frac{\nabla p}{a_P} \tag{96}$$

The continuity equation, Eq. (9), states that the velocity field has no divergence. Taking the divergence of Eq. (96) leads to an expression between the velocity and pressure.

$$\nabla \cdot \mathbf{U}_P = \nabla \cdot \mathbf{U}^* - \nabla \cdot \left(\frac{\nabla p}{a_P} \right) = 0$$
$$\implies \nabla \cdot \left(\frac{\nabla p}{a_P} \right) = \nabla \cdot \mathbf{U}^*$$

Both divergence terms are discretized using the Gauss law (Eq. 69)

$$\sum_f \left((\frac{1}{a_P})_f (\nabla p \cdot \mathbf{S}_f)_f \right) = \sum_f \phi_f^* \tag{97}$$

The coefficient

$$\frac{1}{a_P}$$

that comes from the coefficient matrix of the momentum equation and is stored for each cell, is interpolated onto the faces using the linear interpolation. The surface normal pressure gradient

$$\nabla p \cdot \mathbf{S}_f$$

is evaluated partly in an explicit and partly implicit manner like the first part of the diffusion term in the momentum equation (see Eqs. (85) and (86)).

To account for the explicit non-orthogonality correction, the pressure equation (97) is repeated several times. After solving the pressure equation, a new volume flux ϕ_f is obtained by subtracting a flux term (F_p) obtained from the actual discrete pressure equation based on Eq. (97) from ϕ_f^*, so that ϕ_f has zero divergence i.e. the convective flux exactly fulfills the continuity constraint.

$$\phi_f = \phi_f^* - F_p \tag{98}$$

After obtaining flux ϕ_f, the pressure is explicitly under-relaxed as

$$p^{n+1} = \alpha_p p + (1 - \alpha_p) p^n \tag{99}$$

Finally, a new velocity is obtained by correcting $\mathbf{U}^*$ with the latest pressure gradient

$$\mathbf{U}^{n+1} = \mathbf{U}^* - \frac{1}{a_P} \nabla p^{n+1} \tag{100}$$

3.4 Linear Solvers

The matrix equations arising from the governing equations are typically sparse in CFD. In this Chapter, solution methods for a general matrix equation, written as

$$A\phi = Q \tag{101}$$

are presented mainly by following the description by Ferziger and Perić [20]. Since the matrices are sparse, they are solved iteratively and for which there is a range of different algorithms available. Two solvers, the preconditioned biconjugate gradient (PBiCG) and the generalized geometric-algebraic multi-grid (GAMG), are used in the computations presented in this study. The PBiCG method is used for solving the equations for velocity and turbulence quantities, and the GAMG for the pressure equation.

The idea of conjugate gradient methods is that solving the matrix equation is reformulated as a minimization problem of a function F.

$$F = \frac{1}{2} \phi^T A \phi - \phi^T Q \tag{102}$$

The minimum is searched for by finding minima in one direction at a time. All directions can be thought to be perpendicular to all preceeding directions and the error is reduced on every iteration. In theory there are as many linearly independent directions as there are equations (rows) in the matrix equation. The rate of convergence depends on the condition number

$$\kappa = \frac{\lambda_{max}}{\lambda_{min}} \tag{103}$$

of the coefficient matrix A where λ_{max} and λ_{min} are the largest and the smallest eigenvalues of the matrix. The condition number is often very large in CFD applications and manipulating it allows to improve convergence. The condition number can be reduced by multiplying the original equation by some matrix so that the solution of the equation is not changed but the eigenvalues of the matrix are. This step is called preconditioning.

Conjugate gradient methods are only applicable for symmetric matrices which is not the case if upwind biasing is used in the discretization procedure. To overcome this problem, unsymmetric systems are turned into symmetric ones for example by

creating a larger system of equations from the original coefficient matrix A and its transpose A^T

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \phi \\ \tilde{\phi} \end{pmatrix} = \begin{pmatrix} Q \\ 0 \end{pmatrix} \tag{104}$$

where $\tilde{\phi}$ plays no role. The size of the matrix is increased but symmetry is achieved.

Pressure equation is solved using GAMG. In GAMG the idea is that the solver starts solving the matrix equation from the finest grid level which corresponds to the computational grid. The grid is first coarsened step by step and then refined while the equations are solved on each grid level. There are also several possibilities for how to do the mapping both from a finer to a coarser grid level (restriction) and the other way around (prolongation). The benefit of using several grid levels is that the solution is considerably faster on the coarser grid levels. Furthermore, the solution on a coarser grid level efficiently takes out large-scale errors. Provided that the benefits due to an improved convergence are greater than what mapping the solution between grid levels and creating grids costs in time, the use of the GAMG method is justified. Clearly, the benefits grow with a growing computational grid.

Before mapping any solution from one grid level to another, the solution should be smooth. This is ensured by using an iterative method that produces smooth solutions. According to Ref. [20, Ch. 5.3.8], it does not play a critical role which methods are used, provided that the method fulfills certain conditions. In particular, with elliptic equations such as the pressure equation, using basic iterative methods on the finest level only is ineffective due to properties related to the eigenvalues of such problems. Thus GAMG is a good choise for solving the pressure equation and also `OpenFOAM` documentation recommends to do so.

4 Steady Computations with a Single Blade

The first computations are conducted in the simplest possible case. One propeller in open-water conditions is analysed as a steady-state problem with periodic boundary conditions so that only one of the five blades is included in the grid. There is measured data of thrust and torque coefficients, K_T and K_Q, with 14 values of advance coefficient J for validation of the computed results. This case is extensively used to evaluate the correctness of boundary conditions, solver settings and discretization schemes. As a result, a computational set-up is fixed and subsequently throughly tested. This chapter includes a documentation of the testing phase while initial computations are omitted for the sake of a clearer presentation. The settings tested will be used as a basis in the more demanding cases where trial and error would be more expensive.

The computational domain consists of a volume around one blade. The complete propeller with hub is shown in Figure 7. The part inside the periodic domain is

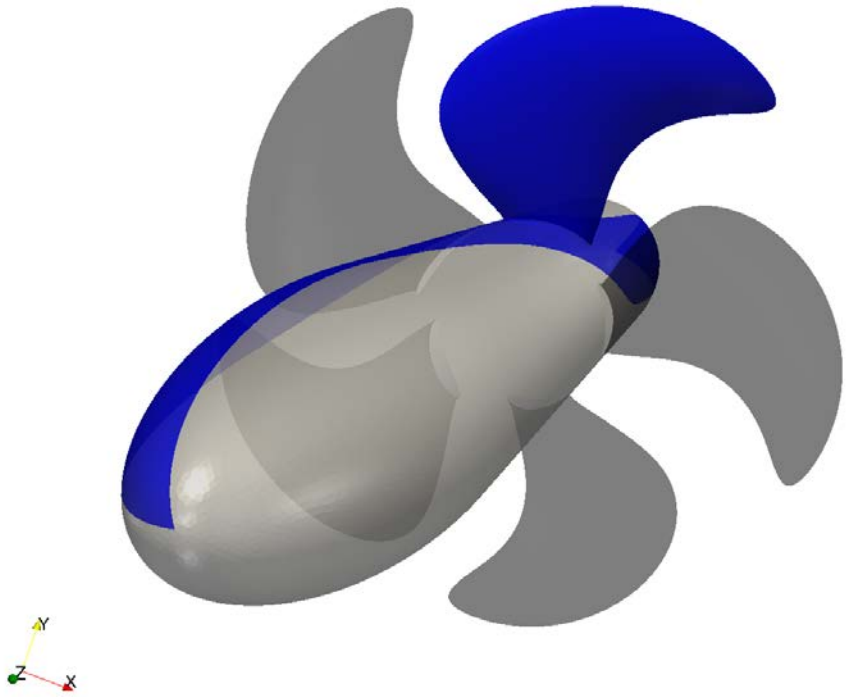


Figure 7: *The complete propeller with the part included in the periodic grid colored in blue.*

colored in blue while the grey color represents the rest of the propeller. The case set-up of the periodic computation is shown in Figure 8. The periodic boundaries, shown in grey, are identical with each other. Inlet and outlet boundaries of the mesh are defined as separate patches so that boundary conditions can be defined on them. Furthermore, the hub and the surrounding boundary (cavitation tunnel wall)

are defined as separate parts of the mesh boundary.

The surface grid on the propeller is shown in Figure 9. The grid is refined close to the leading and trailing edges while the flatter parts of the blade surface are coarser. In order to achieve a better mesh quality, there is a band that follows the leading and trailing edges that allows the construction of a more regular structure of the surface mesh in those regions. Figure 10 shows the volume mesh around the propeller. The subfigures illustrate the internal mesh at different radial positions around the blade. The mesh is mostly tetrahedral but, close to the propeller surface, there are twenty prismatic layers that ensure a sufficient mesh density inside the boundary layer. The boundary layer mesh is best shown in subfigure d) but is present everywhere on the blade surface.

4.1 Boundary Conditions

At the inlet, velocity is fixed to give a desired advance coefficient J. As described in Chapter 1.4.1, the advance coefficient in measurements was evaluated from a velocity measured 150 mm upstream from the propeller plane at a distance of 60 mm from the tunnel wall. Thus reaching exactly the same J value as in measurements was not possible which was, however, accounted for by evaluating an individual advance coefficient in each computation. As can be seen in Figure 11, the J values correspond quite well to the measured ones.

The turbulence quantities were estimated at the inlet by setting turbulence intensity between

$$0.5\% < I = \frac{\sqrt{\frac{2}{3}k}}{|\mathbf{U}_{inlet}|} < 1\% \tag{105}$$

and turbulence viscosity ratio approximately

$$\frac{\nu_T}{\nu} = \frac{k}{\nu\omega} \approx 10 \tag{106}$$

which gives the boundary conditions

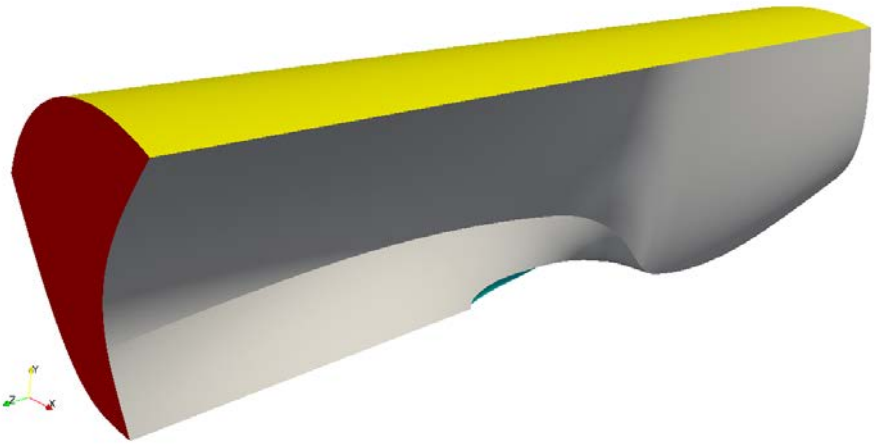
$$k = 0.001\,\mathrm{m}^2/\mathrm{s}^2 \tag{107}$$

$$\omega = 100\,\mathrm{s}^{-1}. \tag{108}$$

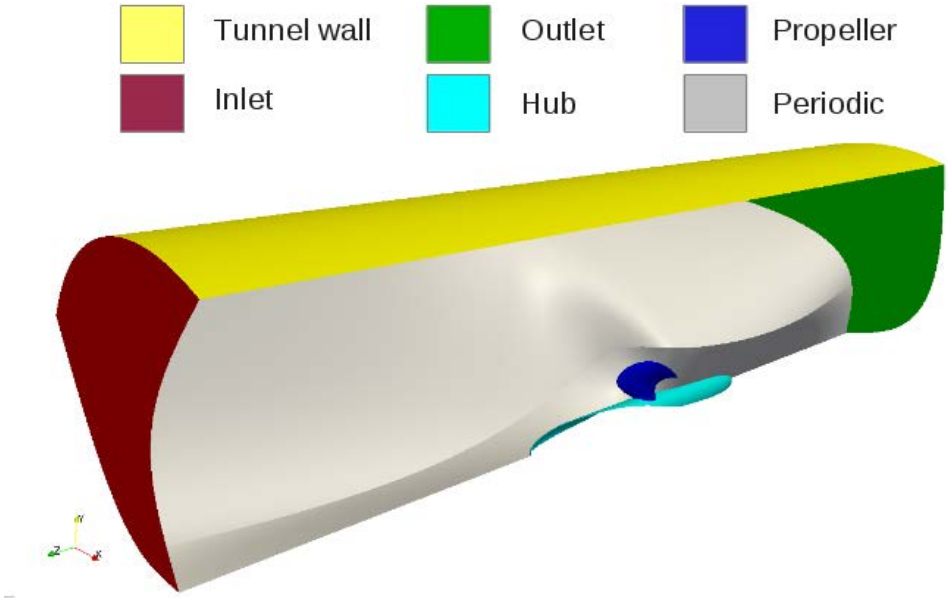
Turbulence handling near walls is as described in Chapter 2.4.

The surface normal gradient of pressure is set to zero ($\nabla p \cdot \mathbf{S}_f = 0$) at all boundaries except for the outlet where pressure is fixed to be zero ($p = 0\,\mathrm{m}^2/\mathrm{s}^2$). Thus, it is the relative (or gauge) pressure that is solved for, which is additionally scaled by density as discussed earlier. All other quantities have the numerical zero-gradient condition at the outlet in order to keep the computation *well defined*.

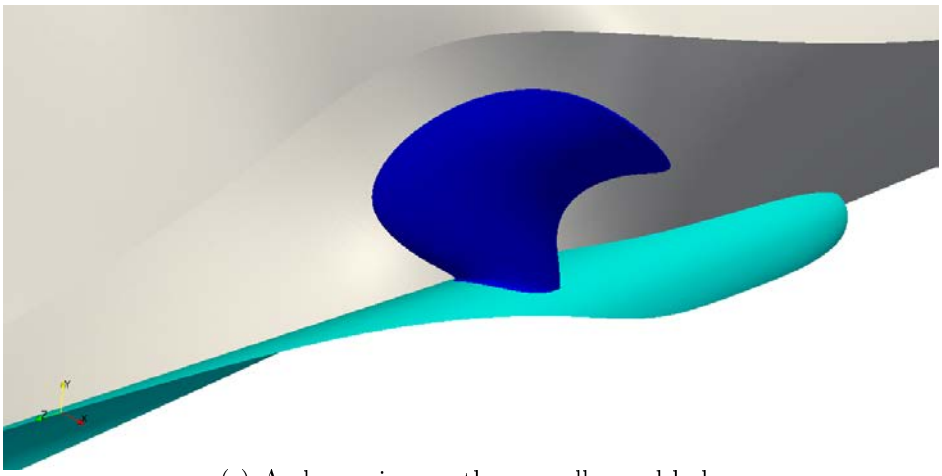
The hub and the tunnel walls were defined as frictionless walls. The velocity component normal to the wall is set to zero in order to prevent flow through it. The velocity component parallel to the surfaces and all other variables have the zero-gradient condition.



(a) Outer boundaries.

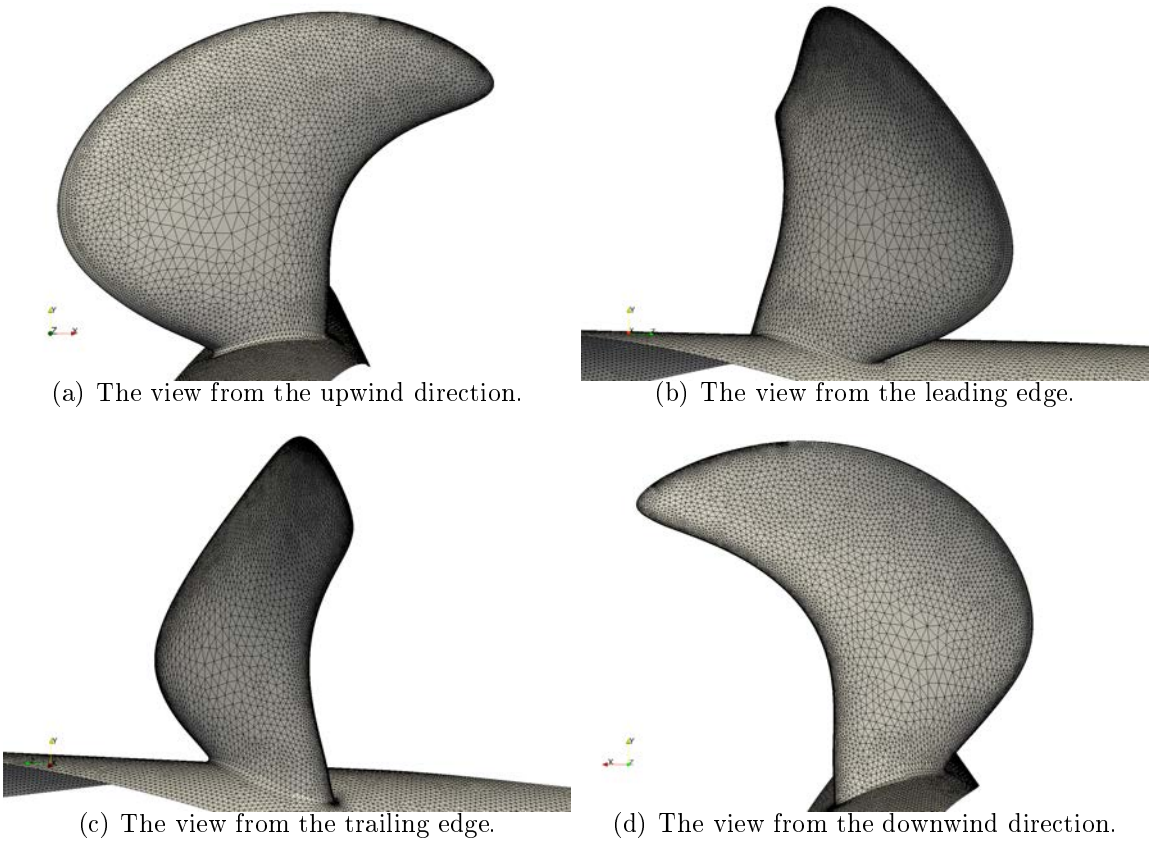


(b) A view with the foremost periodic boundaries hidden.



(c) A closer view on the propeller and hub.

Figure 8: *The case set-up in the steady-state computations with one blade only. Boundary names are given in part b).*



(a) The view from the upwind direction.
(b) The view from the leading edge.
(c) The view from the trailing edge.
(d) The view from the downwind direction.

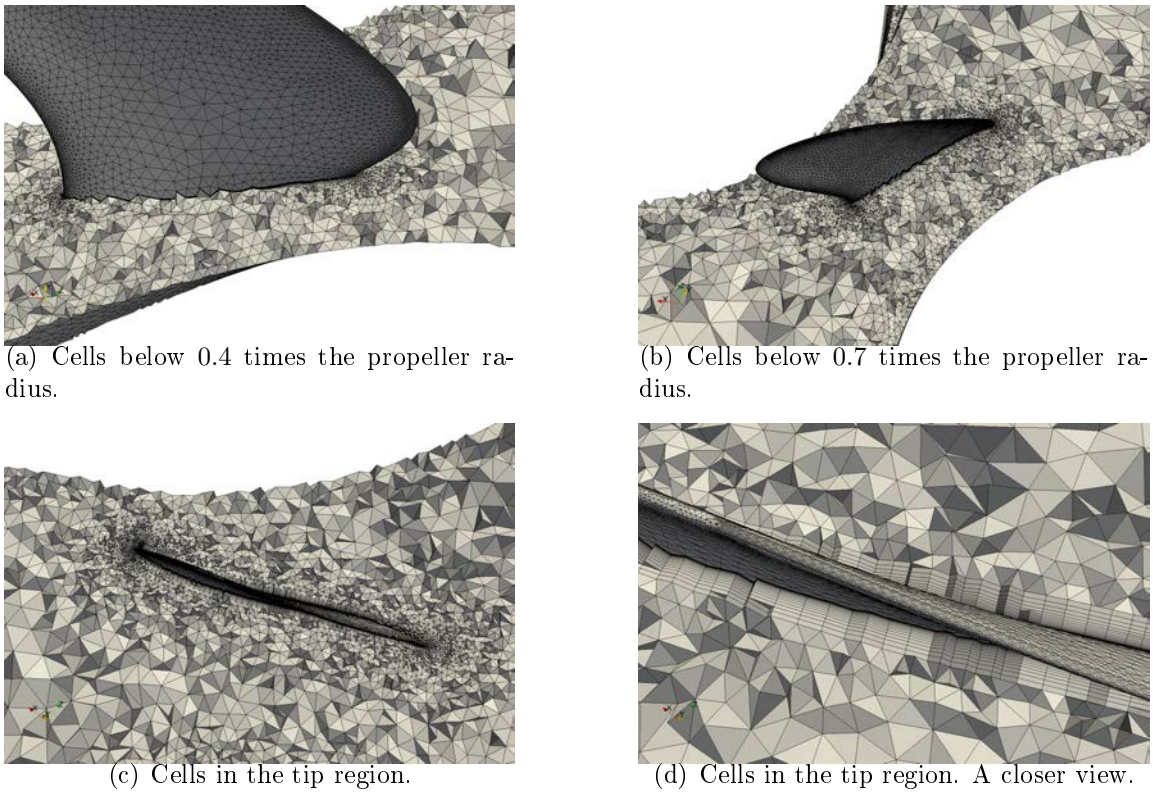
Figure 9: *Surface grid on the propeller surface.*

4.2 Results

The case was computed with several discretizations. Most of the results are not presented here, since they only served as initial tests to give understanding of how different settings affect the computation. With most settings, the force coefficients were within 2 % of each other which tentatively implies that the mesh is adequate and the computation is also otherwise set up correctly.

4.2.1 Force Prediction

With a fixed computational set-up, the performance of the settings was evaluated by comparing the thrust and torque coefficients, K_T and K_Q, against the experimental data over a range of operating points (see Figure 11). Since the correct advance coefficient was difficult to set precisely and the computational results are not in perfect correspondence with the experimental data, an individual advance coefficient was calculated for each computation. The velocity was probed at the location where it is measured with a Prandtl tube in the test facility. By eye, one can draw a conclusion that the results at lower J-values are often slightly overpredicted compared to the experimental data. The evaluation of J, however, reduces the overprediction. All forces lie within 3 percent of the experimental data except for the lowest value



(a) Cells below 0.4 times the propeller radius.

(b) Cells below 0.7 times the propeller radius.

(c) Cells in the tip region.

(d) Cells in the tip region. A closer view.

Figure 10: *Volume mesh around the propeller.*

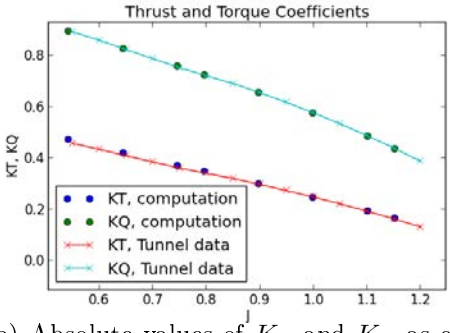
$J = 0.55$. All torques are predicted within one percent from the measurements.

Looking at the relative differences between the computational results and measurements also shown in Figure 11 shows that the torque differs by less than 0.5% from the measurements over the entire range of advance coefficients. Thrust behaves somewhat differently. The difference is the smallest near the propeller design point ($0.9 < J < 1.0$) and grows in other parts of the performance curve. A behaviour like this could be expected since the flow over the blade is relatively smooth near the design point as compared to lower or higher J values. Farther from the design point, the angle of attack of the incoming flow changes which is expected to set more requirements for the computation and increase uncertainties.

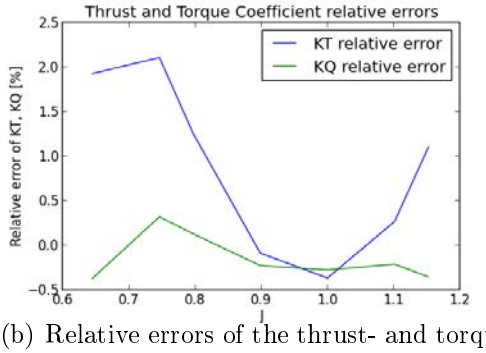
The fact that the results match so well with the measurements is a little bit surprising due to several reasons. First of all, the computational and the measurement set-ups are not identical. The hub in the computations has a cigar-like form while in the measurements the hub continues further upstream. Thus, the incoming flow must be somewhat different in the computations than in the measurements which should affect the force prediction. Secondly, there are simplifications in the computational model such as the `MRF`, frictionless walls and the turbulence model that differ from the real world physics.

Second International Symposium on Marine Propulsors held in Hamburg in 2011 comprised two cases where a propeller was analyzed in open-water conditions. The first case was a propeller in a towing tank and the second case was based on mea-

surements of the same propeller in a cavitation tunnel. More detailed descriptions of the results are described in Ref. [27]. Apart from different measuring equipment, the hub geometries differed between the cases and the measured forces in the cavitation tunnel case had no effects due to cavitation. The measured thrusts differed between the two cases by only about 1%, so the effect of the experimental set-up seems not to be very dominant. Thus, the force prediction presented in this chapter is credible and shows that the computational model used reproduces propeller forces within an accuracy of a few percent. Furthermore, it also predits the trends properly with a varying advance coefficient and is in line with results presented in literature that are based on similar computational methods.



(a) Absolute values of K_T and K_Q as compared against experimental data over a range of advance coefficient values J.



(b) Relative errors of the thrust- and torque coefficients relative to measurements over a range of advance coefficient values J.

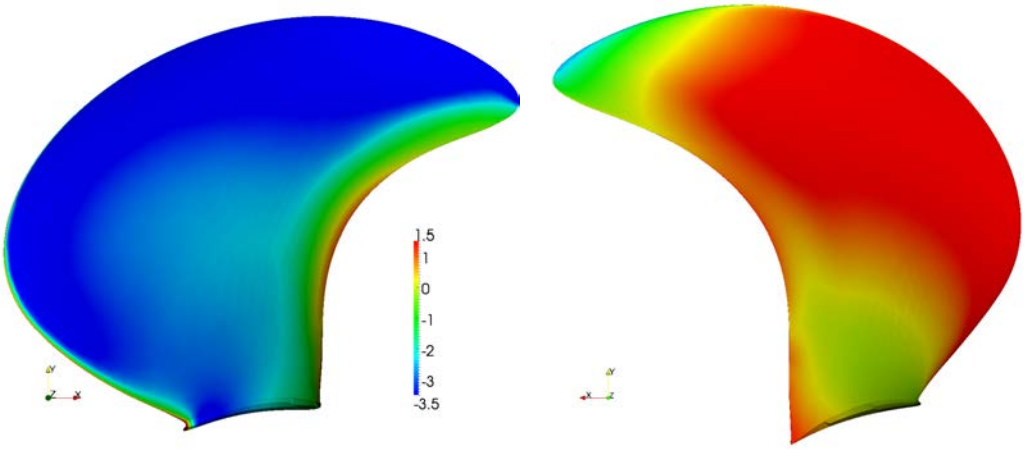
Figure 11: *Open-water performance.*

4.2.2 Distributions on the Blade Surface

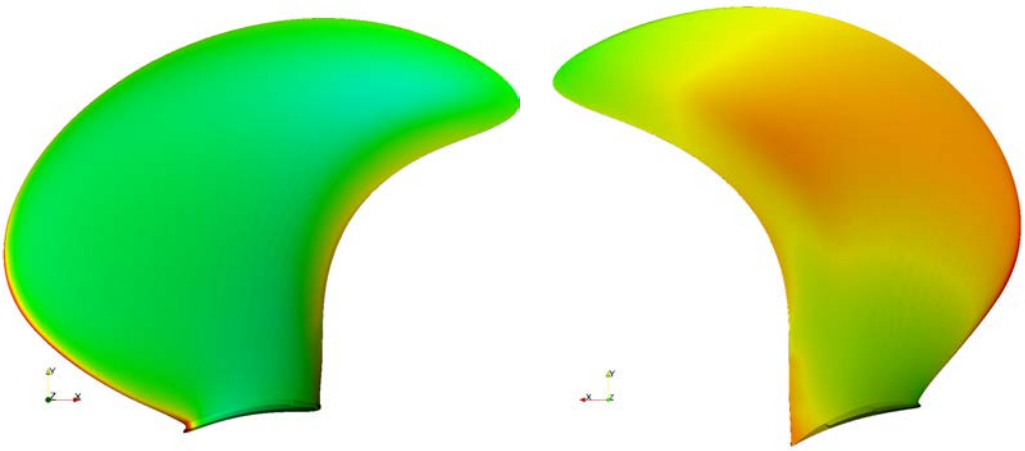
Three operating points, $J = 0.65$, $J = 0.90$ and $J = 1.15$, are inspected in more detail. The pressure coefficient (C_p) distributions on the pressure- and suction sides with all three values of J are presented in Figure 12. Clearly, the loading is larger at the lower J-values which is in line with general knowledge and the open-water curve (Figure 11). The forces acting on propeller increase with a decreasing advance coefficient.

Turbulent viscosity (ν_T) scaled by the molecular kinematic viscosity ν at a distance of 0.5% times the propeller diameters from the blade surface is given in Figure 13. In an incompressible flow, the turbulent viscosity is the only turbulence variable that shows up in the conservation equation for momentum (or continuity). It acts through the diffusion term and increases the viscous forces. The turbulent viscosity seems to increase with an increasing advance coefficient J on the pressure side of the blade and to decrease on the suction side of the blade. This is explained by the fact that the stagnation point moves towards the pressure side when J decreases. Thus at a low J, the flow is smoother on the pressure side, but passes a tighter turn over the leading edge before reaching the suction side which can be seen as an increased tendence to turbulence. With an increasing J, the opposite happens.

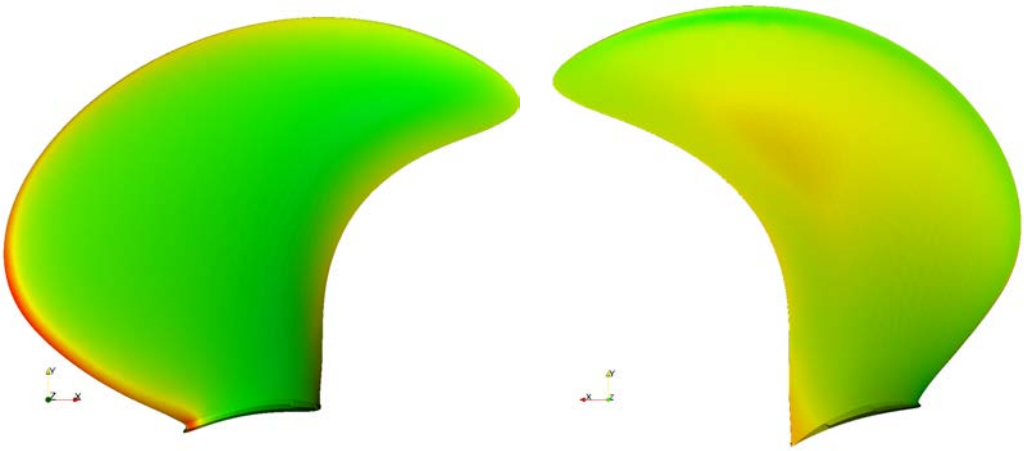
Figure 14 shows wall bounded streamlines on the propeller blade. The surfaces are colored by the friction factor (C_f), that is based on the magnitude of the wall shear stress. The streamlines show that the flow separates close to the trailing edge with all of the three advance coefficients. The separation is shown as a divide in the streamlines and is present on both the suction and pressure side of the propeller. With a decreasing advance coefficient, the tendence for separation on the suction side seems to increase which is expected due to the increased angle of attack of the blade. Apart from areas close to the trailing edge, the flow stays attached to the surface. It could be that the separation should increase more with advance coeffients far from the design point of the propeller and that the current method fails to reproduce the behaviour. It would explain the differences in the errors of the force coefficients with a varying J. A more detailed analysis would require experimental data on the flow field.



(a) $J = 0.65$, suction side. (b) $J = 0.65$, pressure side.

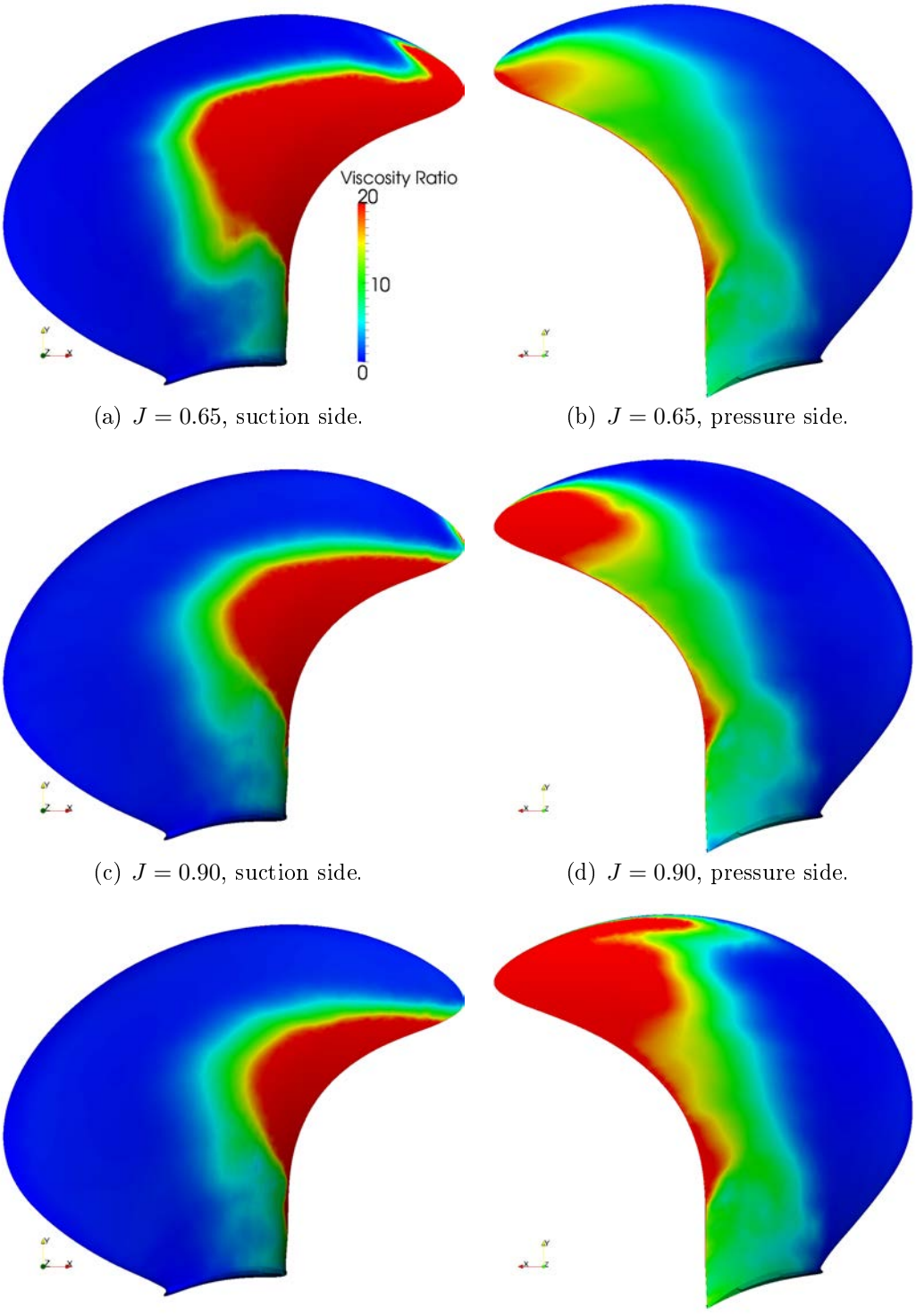


(c) $J = 0.90$, suction side. (d) $J = 0.90$, pressure side.



(e) $J = 1.15$, suction side. (f) $J = 1.15$, pressure side.

Figure 12: *Pressure coefficient C_p on the blade surface.*



(a) $J = 0.65$, suction side.

(b) $J = 0.65$, pressure side.

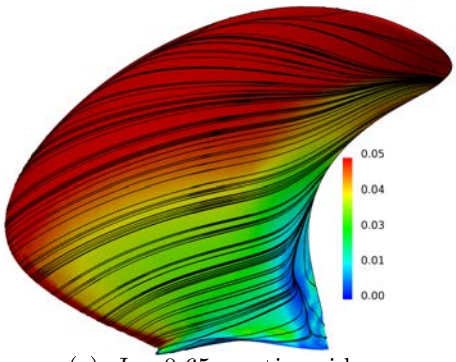
(c) $J = 0.90$, suction side.

(d) $J = 0.90$, pressure side.

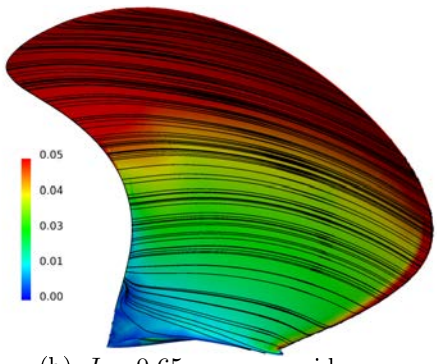
(e) $J = 1.15$, suction side.

(f) $J = 1.15$, pressure side.

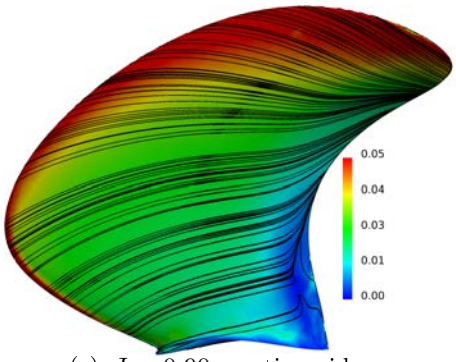
Figure 13: *Turbulent viscosity ratio ν_T/ν on the blade surface.*



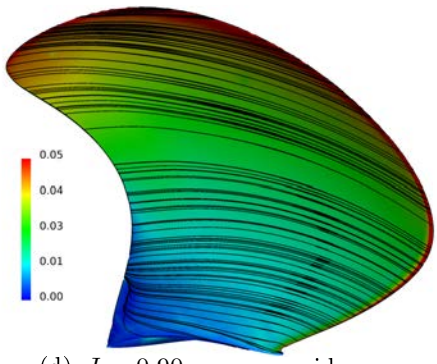
(a) $J = 0.65$, suction side.



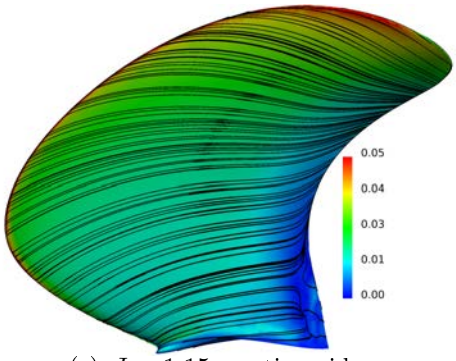
(b) $J = 0.65$, pressure side.



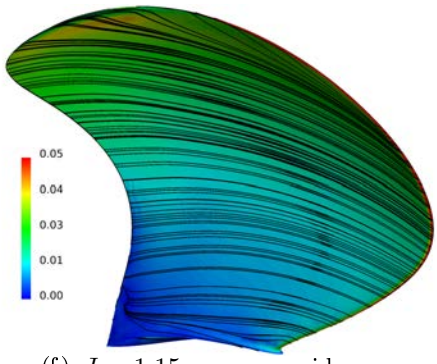
(c) $J = 0.90$, suction side.



(d) $J = 0.90$, pressure side.



(e) $J = 1.15$, suction side.



(f) $J = 1.15$, pressure side.

Figure 14: *Friction factor C_f and streamlines on the blade surface.*

4.3 Grid Convergence

The grid convergence was studied using four different grid densities. Compared to the base mesh described earlier, one coarser and two finer meshes were created. The modifications to the base mesh were made in the surface mesh and in the cell sizes normal to the blade surface. Between two adjacent grid levels, there is a factor of 4/3 in the first cell height and in the number of points along each edge on the surface. The main parameters of each mesh are given in Table 1. The meshes are numbered so that the coarsest mesh is indexed as mesh No. 1 and the finest mesh is mesh No. 4. Thus, the base mesh used earlier corresponds to mesh No. 2.

Table 1: *Main parameters of the four meshes used in the grid convergence study.*

	1	2	3	4
total number of cells	2 728 626	3 095 947	3 704 111	4 652 284
surface mesh	24 226	33 605	50 552	74 767
1. cell height	$1.6 \cdot 10^{-5}$	$1.18 \cdot 10^{-5}$	$9.0 \cdot 10^{-6}$	$6.75 \cdot 10^{-6}$

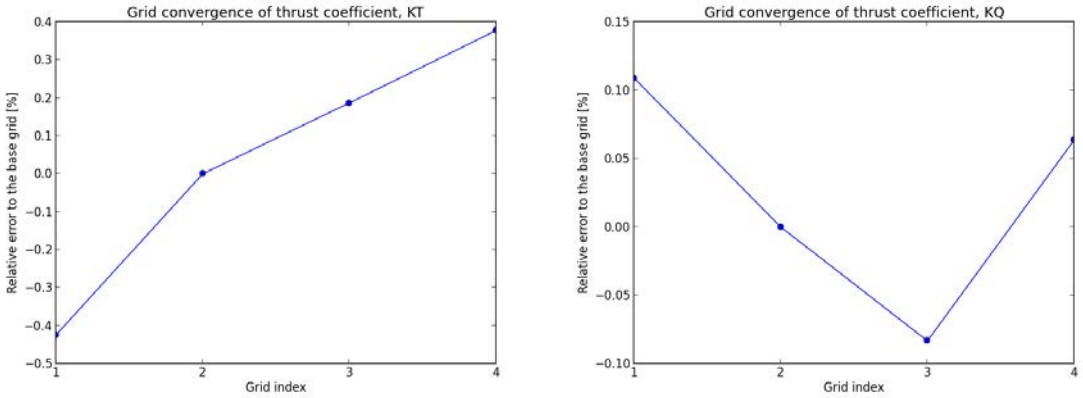
The periodic case was computed at $J = 0.90$ with these four grids and the results are given in Figure 15. Generally in grid convergence studies, the results are estimated on an infinitely fine grid by means of extrapolation based on results obtained from different grid levels. One way is the Richardson extrapolation recommended by ITTC [28]. In order to conduct the extrapolation, at least three grid levels need to be concerned. Additionally, the behaviour of the values from one grid level to another should monotonically approach some level so that the difference between two adjacent grid levels always decreases when going towards the finer grids. This is not the case with the torque coefficient, K_Q, as can be seen in Figure 15. In such cases one can estimate the uncertainties by simply bounding the expected value between the smallest and the largest values [28]. It is, however, debatable if there could be new extreme values with finer grids than those used here.

Thrust coefficient K_T does behave monotonically but it does not show any asymptotic behaviour over the entire range of the grid refinement levels. In this case, there are no statistical methods that would allow the estimation of the uncertainties without conducting new computations with more grids. Thus, it is simply stated that the grid-based error in the base grid (grid number 2) is probably in the order of one or two percent. All of the four grids are relatively fine as will be seen from the y^+ distributions and thus there is no reason to expect that the results would change considerably from those obtained with these grids. With new grids, least-squares based methods discussed by for example Eça and Hoekstra in [29] could be applied to find a more reliable estimate for the grid-based uncertainty.

There are several possible reasons for the non-asymptotic behaviour. For example, a change in the behaviour of the turbulence model may lead to noisy behavior of the data. Furthermore, the fact that grid density is represented by only one value assumes that all grids are geometrically similar i.e. that the skewness and orthogonality properties of the cells are the same in all grids. In the case of an unstructured

grid this is not the case, which reflects on the results [29]. As an example, non-orthogonalities in the grid directly affect the computation of laplacian terms that play an important role in solving the pressure field. Pressure, on the other hand, dominates propeller forces.

The results may also change due to geometrical differences between the grids. Since the surface grid is different on each grid level, also the surface area differs between the grids and the surface integration is performed differently with each grid. The differences in surface areas were found to be less than 0.2% between all of the grids so the influence of the area is quite negligible.

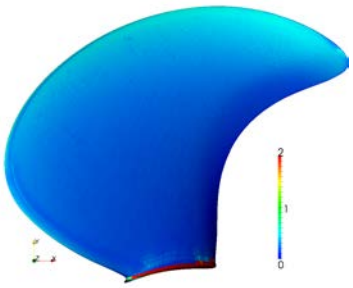


(a) Thrust coefficient, K_T. (b) Torque coefficient, K_Q.

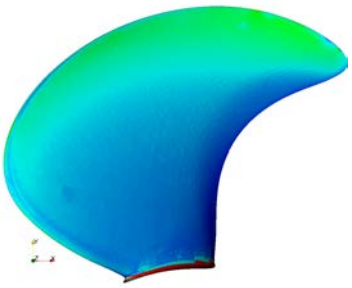
Figure 15: *Force coefficients computed with four different grids.*

The y^+ values in the first cells next to the wall play an important role in the quality of the results. In Figures 16 and 17, there are the distributions of the y^+ values on the blade computed with all of the four grids on the suction and the pressure sides, respectively. With the finest grid, the y^+ values are well below one so the surface normal density of the grid is adequate. Up to the second coarsest grid (grid number 3), y^+ is mostly below 2, which is usually considered fine enough. In the coarsest mesh, there are significant regions where $y^+ > 2$. The contours present the discrete distibutions of the values without any interpolation that would smooth the field in order to emphasize the differences in the grids.

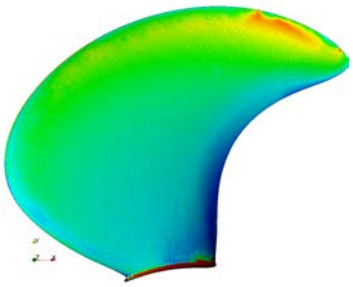
The grid dependence study shows that there is dependence on the grid in the solutions, but it is relatively small. The order of magnitude of the errors is approximately 1% and in the case of the torque coefficient, it is even smaller. The error being small, it can be expected that the base grid is adequate and similar meshes can be used to study the effects caused by a hybrid mesh and time discretization.



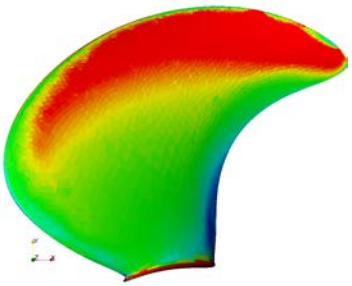
(a) Grid 1, suction side.



(b) Grid 2, suction side.

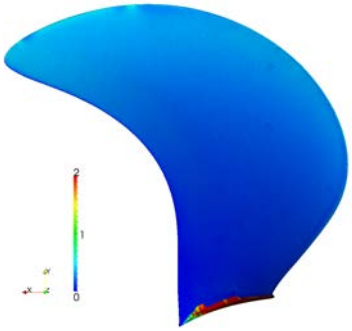


(c) Grid 3, suction side.

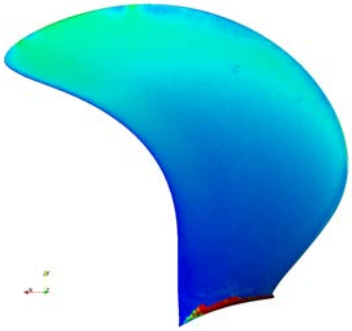


(d) Grid 4, suction side.

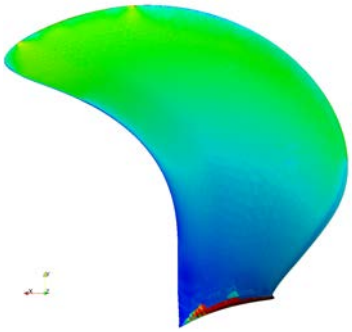
Figure 16: y^+ *on the suction side with four mesh densities at* $J = 0.90$.



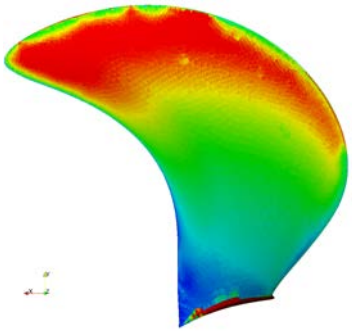
(a) Grid 1, suction side.



(b) Grid 2, suction side.



(c) Grid 3, suction side.



(d) Grid 4, suction side.

Figure 17: y^+ *on the pressure side with four mesh densities at* $J = 0.90$.

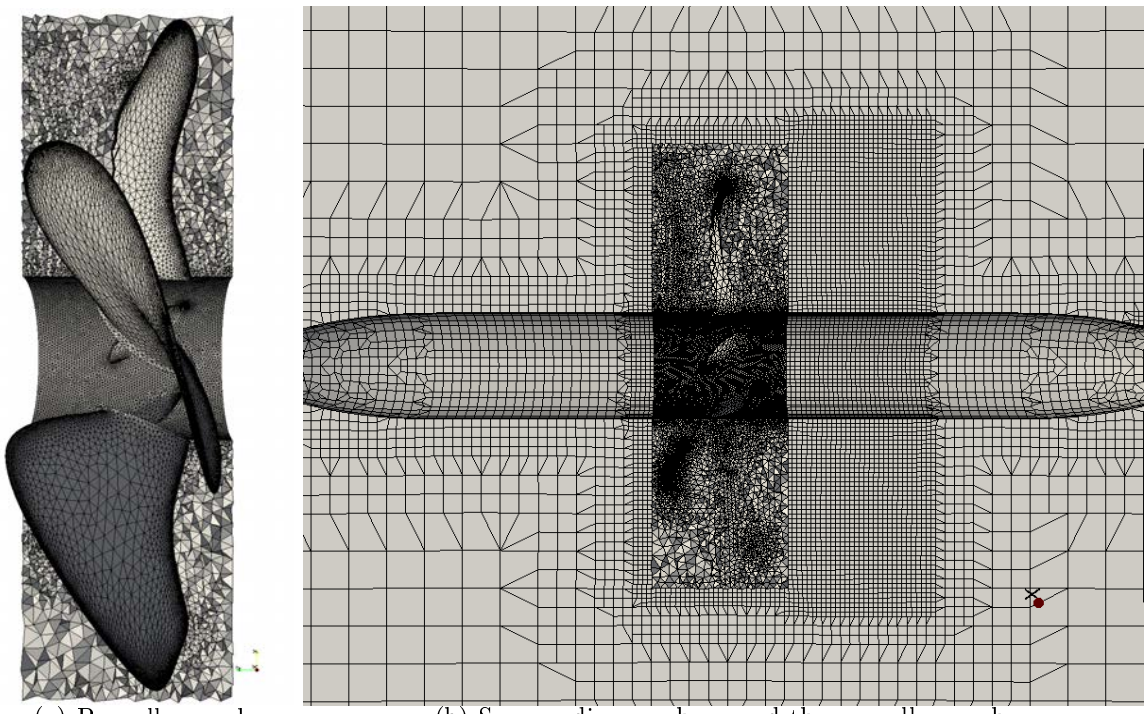
5 Steady Computations with the Hybrid Mesh

The second computed case is the same as that analysed previously but the mesh consists of the two following parts:

- a cylidrical volume around the propeller with a mesh similar to that used earlier (referred to as *propeller mesh*) and
- a hexahedral mesh around the propeller mesh (referred to as *surroundings*).

The propeller mesh was created with the same template as the mesh used earlier and consists, for the most part, of tetrahedral cells, but has prismatic cells in the boundary layer. The tetrahedral propeller mesh was cut close to the propeller so that it confines a cylindrical volume. Both the `MRF` and the moving mesh require that the boundary faces of the propeller mesh have no component in the direction of the rotational movement so the cylindrical shape is adequate in that sense.

The surrounding mesh was created using `snappyHexMesh` that is part of the `OpenFOAM` distribution. A hole for the propeller mesh was left inside the surrounding part so that the two separate meshes could be merged with the mesh handling tools available in `OpenFOAM`. The hybrid mesh is illustrated in Figure 18.



(a) Propeller mesh. (b) Surrounding mesh around the propeller mesh.

Figure 18: *Hybrid mesh.*

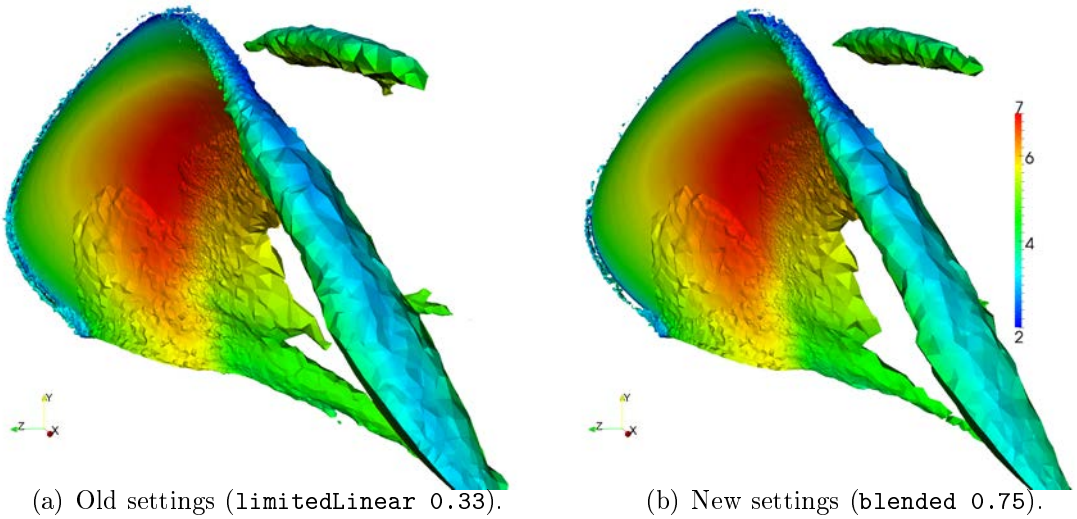
One main objective in the computations with the hybrid mesh is to inspect how a discontinuity in the mesh with the Arbitrary Mesh Interface (AMI) between the regions affects not only the thrust and torque coefficients, but also the propagation

of flow structures downstream of the propeller. The correct propagation of the wake is important when there are multiple propellers behind one another.

5.1 Stability

With the hybrid mesh, the computation was not as stable as earlier and different discretizations were needed. The instability was traced back to the solution of the momentum equation. A stable computation was achieved by discretizing the convection of momentum with the `blended` scheme with a ratio of 0.75 instead of `limitedLinear` that was used earlier.The new method is more dissipative but also considerably more robust. Probably the main reason for the additional robustness is the fact that the limiter does not depend on the solution itself as in the case of the `limitedLinear` scheme. Clearly, due to a greater contribution of the first-order upwind, diffusivity is increased.

In order to evaluate the performance of the new settings, they were tested in the periodic case and compared to earlier results and measurements. In Figure 19, there are iso-volumes of vorticity ($\nabla \times \mathbf{U} = 300\ \frac{1}{\mathrm{s}}$) computed with the old and the new settings. One can see that the new settings produce less vorticity. The tip vortex is slightly thinner and also less vorticity emanates from the central parts of the blade. No considerable difference can be seen in the distance that the tip vortex is transported. The decrease in vorticity could be expected due to the higher diffusivity of the new settings but these results imply that no dramatic losses are caused in the prediction of propeller wake if the `blended 0.75` scheme is used.



(a) Old settings (`limitedLinear 0.33`). (b) New settings (`blended 0.75`).

Figure 19: *Iso-volumes of vorticity* $\omega = 300\ \frac{1}{\mathrm{s}}$ *in the periodic case with the old and the new settings. Colored by velocity magnitude.*

In addition to the propagation of the tip vortex, also the force coefficients were evaluated with the new settings. They change from those computed with the `limitedLinear 0.33` scheme. The relative differences to both earlier results and to measurements (relative error) of the thrust and torque coefficients, K_T and K_Q, at

$J = 0.65$, 0.90 and 1.15 are given in Tables 2 and 3, respectively. The force coefficients change by less than 1% from the previous results which can still be considered acceptable concerning the overall accuracy of the simulation. In summary, the new settings yield larger errors in the force coefficients but they are, on the other hand, considerably more robust. Thus, they provide an adequate computational tool for the purposes of this study.

Table 2: *Comparison of thrust coefficient, K_T, computed with the* `blended 0.75` *scheme ($K_{T,BL}$) to that computed with the* `limitedLinear 0.33` *scheme ($K_{T,LL}$) and to the measured values ($K_{T,M}$). Computations conducted with the periodic grid.*

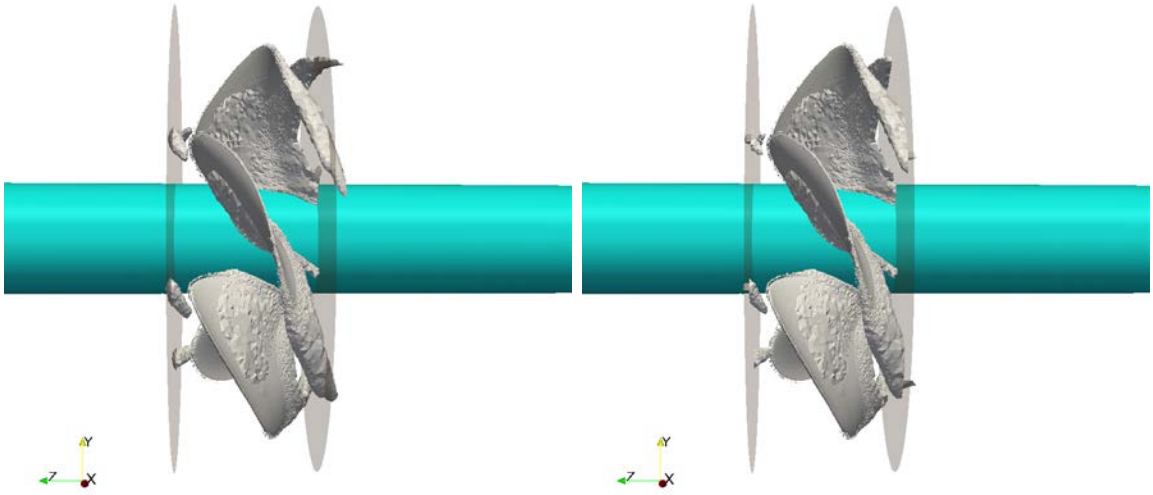
J	Relative difference $\left(\frac{K_{T,BL}-K_{T,LL}}{K_{T,LL}}\right)$	Relative difference $\left(\frac{K_{T,BL}-K_{T,M}}{K_{T,M}}\right)$
0.65	0.81	3.50
0.90	0.40	0.64
1.15	0.86	1.24

Table 3: *Comparison of torque coefficient, K_Q, computed with the* `blended 0.75` *scheme ($K_{Q,BL}$) to that computed with the* `limitedLinear 0.33` *scheme ($K_{Q,LL}$) and to the measured values ($K_{Q,M}$). Computations conducted with the periodic grid.*

J	Relative difference $\left(\frac{K_{Q,BL}-K_{Q,LL}}{K_{Q,LL}}\right)$	Relative difference $\left(\frac{K_{Q,BL}-K_{Q,M}}{K_{Q,M}}\right)$
0.65	1.07	1.22
0.90	0.72	0.72
1.15	1.15	0.39

5.2 Comparison of MRF Domains

In order to further analyze how the `MRF` method is to be used, the hybrid mesh case is computed using two different `MRF` domains. In the first case, `MRF` is only applied inside the propeller mesh, and, in the second case, it is active everywhere in the mesh and thus similar to earlier computations. Figure 20 illustrates iso-volumes of vorticity, $\omega = \nabla \times \mathbf{U} = 300\ \frac{1}{\mathrm{s}}$ in order to track the tip vortex emanating from the propeller. In the case illustrated on the right-hand-side, `MRF` is only applied inside the propeller mesh and, in that region, the two solutions are very close to each other. In the left-hand-side case where `MRF` is applied in the entire mesh, the tip vortex is transported through the mesh boundary. The mesh boundaries are shown as the shaded grey surfaces on the upstream and downstream sides of the propeller. A small change can be seen in the vortex when it penetrates the mesh boundary, but it is nevertheless transported through the interface quite nicely. If the `MRF` is deactivated outside the propeller mesh, however, the vortex immediately dies at the boundary. It seems that the mesh interface itself does not affect the solution as much as the `MRF` boundary. The reason for the vortex dying at the `MRF` boundary is discussed later.



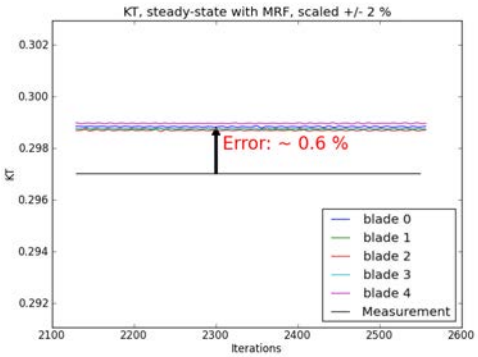
(a) `MRF` method is applied in domain 2. Outer disturbances are not shown.

(b) `MRF` method is applied inside the propeller mesh.

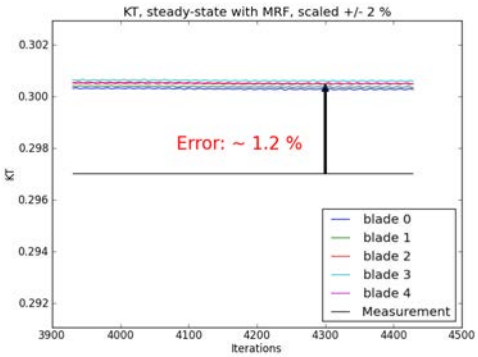
Figure 20: *Iso-volumes of vorticity with the* ***MRF*** *method applied in two different domains.*

The thrust and torque coefficients computed with the hybrid mesh are shown in Figure 21. Applying the `MRF` only inside the propeller mesh yields lower thrust and torque coefficients which, compared to measurements, have errors of 0.5 % and 0.6 %, respectively. The results are also in line with the periodic computations with the same settings. However, applying `MRF` everywhere with the hybrid mesh gives different results. The force coefficients become larger and have errors of about 1.2 % (K_T) and 1.3 % (K_Q).

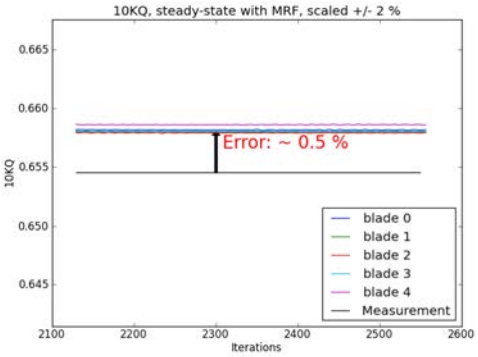
In either computation, there are no significant differences between single blades (approximately 0.1 %) which is expected and also an important implication of how well the computational solution method works. The mesh introduces some asymmetry to the computation but the case itself is symmetrical so that all blades should yield the same results. Since the propeller mesh is identical around each blade, all differences must be due to the numerical solution procedure and to the asymmetry in the surrounding mesh and the mesh interface. The variation in the coefficient values between different iterations is yet by an order of magnitude smaller.



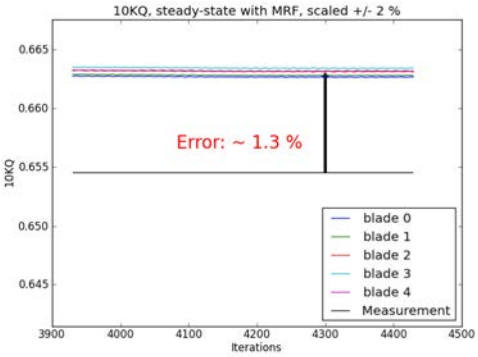
(a) `MRF` method is applied in the entire mesh. Thrust coefficient, K_T.



(b) `MRF` method is applied inside the propeller mesh. Thrust coefficient, K_T.



(c) `MRF` method is applied in the entire mesh. Torque coefficient, 10 K_Q.



(d) `MRF` method is applied inside the propeller mesh. Torque coefficient, 10 K_Q.

Figure 21: *Force coefficients (K_T and K_Q) with the `MRF` method applied in two different domains at advance coefficient $J = 0.90$.*

Contours of tangential velocities at adjacent cross-sections behind the propeller are studied in order to see how the propeller wake propagates downstream. An overall picture of the contours under focus and their locations are shown in Figure 22.

The cross-sections are shown separately in Figure 23. The first three contours at $z = 0.025$ m, $z = 0.030$ m and $z = 0.035$ m are from inside the propeller mesh and the three latter contours at $z = 0.045$ m, $z = 0.050$ m and $z = 0.055$ m from the downstream side of the propeller mesh boundary. In the first contours, there are small flow structures near the blade tip that imply the existence of the tip vortex. Further downstream they gradually become weaker. The same trend can be seen in the overall flow structure. Velocity peaks become smoother towards downstream and thus the high velocity region grows and becomes more uniform. The results shown here are from the computation with `MRF` applied inside the propeller mesh only.

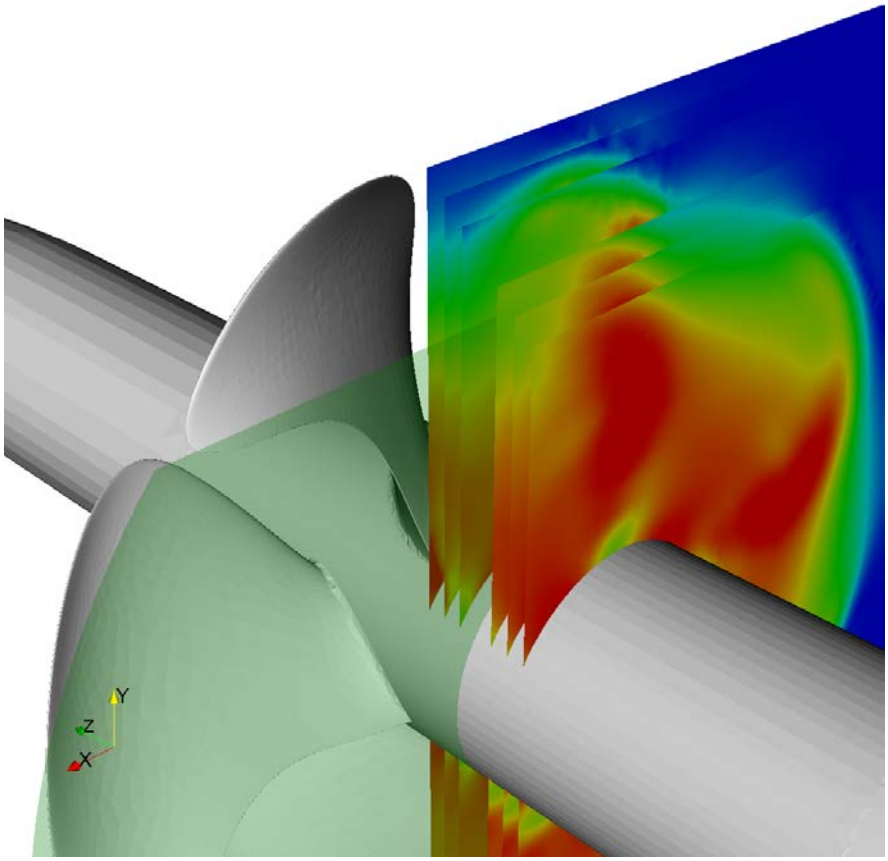
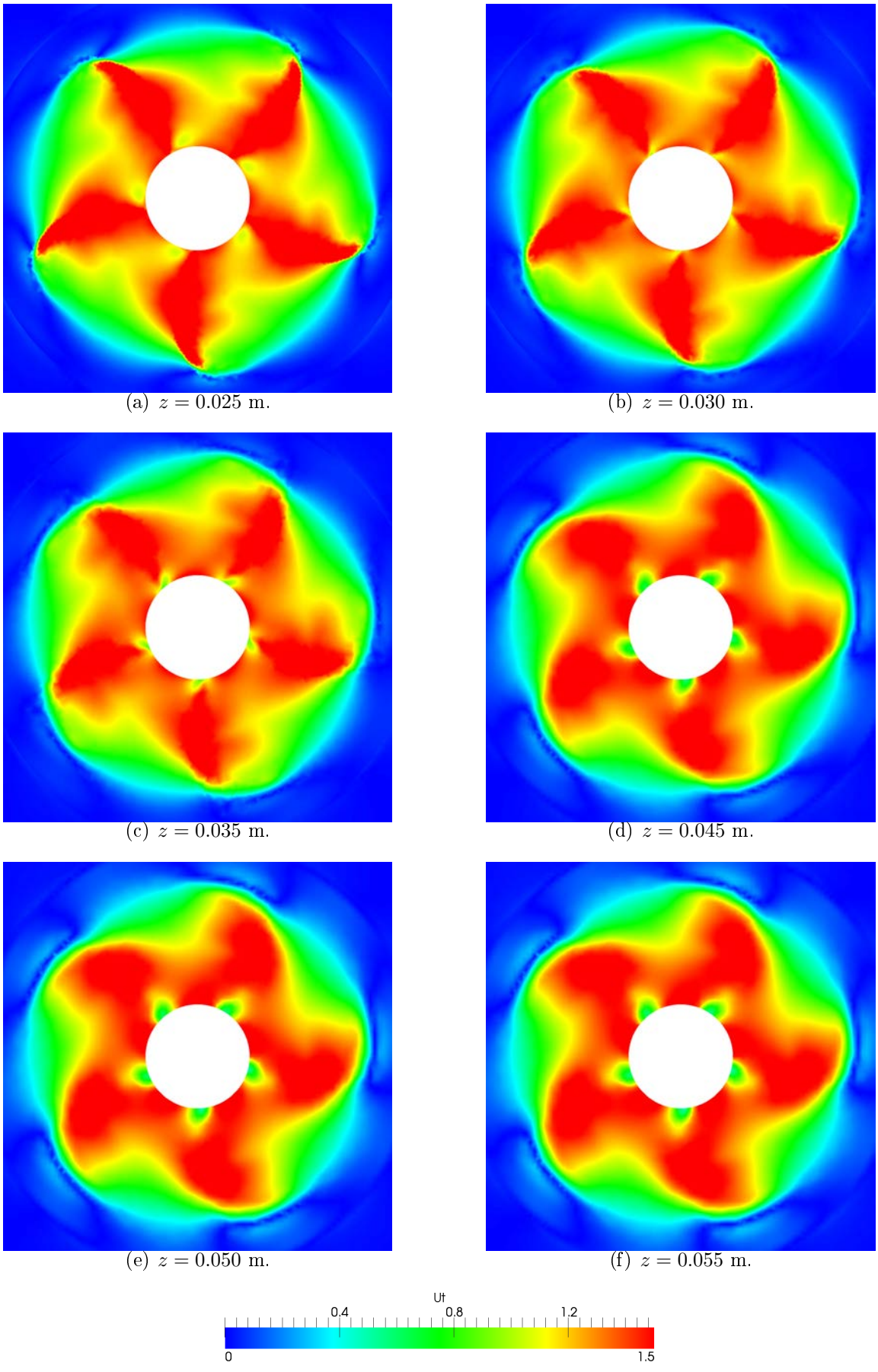


Figure 22: *Cross-sections behind the propeller. The position of the propeller mesh boundary is shown in green.*

Tangential velocities at different radial positions behind one propeller blade are plotted in Figure 24. The picture on the left-hand-side shows the velocities in the case when the `MRF` is only applied inside the propeller mesh. In the right-hand-side picture, the `MRF` has been applied in the entire mesh. The location of the propeller mesh boundary (and the `MRF` boundary in the first case) is shown as a vertical line (at $z = 0.04$ m). There is a slight notch in all of the velocity profiles across the mesh boundary but the overall form of the profiles does not seem to be disturbed very much. Clearly, an interpolation at the mesh discontinuity must have some kind of an effect on the velocities in a general case but the results show that it is not very large.

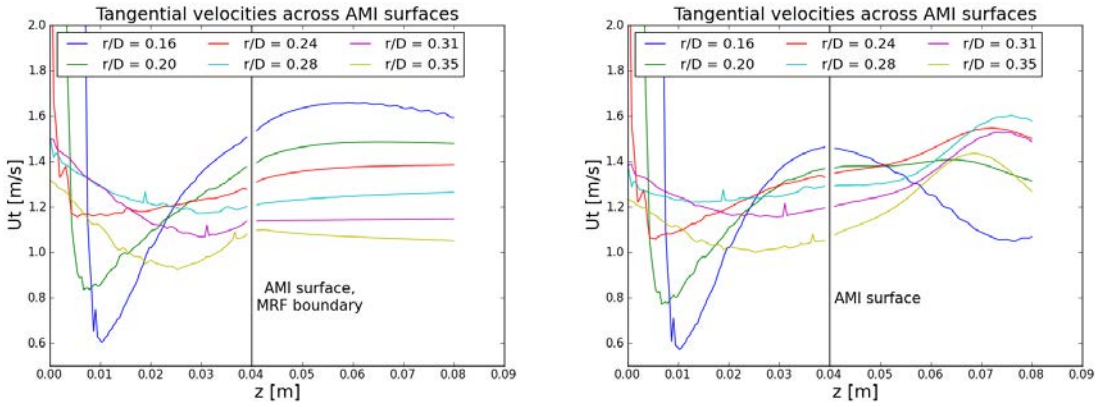


(a) $z = 0.025$ m. (b) $z = 0.030$ m.

(c) $z = 0.035$ m. (d) $z = 0.045$ m.

(e) $z = 0.050$ m. (f) $z = 0.055$ m.

Figure 23: *Tangential velocities behind the propeller.* `MRF` *applied in the propeller mesh.*



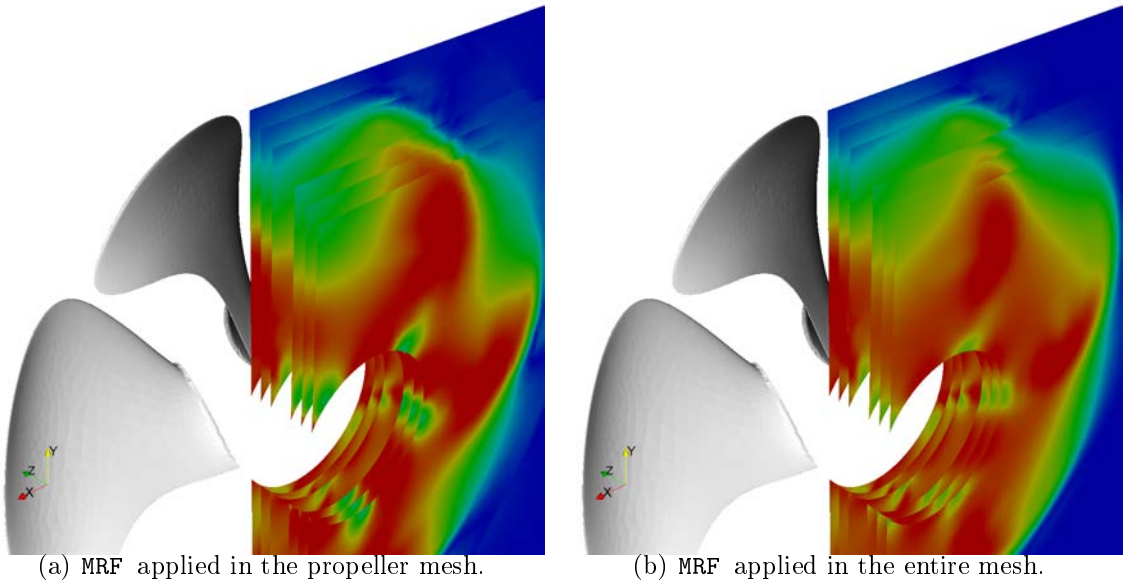
(a) `MRF` applied in the propeller mesh. (b) `MRF` applied in the entire mesh.

Figure 24: *Tangential velocities behind the propeller at several radial positions.*

A much clearer trend can also be seen in the same figure. When the `MRF` is active the flow field is convected by the *relative* velocity which is seen as a different behaviour of velocities at different radial positions. In the case where `MRF` is only applied inside the propeller mesh, it is the *absolute* velocity that convects flow structures outside of the propeller mesh. In that case the flow is transported downstream more or less as it is on the `MRF` boundary, since the axial velocity dominates over other components.

The same effect is seen in the velocity fields at different cross-sections. The contours from Figure 23 and corresponding contours from the case of the `MRF` applied everywhere are put together into Figure 25. The last three cross-sections in the two cases show the different behaviour of the overall flows. If `MRF` is applied, the flow rotates against the propeller rotation (counter clock-wise) when going downstream which is caused by the relative velocity being responsible for convection. In the part where no `MRF` is applied, the rotation changes direction and is much weaker, hardly visible in still pictures. An animation makes the behaviour more apparent.

At the last cross-sections in both cases, the velocity fields differ from each other. In the case of `MRF` applied everywhere, the velocity differences are much smoother than in the case of `MRF` applied only inside the propeller mesh. A possible reason for this is grid based diffusion that is stronger if the flow direction is not aligned with the cells. The mesh created with `snappyHexMesh` is aligned with the coordinate axes (x, y, z), so removing `MRF` at the mesh boundary makes the flow more aligned with the cells and thus reduces numerical diffusion. On the other hand, numerical diffusion only differs between the cases outside of the propeller mesh so it is probably not the main reason for the different results near the propeller. Figure 26 shows that the difference is present already inside the propeller mesh and upstream of it. The tangential velocities are illustrated in both sides of the upstream side boundary of the propeller mesh and the difference seems to be present already upstream of the propeller mesh. The differences in the velocities before the propeller mesh imply that the upstream side of the propeller mesh is probably not far enough from the propeller which would cause the differences.

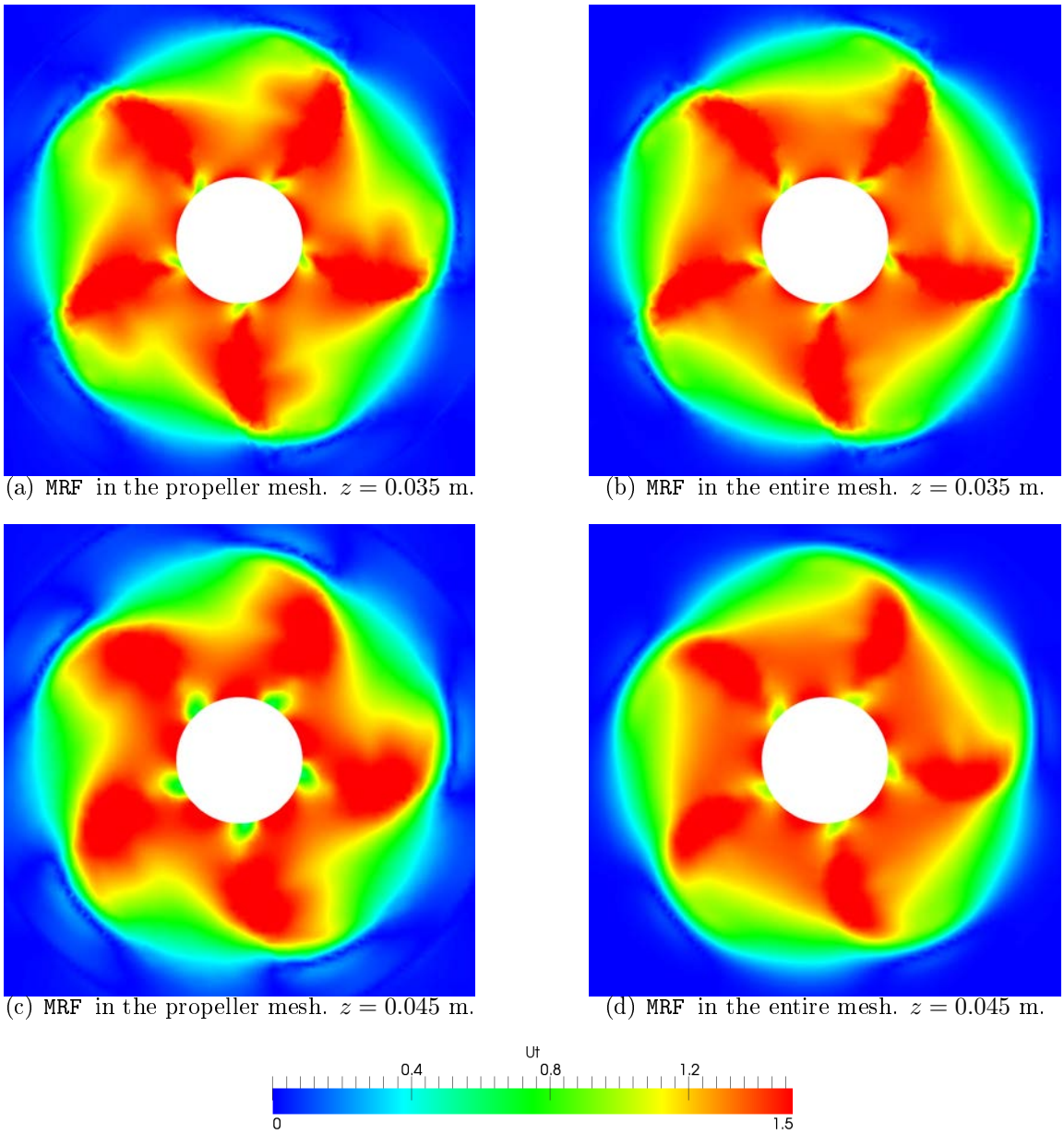


(a) `MRF` applied in the propeller mesh. (b) `MRF` applied in the entire mesh.

Figure 25: *The effect of the MRF method on the propagation of flow structures.*

The propagation of the flow reveals one interesting feature of the `MRF`. Looking at the velocities at, say, the propeller mesh boundary, the solution corresponds to a situation where the propeller is in a different position than what the modelled geometry implies. The wake at the boundary looks like it is coming from a propeller that has not rotated as much as the modelled propeller. Thus each cross-section, or in fact each point, corresponds to a different propeller position and that position depends on the time that it takes for the flow to come from the propeller to that specific point. When applying the `MRF` stops, the physics change abruptly so it is not surprising that something happens to flow structures such as tip vortex across the boundary.

As a conclusion, the propagation of the flow across the *mesh* boundary seems to work nicely. An effect of a spatial change in the `MRF` method (stopping it or changing it abruptly) is, however, not that simple and implies problems in steady-state CRP computations. Changes in the `MRF` source do affect the flow downstream of the propeller but also upstream of it. If possible, the upstream side of the `MRF` domain should be as far as possible from the propeller to avoid any effects on the flow around the propeller. In the case of a CRP, however, it is not possible to take the boundary far from both propellers.



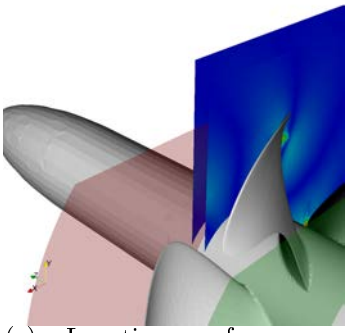
(a) `MRF` in the propeller mesh. $z = 0.035$ m.

(b) `MRF` in the entire mesh. $z = 0.035$ m.

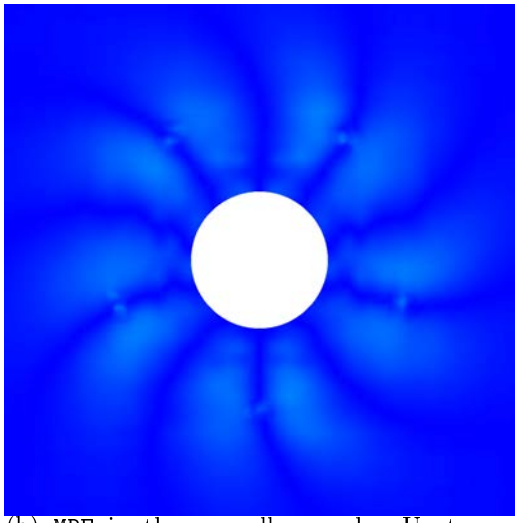
(c) `MRF` in the propeller mesh. $z = 0.045$ m.

(d) `MRF` in the entire mesh. $z = 0.045$ m.

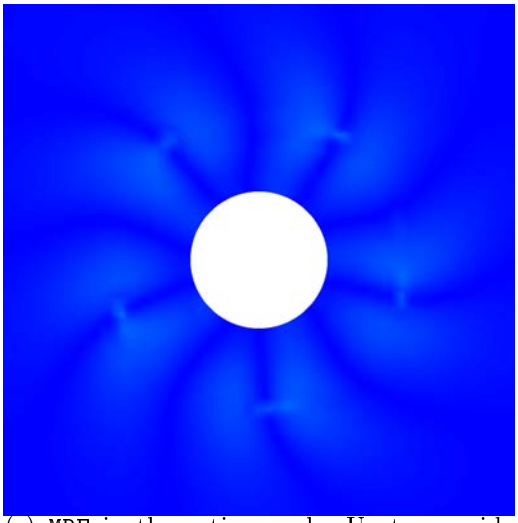
Figure 26: *Comparison of velocities around the downstream side mesh boundary.*



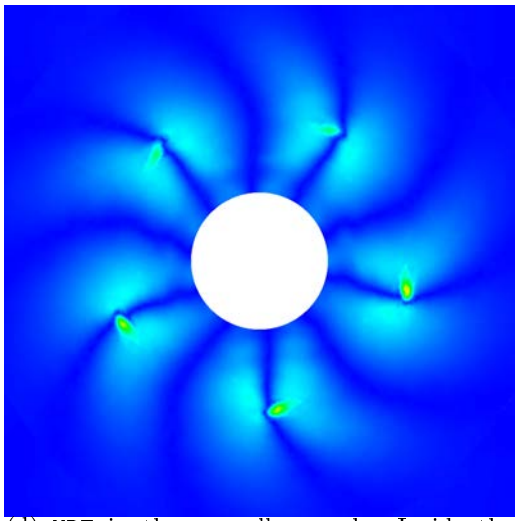
(a) Location of cross-sections. Upstream boundary of the propeller mesh shown in red.



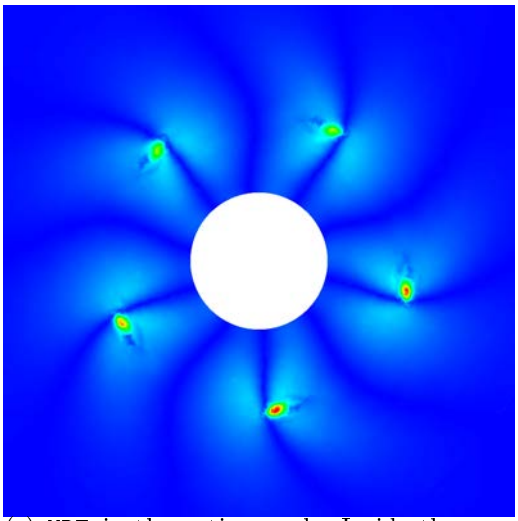
(b) `MRF` in the propeller mesh. Upstream side of the propeller mesh.



(c) `MRF` in the entire mesh. Upstream side of the propeller mesh.



(d) `MRF` in the propeller mesh. Inside the propeller mesh.



(e) `MRF` in the entire mesh. Inside the propeller mesh.

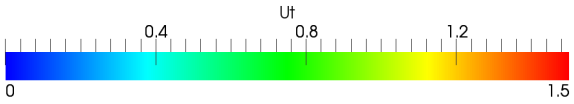


Figure 27: *Tangential velocities upstream of the propeller.*

6 Time-Accurate Computations with the Hybrid Mesh

In this chapter, the same hybrid mesh that was described in Chapter 5 is used with the time-accurate code `pimpleDyMFoam`. The boundary conditions are the same as earlier except for the propeller blades. They are defined as moving walls and the movement is defined to be the rotating movement.

The main focus in the time-accurate computations is to learn about errors arising from temporal discretizations and how they depend on chosen settings. Three main parameters that are expected to influence the temporal discretization are

- time discretization,
- time step size and
- the number of iterations inside one time step.

Increasing the number of iterations affects not only the accuracy but also increases the computational time. Therefore, it is important to find a number of iterations that gives an acceptable accuracy without spending too much time on unnecessary iterations.

The time step size clearly affects the computational time required for a simulation over a certain time interval by affecting the number of taken time steps. On the other hand, the longer the time step the more iterations are needed inside each time step, since the two adjacent solutions differ more from each other.

The type of the temporal discretization affects the accuracy of the computation but its effect on the simulation time is probably not very important. It does, however affect the memory requirements considerably. Both the Euler and the backward time discretizations are used and the number of iterations inside a time step is varied over a range from five to 40.

The open-water case is computed with three time steps corresponding to the following propeller rotations:

- $0.5°/\Delta t$,
- $2°/\Delta t$ and
- $5°/\Delta t$.

A clear motivation for the use of iterations inside a time step is given in Figure 28 that illustrates the iso-surface of a Courant number of $Co = 2$ on the left-hand-side and the isosurface $Co = 10$ on the right-hand-side. The maximum Courant numbers occur at the propeller blade tip where the mesh is dense and velocities are high but there are also regions with $Co > 2$ between the blades. These regions are caused by a denser region in the volume mesh that can be seen in the outer boundary of the propeller mesh also illustrated in the figures. The maximum Courant numbers are approximately $Co = 95$ in the case of the smallest time step. With 2 degrees of propeller rotation per time step it is $Co \approx 375$. The use of inner iterations allows the introduction of under-relaxation also in the transient computation, which

considerably stabilizes the solution procedure. In order to use this kind of a grid and, at the same time, take the Courant number down to approximately $Co = 1$, the time step would need to be in the order of 0.005° propeller rotation per time step, which would increase the number of time steps enormously.

(a) Courant number $Co = 2$. (b) Courant number $Co = 10$.

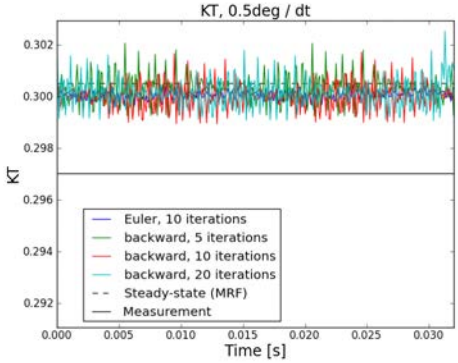
Figure 28: *Isovolumes of Courant number.*

The three parameters pose requirements to one another and they need to be considered as a whole in order to choose resonable settings. In the following, each parameter is varied separately and resulting changes in the force coefficients and in the wake field studied. Only one advance coefficient, $J = 0.90$, is used in the time-dependent case.

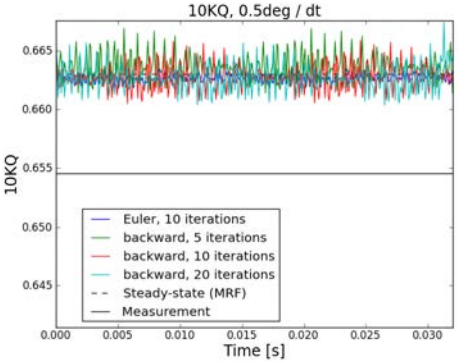
6.1 Force Prediction

The resulting force coefficients are plotted against time in Figure 29. In each subfigure, the measured value is given as a solid horizontal line and the steady-state result from Chapter 5 is represented by the dashed horizontal line. The steady-state results are those computed with `MRF` applied in the entire grid.

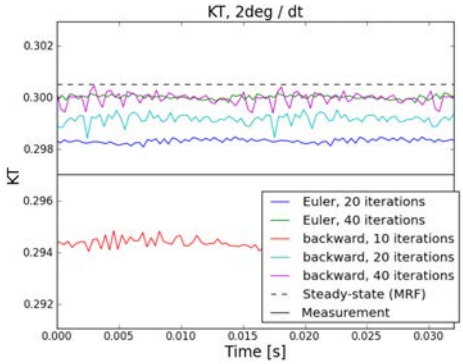
The first two subfigures illustrate the results with the smallest time step corresponding to 0.5° propeller rotation per time step. Both time discretizations yield the same average results. The backward-scheme gives more oscillations which is probably due to its lower dissipation compared to the Euler method. The number of iterations inside a time step does not affect the average results either which means that the error due to time discretization is not dominant compared to other factors. A further implication of a sufficiently accurate temporal discretization is the fact that the results are very close to the steady-state results computed with `MRF` applied in the entire grid. These results are referred to as *temporally converged* results in the following.



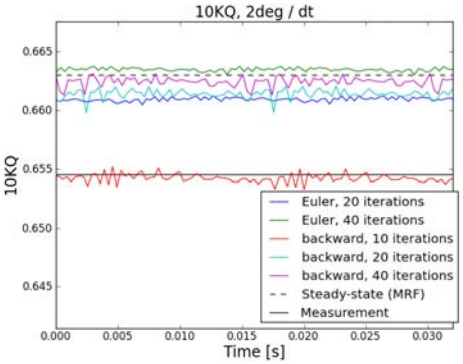
(a) Thrust coefficient K_T. Mesh movement per time step corresponds to 0.5°.



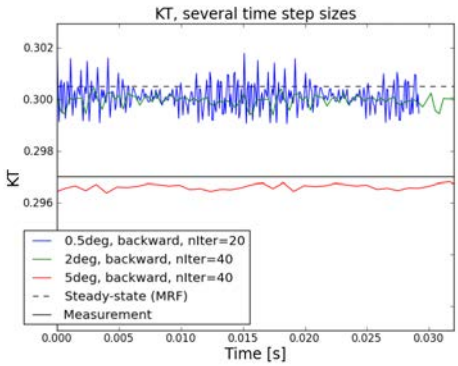
(b) Torque coefficient K_Q. Mesh movement per time step corresponds to 0.5°.



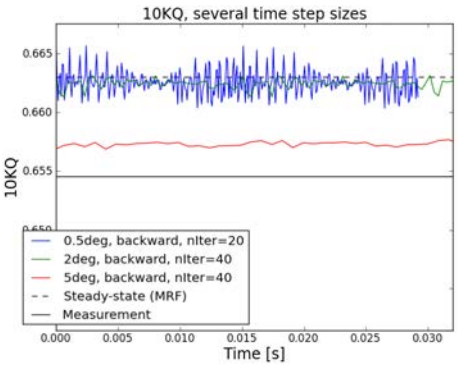
(c) Thrust coefficient K_T. Mesh movement per time step corresponds to 2°.



(d) Torque coefficient K_Q. Mesh movement per time step corresponds to 2°.



(e) Thrust coefficient K_T. Effect of the time step size.



(f) Torque coefficient K_Q. Effect of the time step size.

Figure 29: *Force coefficients (K_T and K_Q) against time at $J = 0.90$.*

With a larger time step, 2° propeller rotation per time step, shown in the next two subfigures, the time discretization plays a clear role. With an increasing number of iterations inside a time step, the results approach the level obtained with the smaller time step (rotation of $0.5°/\Delta t$). Using ten iterations yields results that differ from the temporally converged results by approximately two percent. With 20 iterations, the values are closer and with 40 iterations the results are the same as with the smaller time step. Thus it seems that the numerical error due to time discretization does not play a significant role even with 2 degrees of rotation per time step but the error due to insufficient iteration (iterative error) does affect the solution.

If other parameters are kept constant the backward discretization predicts the forces better than the Euler discretization. This could be expected since the error of the backward scheme is theoretically lower than that of the Euler scheme. According to the conclusion that only the iterative error affects the forces with the time step of 2° rotation per time step, not the numerical error, the differences in the numerical errors should not explain the difference, however. Based on these results, the backward scheme seems to reduce the iterative error, too, compared to the Euler method. Thus, the results encourage to use the backward scheme whenever there is enough memory available.

The lowest subfigures illustrate results with the three used time step sizes. As mentioned earlier, the two smallest time steps yield the same average values, if enough iterations are used inside a time step. A time step corresponding to a propeller rotation of 5° clearly underpredicts the forces with 40 inner iterations. A larger amount of iterations might lead to better results but that was not tested. This subject will be further analyzed later with the conclusion that even a considerably larger amount of iterations might not yield correct results.

An interesting trend is seen in the temporal variation of the forces. The larger the time step the less oscillations there are in the results. The time discretization (that was just found not to affect solutions with sufficiently low time steps) filters away oscillations from the computations. This observation is in line with the theory. The numerical error damps oscillations and with an increased time step the numerical error grows. Based on these observations, it can be concluded that the numerical error in the case of 2° rotation per time step is small enough not to affect the average value of predicted forces but does still affect the temporal oscillations of the forces. Even if the results are the same with different time step sizes provided that a sufficient amount of iterations are conducted within a time step, one thing needs to be kept in mind. The case itself is probably not time-dependent, in particular in the sense of global forces. The time step determines the time scale of reproducable flow phenomena, since the time scale of events one is interested in must, of course, be larger than the time step used in order for the computation to be able to reproduce them. If there are structures on a smaller time scale in other cases such as the CRP-case the results should also have a stronger dependence on the time step size.

Table 4: *Comparison of force coefficients, K_T and K_Q, computed with moving mesh ($0.5°/\Delta t$), backward discretization and ten inner iterations) and steady-state (`MRF` applied everywhere).*

	K_T	$10\,K_Q$
`steady-state`	0.3005	0.6630
`time-accurate`	0.3000	0.6625
`Relative difference (%)`	0.17	0.08

Compared to the steady-state results obtained with `MRF` applied everywhere in the mesh, the time-accurate results are very similar. This fact was already seen qualitatively in Figure 29 but, for a quantitative comparison, the average values computed with the smallest time step (0.5° rotation per time step), backward discretization and ten inner iterations and those given by the steady-state computation are given in Table 4. The results differ only by a fraction of a percent which is well below the accuracy of the computations themselves. Thus the results are in line with earlier computations and verify the correct usage and performance of the time-accurate code.

6.2 Convergence Within a Time Step

In order to better understand the connection between the time step size and convergence inside a time step, convergence histories of pressure inside one time step are plotted in Figure 30. The computations that yielded results unaffected by the time discretization (temporally converged results) are plotted in green, and the computations with errors in red. With a short time step, the residual level in the beginning of a time step is lower than in the case of a longer time step, and that level is independent on the number of iterations conducted inside the time step. A longer time step seems not only to decrease the convergence rate but also the achievable level of accuracy inside a time step. These results show that the computation where propeller is rotated by 5° in a time step would probably not give good results even with a very large number of inner iterations.

It is difficult to see from the residual histories when the iterative error no more impairs the solution. The residuals in cases of 0.5° rotation per time step with 5 iterations are higher than the residuals with 2° rotation and 20 iterations. However, the results in the latter case have an error while the other results are good. There does not seem to be any clear correlation between the residuals and the force prediction. It is, however, interesting to see that the convergence first reaches a maximal rate which then decreases with an increasing number of iterations. This implies that it might be beneficial to have such a short time step that the convergence level required to remove the effect of iterative error is reached during the fastest convergence, whatever the required level may be. A further optimisation of the solution procedure is left for further studies. For example, the use of a PISO-loop at the end

of a time step should decrease the iterative error but it was omitted in this study due to stability reasons.

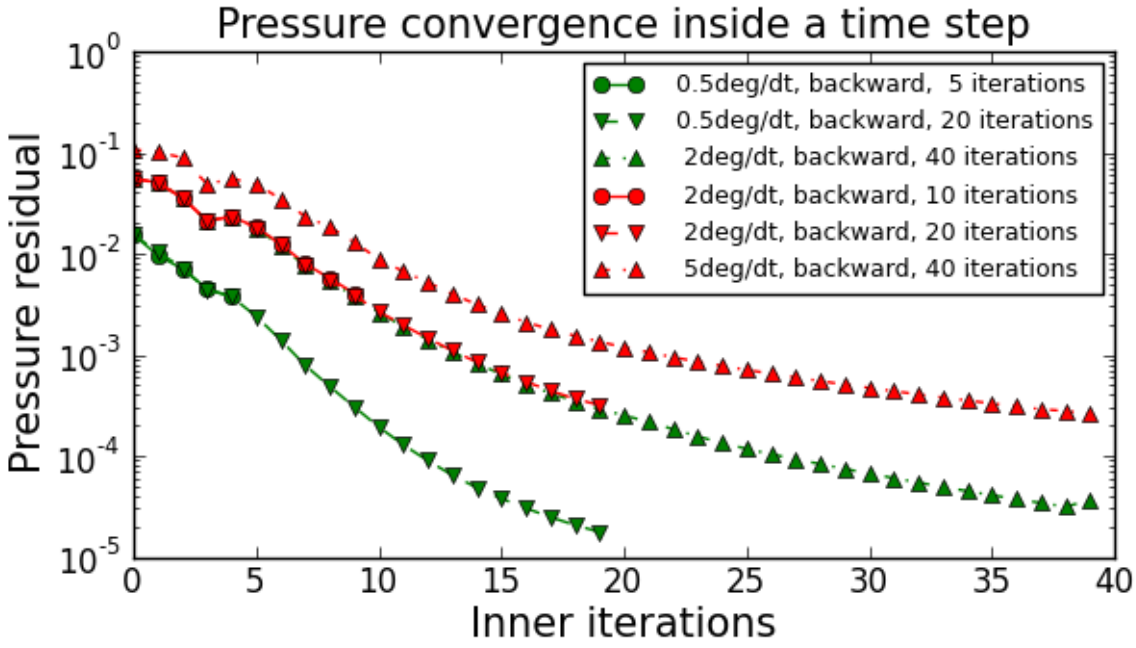


Figure 30: *Residual histories of pressure.*

6.3 Wake Prediction

The velocity fields at 0.055 m behind the propeller (see the last cross-section in Figure 22) is shown in Figure 31. The first two contours are from the cases with 0.5° rotation per time step that both reproduced the forces well. The figure on the left-hand-side should, with the smallest time step, backward time discretization and 20 iterations per time step, theoretically be the most accurate out of all of the six cases under consideration. There seem to be less velocity peaks in the wake, however, than in the less accurate case with only five iterations per time step (top right subfigure).

The same trend can be seen in other subfigures. The less iterations conducted within a time step the sharper velocity peaks there are in the wake (see 'middle left' and 'middle right' subfigures). Similarly, the longer the time step the stronger the peaks are (see 'middle left' and 'bottom right' subfigures). The numerical error related to the time discretizations is diffusive in its nature so theoretically, a longer time step should yield a smoother velocity field. How the iterative error affects is not that clear.

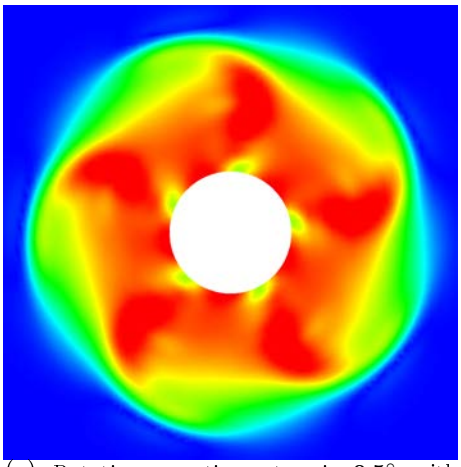
Comparing the cases with the Euler and backward schemes (see 'middle right' and 'bottom left' subfigures) shows that the Euler scheme gives a smoother wake than the backward scheme. By theory, this is expected since the error related to the Euler scheme should be more diffusive.

The streamwise propagation of the propeller wake is illustrated in Figure 32. The cases are the same as in Figure 31 and the same observations made based on Figure 31 are confirmed by looking at any streamwise position behind the propeller. There is a possible explanation to the results that seemingly contradict with theory. A less diffusive (more accurate) computation is able to reproduce smaller flow structures that effectively smooth the flow field. A numerical dissipation preventing such flow structures can thus reduce mixing processes in the flow and thus give sharper gradients in the propeller wake. The fact that there are less oscillations in the force coefficients with longer time steps is in line with this explanation.

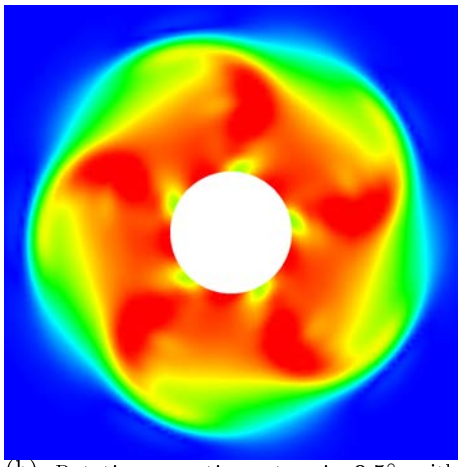
Anyway, even if the propeller forces may be correctly predicted with less accurate handling of temporal changes, the wake may not be correctly reproduced with the same settings. In a case of two propellers, the differences in the wake fields certainly affect at least the distribution of forces at the aft propeller. Thus, they can also affect the global force prediction. The sensitivity of forces on the wake prediction needs to be treated more carefully in the CRP computations. At this point it is only clear that the forces on the front propeller do not directly indicate anything of the quality of the propeller wake.

In all cases in Figure 32, the mesh boundary can be seen in the flow field, in particular in the outer part of the wake. No clear differences can be pointed out between different numerical settings, however. Somehow the boundary isolates structures on its opposite sides. Based on these figures, the mesh boundary does not seem to impair the solution too much but anyway attention must be paid on the subject in the CRP computations.

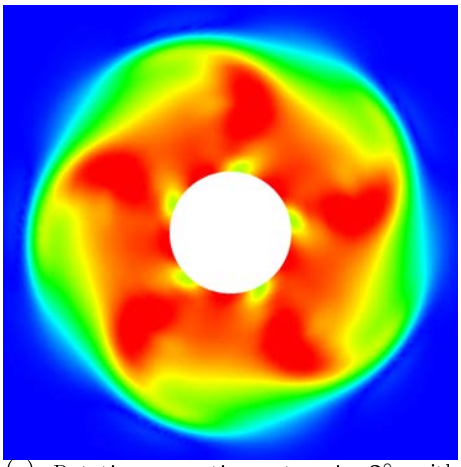
The time-accurate code seems to work as expected. The force prediction is in line with the steady-state results and, even if the case itself is more or less steady, the effect of important parameters affecting the error of the temporal discretizations can be seen. As a conclusion, the usage of the second-order backward scheme is encouraged. Using a time step of 0.5° rotation per time step, should yield good results quite efficiently with even less than ten iterations per time step, provided that it is enough to resolve the global flow correctly. In a CRP-case, a rotation of 2° per time step is probably not enough to predict all transient effects between the passing blades but can be used in initial computations where it is not necessary to obtain accurate results.



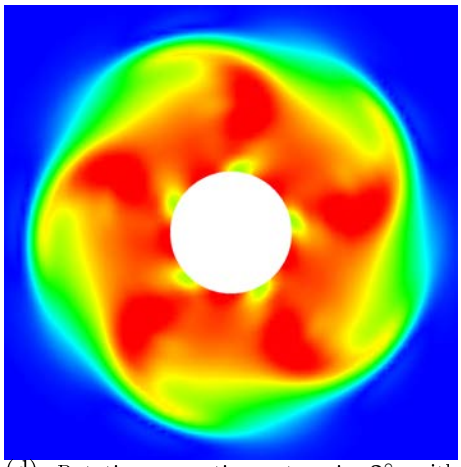
(a) Rotation per time step is **0.5°** with the `backward` time discretization and `20 iterations`.



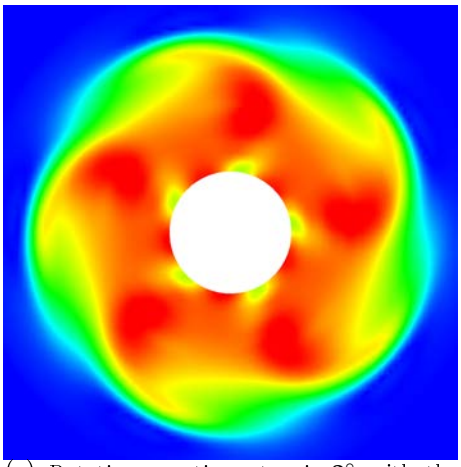
(b) Rotation per time step is **0.5°** with the `backward` time discretization and `5 iterations`.



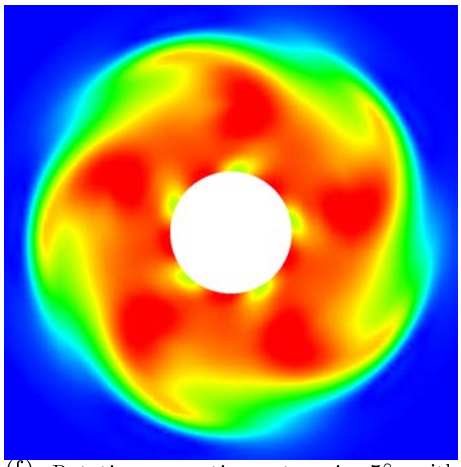
(c) Rotation per time step is **2°** with the `backward` time discretization and `40 iterations`.



(d) Rotation per time step is **2°** with the `backward` time discretization and `20 iterations`.



(e) Rotation per time step is **2°** with the `Euler` time discretization and `20 iterations`.



(f) Rotation per time step is **5°** with the `backward` time discretization and `40 iterations`.

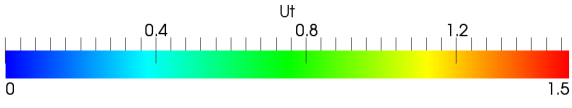
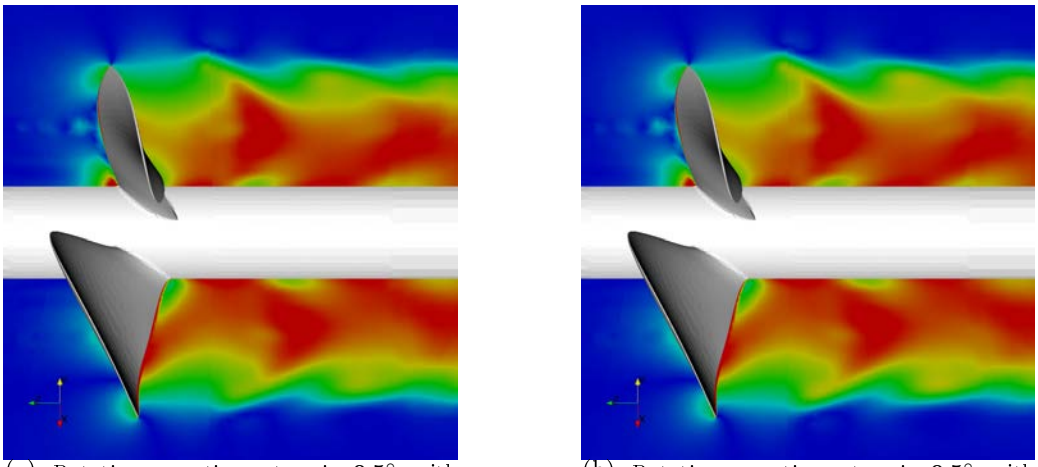
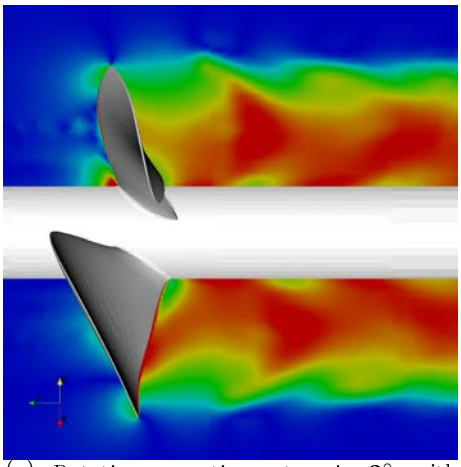


Figure 31: *Tangential velocities at a distance of* 0.055 *m behind the propeller (see the last cross-section in Figure 22).*

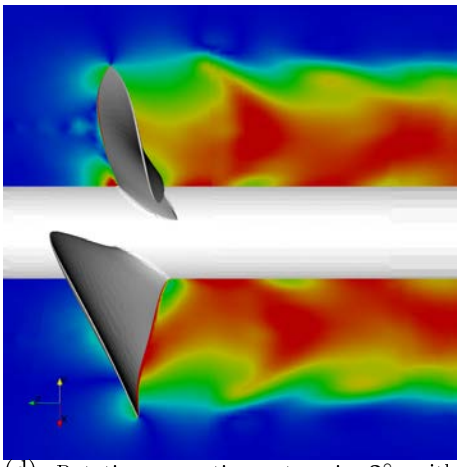


(a) Rotation per time step is **0.5**° with the `backward` time discretization and `20 iterations`.

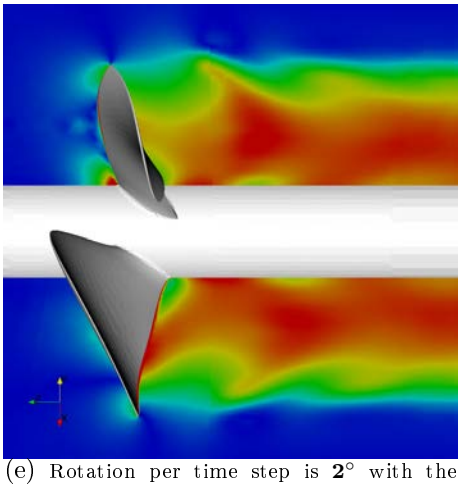
(b) Rotation per time step is **0.5**° with the `backward` time discretization and `5 iterations`.



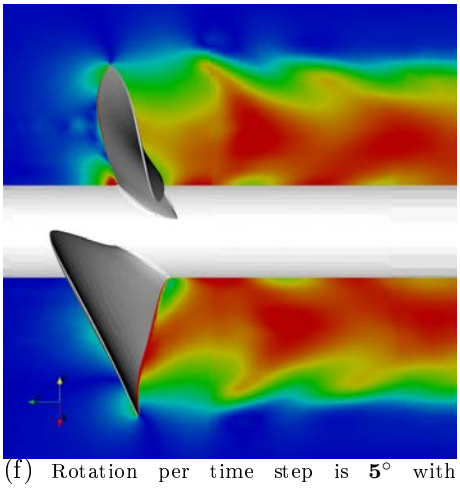
(c) Rotation per time step is **2**° with the `backward` time discretization and `40 iterations`.



(d) Rotation per time step is **2**° with the `backward` time discretization and `20 iterations`.



(e) Rotation per time step is **2**° with the `Euler` time discretization and `20 iterations`.



(f) Rotation per time step is **5**° with the `backward` time discretization and `40 iterations`.

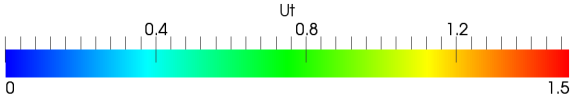


Figure 32: *Propagation of tangential velocities behind the propeller.*

7 CRP Computation

All computational methods, that were chosen for the computations in this work, have been applied on the open-water case. The computational set-up has been chosen, fixed and tested, and knowledge has been gained on how to use the relevant methods. In this chapter, the methods are applied on the more complicated case that has a thruster unit with two pushing propellers. Measurements have been conducted with the same set-up in a towing tank as discussed in Chapter 1 [30].

The modelled geometry is shown in Figure 33. The yellow surfaces (outer walls, hub and shaft) are modelled as symmetry planes which effectively means that nothing can flow through them and no friction is accounted for. Propeller blades are handled in the same way as in the open-water case and the thruster surface is modelled as a wall. The thruster is divided in two parts. The lower part is submerged in water in the experimental set-up and the upper part is above the water line. The computational domain spans approximately six propeller diameters upstream and five diameters downstream of the front propeller. There are ten diameters free space to both sides and downwards from the propeller axis and the space between the two propellers is a quarter of the propeller diameter.

There are two propeller domains and a surrounding region in the computational mesh that have all been meshed separately and afterwards merged together. The surfaces with discontinuities are illustrated in the lowest subfigure of Figure 33. The boundaries of the front propeller mesh are colored in turquoise and the boundaries of the aft propeller mesh in blue consistently with the propeller blades in other subfigures. Between the two propeller meshes, there is yet another meshing interface (grey in the lowest subfigure) where both sides move in the time-dependent case and where the `MRF` source changes direction in the steady-state computation.

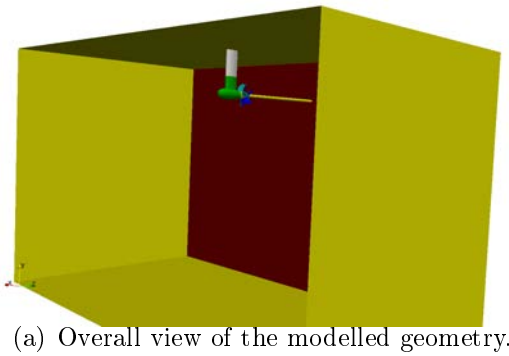
The upstream side of the front propeller mesh is perpendicular to the incoming flow direction (z-axis) and both outer boundaries are cylindrical surfaces. The boundary between the two propeller meshes is divided in three parts. There is one part perpendicular to the z-axis beginning from the hub and one limiting to the cylindrical outer boundaries. Between them, there is an axisymmetric surface that connects the other two parts. The reason for dividing the surface is to bring the mesh interface sufficiently far from both propellers. Since the propeller tips are bent towards the downstream direction, a straight surface would unavoidably be close to either one or both of the propellers. The downstream side boundary of the aft propeller mesh is a single axisymmetric surface that begins from the end of the hub and is tilted towards the downstream direction in order to avoid having the boundary too close to the aft propeller near the propeller tip.

The mesh has a total of 11.8 Mio. cells which is the largest number of cells that can be computed with the 16GB of RAM available in the workstation that was used for most of the computations. The propellers are meshed in exactly the same way as the propeller in the open-water case. The blade surfaces are divided and the surface mesh density is defined identically. Furthermore, the volume mesh in the surface normal direction is also similar to the mesh in the open-water case. Referring to the results in the previous chapters, such a mesh around the propellers is not expected

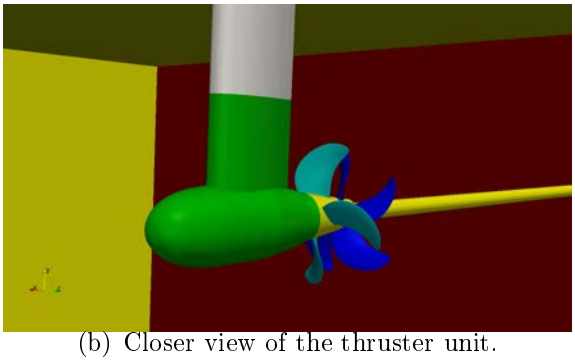
to cause any considerable errors into the solution. The main parameters of the propellers are approximately as given in Table 5.

Table 5: *Main dimensions of the propellers.*

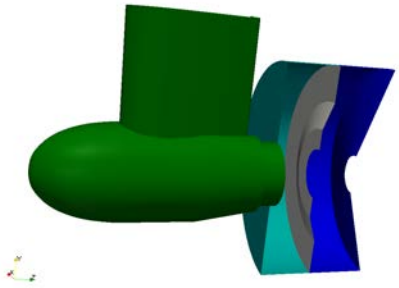
	Front propeller	Aft propeller
D [mm]	213	203
D_{hub} [mm]	62	48
Number of blades	4	5



(a) Overall view of the modelled geometry.



(b) Closer view of the thruster unit.



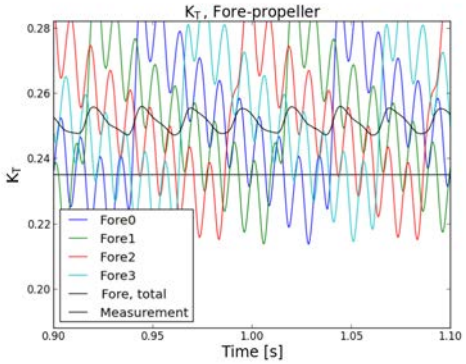
(c) Propeller mesh boundaries (mesh discontinuities).

Figure 33: *Geometry in the CRP case.*

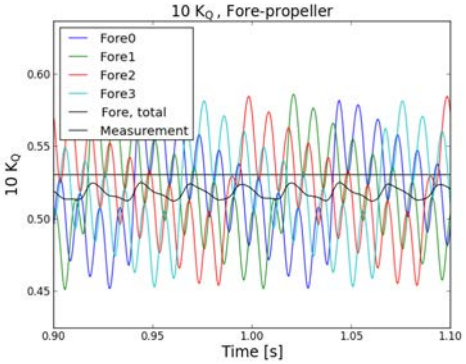
7.1 Results

As opposed to the open-water case, the CRP case is truly time-dependent which is also seen in the global forces. No results were obtained with the `MRF` and it was concluded that steady-state computations with multiple propellers are, if not impossible, clearly involved with difficulties.

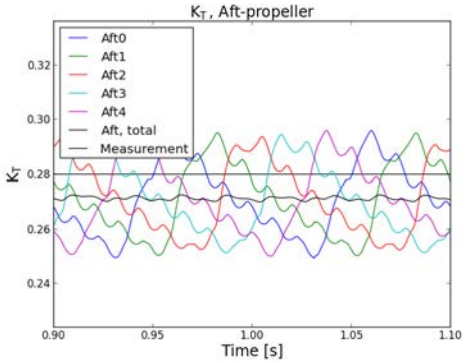
With the time-accurate code, thrust and torque coefficients computed with the Euler time discretization, 2° of propeller rotation per time step and 35 inner iterations are shown in Figure 34. The contribution of each blade is shown separately with different colors and the total value of the propeller is illustrated in black. The total contribution varies in time at the frequency of a single blade of the front propeller passing the thruster unit. In the contributions of a single blade (colored lines in Figure 34), an additional frequency of the front propeller blade passing an aft propeller blade is observed.



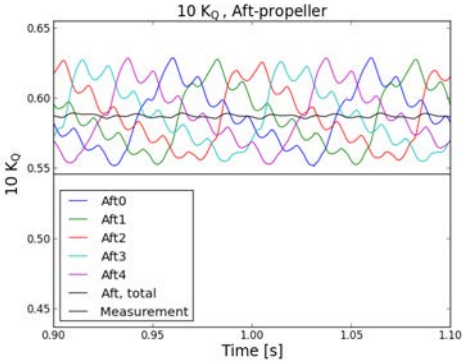
(a) Thrust coefficient, K_T on the front propeller.



(b) Torque coefficient, K_Q on the front propeller.



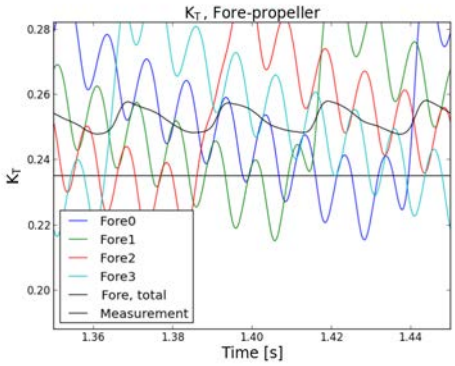
(c) Thrust coefficient, K_T on the aft propeller.



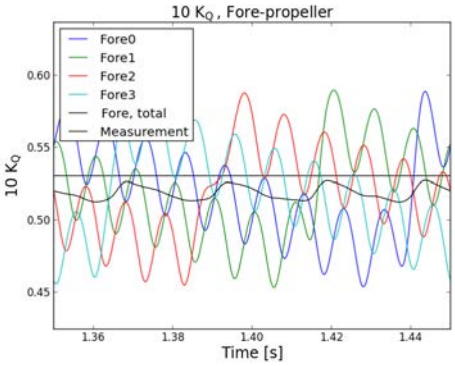
(d) Torque coefficient, K_Q on the aft propeller.

Figure 34: *Thrust and torque coefficients of the front and aft propellers of the CRP unit. Rotation per time step is* 2° *with the Euler time discretization and 35 iterations.*

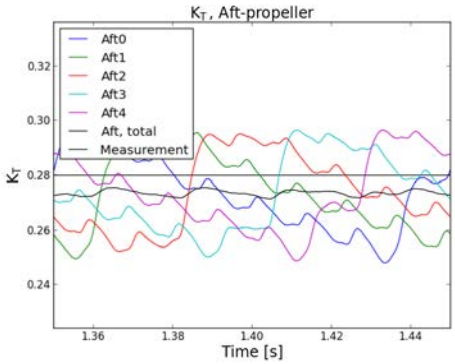
The force coefficients computed with a 0.5° propeller rotation per time step are given in Figure 35. Additional variation of the coefficients can be observed at the aft propeller. The shorter time step seems to reproduce more details about the interaction between the propellers than the longer time step which is in line with the discussion in the open-water case. The change in average values was, however, in the order of only half a percent.



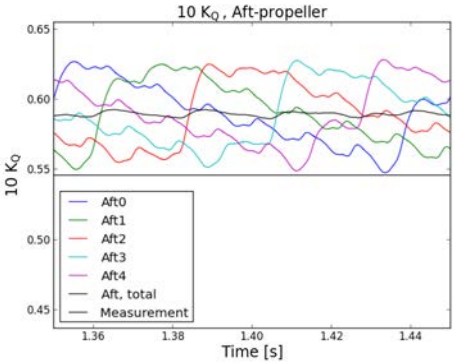
(a) Thrust coefficient, K_T on the front propeller.



(b) Torque coefficient, K_Q on the front propeller.



(c) Thrust coefficient, K_T on the aft propeller.



(d) Torque coefficient, K_Q on the aft propeller.

Figure 35: *Thrust and torque coefficients of the front and aft propellers of the CRP unit. Rotation per time step is* 0.5° *with the Euler time discretization and 17 iterations.*

The average levels of the force coefficients are given in Table 6. The values are given in relative differences to the measured values in percent. The thrust on the front propeller is overpredicted in both cases as is the torque on the aft propeller. The front propeller torque and the aft propeller thrust are underpredicted by two - three percent. The errors are considerably larger than those observed in the open-water case.

The results show that the time discretization with the parameters used clearly affects the solution. It does, however, not explain the whole deviation from the measured values. Despite finding the backward scheme beneficial compared to the

Euler scheme, all the CRP results presented here are computed with the Euler scheme. The reason for this is the larger memory requirements of the backward scheme.

Table 6: *Comparison of the aft and the front propeller force coefficients to the measured values. Results are given in relative differences to measured values in percent.*

	K_T, fore	K_Q, fore	K_T, aft	K_Q, aft
$2°/\Delta t$, 16 iterations	5.2	-3.6	-2.5	8.2
$2°/\Delta t$, 35 iterations	6.9	-2.3	-3.1	7.6
$0.5°/\Delta t$, 17 iterations	7.4	-2.2	-2.3	8.0

The y^+ values illustrated in Figure 36 reveal one source of uncertainty. The front propeller mesh yields to y^+ values of greater than four in large areas of the blades. In the open-water case, the y^+ values were smaller than 2 in most parts and there was still a dependence on the grid in that case. There are also areas with high y^+ values on the thruster which may affect the boundary layer development on the thruster and thus the thruster wake, too.

Figure 37 shows the velocity magnitude on a plane located behind the thruster. The velocity field clearly shows that the mesh between the thruster and the front propeller is too coarse. The non-conformal boundary between the surrounding part and the propeller meshes is clearly visible which is reflected in the results. Since the inflow to the front propeller is incorrect, it is difficult to evaluate the correctness of the force prediction on either of the propellers.

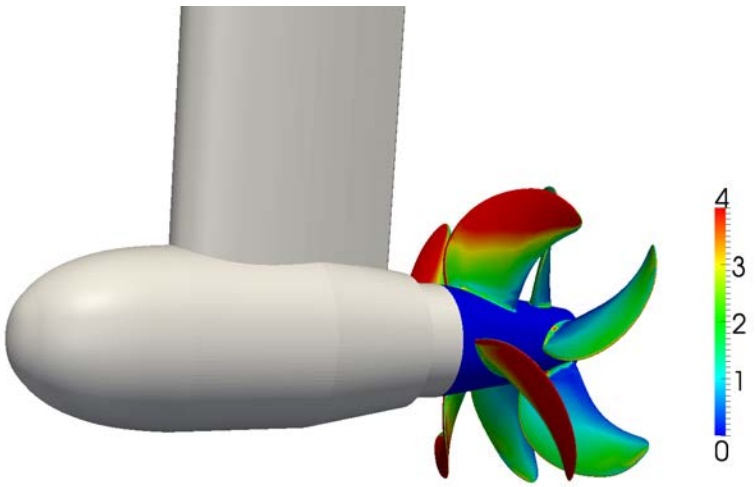
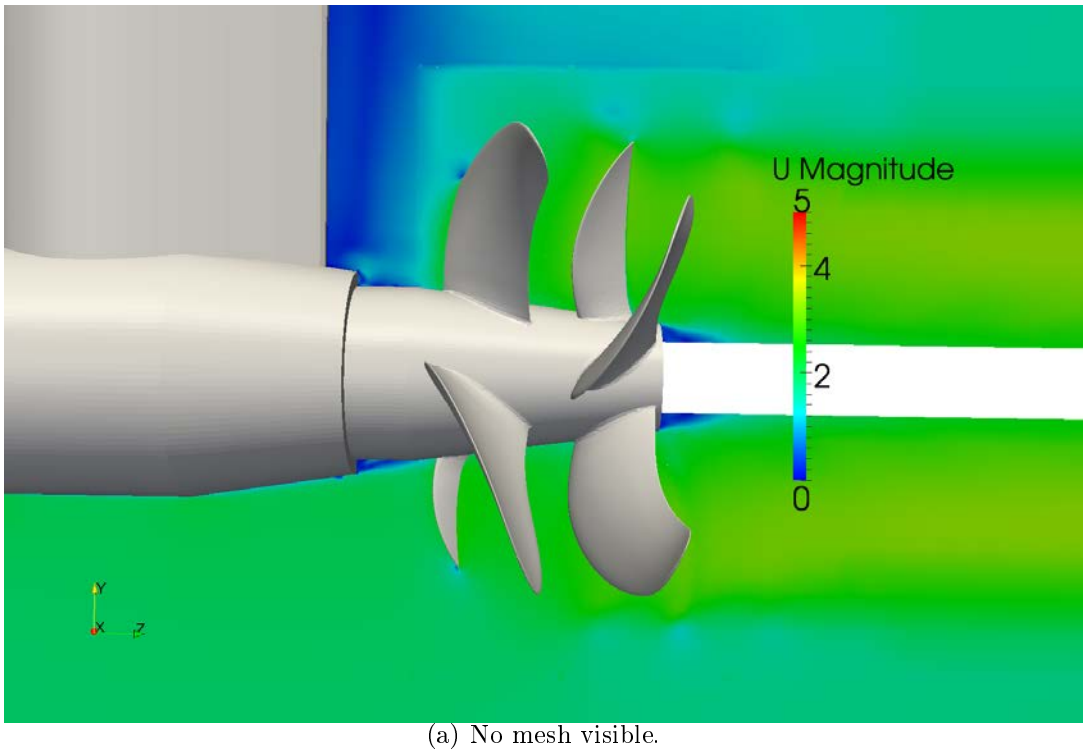
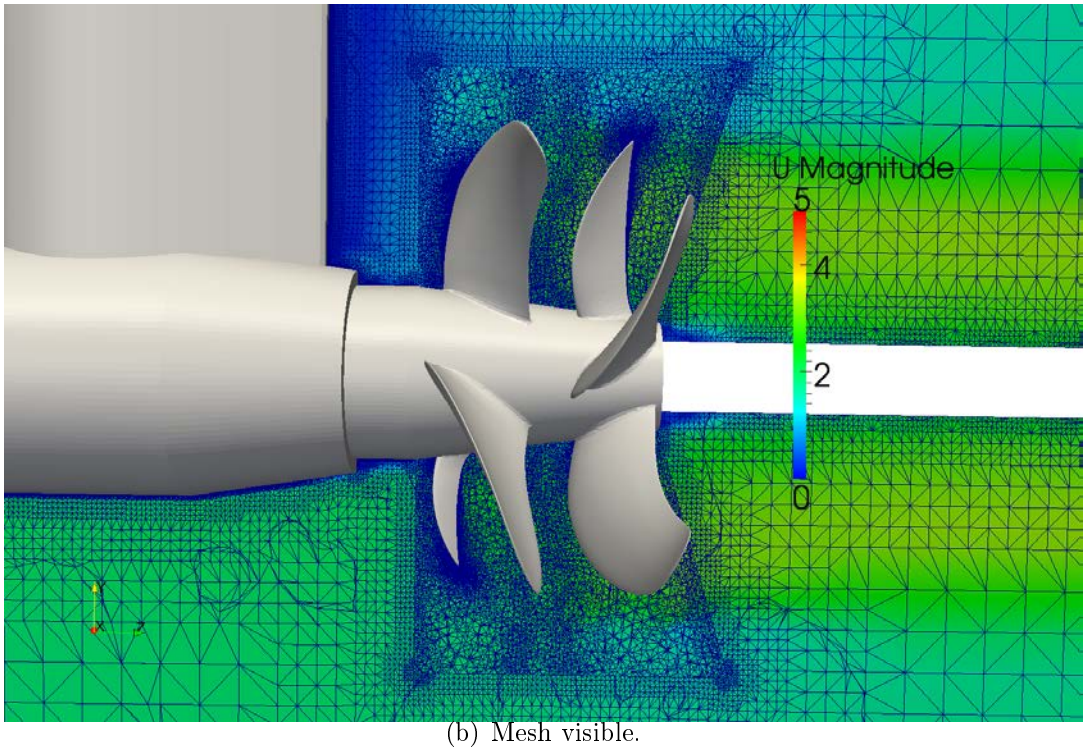


Figure 36: y^+ *values.*



(a) No mesh visible.



(b) Mesh visible.

Figure 37: *Velocity field behind the thruster unit. The mesh boundary is clearly seen due to too coarse a mesh.*

8 Conclusions and Discussion

Computational methods available in `OpenFOAM` were tested by applying them on the analysis of marine propellers. The methods included the Moving Reference Frame (`MRF`) used to reduce the analysis of a rotating propeller into a steady-state problem. Furthermore, time-accurate computations were conducted using moving meshes and an iterative solution process inside each time step.

All methods were found to work well when applied on the analysis of a single propeller in open-water conditions. The final computational set-up is stable and predicts propeller forces within approximately two percent of the measured values at moderate values of advance coefficient. The methods have not been tested in bollard pull conditions.

When applied on a case with two propellers and a thruster unit in front of them, no steady-state results could be obtained. Even with the time-accurate code, force prediction was no longer as good as in the case of a single propeller in open-water conditions. There are several possible reasons for this. As discussed in Chapter 7, the mesh was found to be too coarse both on the front propeller and in the space between the front propeller and the thruster. With the current set-up and mesh, forces were predicted within eight percent of the measured values. According to the literature review in Chapter 1, an accuracy of approximately five percent could be expected in CRP computations in open-water conditions and additional uncertainties are caused by the propellers operating in the thruster wake. In this respect, the results are acceptable but, since it was shown that the prediction of the thruster wake clearly poses uncertainties due to a coarse grid in this study, the evaluation of the reliablility of the code requires further studies.

The simulation must be continued with a refined mesh in order to reach reliable estimates on the best possible accuracy of the computational method. It can be stated here that such computations were made later and the effect of the grid discontinuities on the results was solved. Further results will not be given in this work. In order to make further studies, a sufficient amount of computer resources should be available. Most of the computations presented here were conducted with a workstation with six cores and 16GB of RAM. For further studies, the amount of RAM needs to be increased in order to allow the use of denser meshes and more accurate time discretizations. Doubling the RAM to 32GB would enable much more accurate studies. Furthermore, the number of cores should be increased. Adopting a workstation with 16 cores would probably decrease the computation time even if finer meshes were used. That would enable tests with a reasonable response time which is essential for efficient working. The most demanding computations in this study were conducted with 128 cores at CSC - IT Center for Science, that can provide sufficient computer capacity. A more extensive use of the capacity offered by them would provide a good basis for efficient further studies.

An interesting path for additional studies is the evaluation of ways to reduce the cost of the computations. In this study, no wall functions were applied to reduce the number of cells near solid structures, since the main focus was on the evaluation of the other methods. Since the methods have now been found to work reliably, the

introduction of wall functions is an obvious topic for further studies. Even without the wall functions, it could be expected that also multi-propeller applications can be analysed in a feasible way with `OpenFOAM`. With the help of a few workstations, comparisons can be conducted in the time span of some weeks. Keeping in mind the possibility of clusters and the advances in the computer technology, the duration of the computations does not seem to be a problem. The lack of licence costs in `OpenFOAM` further contributes to its feasibility. All in all, it seems probable that, after the proposed further studies, `OpenFOAM` will provide a reliable tool for the analysis of multi-propeller marine applications in R&D purposes.

References

[1] Lewis, E., *Principles of Naval Architecture Volume II Resistance, Propulsion and Vibration.* New Jersey, NJ: The Society of Naval Architects and Marine Engineers, 1988.

[2] Katz, J. and Plotkin, A., *Low-Speed Aerodynamics.* Cambridge: Cambridge University Press, 2nd ed., 2001. ISBN 0–521–66219–2.

[3] *OpenFOAM.* http://www.openfoam.org.

[4] Carlton, J., *Marine Propellers and Propulsion.* Elsevier Ltd.

[5] Sanchez-Caja, A., Sipilä, T., and Pylkkänen, J., "Simulation of Viscous Flow around a Ducted Propeller with Rudder Using Different RANS-Based Approaches," in *First International Symposium on Marine Propulsion*, 2009.

[6] Ansys Inc., Canonsburg, PA 15317, USA, *ANSYS FLUENT 13.0 User's Guide*, 2010.

[7] Guo, C., Ma, N., and Yang, C., "Numerical Simulation of a Podded Propulsor in Viscous Flow," *Journal of Hydrodynamics*, Vol. 21, No. 1, 2009, pp. 71–76.

[8] Sanchez-Caja, A., Rautaheimo, P., Salminen, E., and Siikonen, T., "Computation of the Incompressible Viscous Flow around a Tractor Thruster Using a Sliding-Mesh Technique," in *Seventh International Conference on Numerical Ship Hydrodynamics*, July 1999.

[9] Wang, Z. and Xiong, Y., "Effect of Time Step Size and Turbulence Model on the Open Water Hydrodynamic Performance Prediction of Contra-Rotating Propellers," *China Ocean Eng.*, Vol. 27, No. 2, 2013, pp. 193–204.

[10] Feng, X., Lu, C., and Wu, Q., "Numerical Researches on interaction between Propellers in Uniform Flow," *Journal of Hydrodynamics*, Vol. 24, No. 5, 2012, pp. 675–683.

[11] Dhinesh, G., Murali, K., and Subramanian, V., "Estimation of hull-propeller interaction of a self-propelling model hull using RANSE solver," *Ships and Offshore Structures*, Vol. 5, No. 2, 2010, pp. 125–139.

[12] Xia, L., Lundberg, J., and Bensow, R., "Performance Prediction of a Nozzle Propeller," in *29th Symposium on Naval Hydrodynamics*, (Gothenburg).

[13] SVA, http://www.svapotsdam.de/towingtank.html, *SVA website*, Accessed 2/17/2014.

[14] Farrell, P. E. and Maddison, J. R., "Conservative interpolation between volume meshes by local Galerkin projection," *Comput. Methods Appl. Mech Engrg*, Vol. 200, 2011, pp. 89–100.

[15] White, F. M., *Viscous fluid flow*. Singapore: McGraw-Hill Book Co., 3nd ed., 2006. ISBN 0–07–124493–X.

[16] Holmes, D. and Tong, S., "A Three-Dimensional Euler Solver for Turbomachinery Blade Rows," *J. of Engineering and Gas Turbines and Power*, Vol. 107, April 1985.

[17] Siikonen, T. and Pan, H., "Application of Roe's Method for the Simulation of Viscous Flow in Turbomachinery," in *Proceedings of the First European Computational Fluid Dynamics Conference*, (Brussels), pp. 635–641, Elsevier Science Publishers B.V., Sept. 1992.

[18] Mansfield, M. and O'Sullivan, C., *Understanding Physics*. Chichester: John Wiley & Sons Ltd, 2006. ISBN10 0–470–06529–X.

[19] *See the MRF development*. http://openfoamwiki.net/index.php/See_the_MRF_development, 2013.

[20] Ferziger, J. and Perić, M., *Computational methods for fluid dynamics*. Berlin: Springer-Verlag, 3^{rd} ed., 2002. ISBN 3–540–42074–6.

[21] Wilcox, D., *Turbulence modeling for CFD*. La Canada: DCW Industries, Inc., 1993. ISBN 0–9636051–0–0.

[22] Hellsten, A., "Some improvements in Menter's $k-\omega$ SST turbulence model," in *29th AIAA Fluid Dynamics Conference*, (Albuquerque, New Mexico, USA), AIAA paper 98–2554, 1998.

[23] OpenFOAM Foundation, *OpenFOAM Programmer's Guide, version 2.2.0*.

[24] Hirsch, C., *Computational Methods for Inviscid and Viscous Flows*, Vol. 2 of *Numerical Computation of Internal and External Flows*. Chichester: John Wiley & Sons Ltd, 1990. ISBN 0–471–92351–6.

[25] Patankar, S., *Numerical Heat Transfer and Fluid Flow*. Washington, D.C: Hemisphere, 1980.

[26] Issa, R. I., "Solution of the Implicitly Discretised Fluid Flow Equations by Operator-Splitting," *Journal of Computational Physics*, Vol. 62, No. 11, 1986, pp. 40–65.

[27] Balkmann, U., Heinke, H.-J., and Lübke, L., "Potsdam Propeller Test Case (PPTC), Test Case Description," in *Second International Symposium on Marine Propulsors*.

[28] "Uncertainty Analysis in CFD, Verification and Validation, Methodology and Procedures, $7.5-03-01-01$," International Towing Tank Conference (ITTC), 2005.

[29] Eça, L. and Hoekstra, M., "A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies," *Journal of Computational Physics*, Vol. 262, 2014, pp. 104–130.

[30] "Comparative Open Water and Cavitation Tests with Three Thrusters with CR Propellers," SVAtech GmbH, Potsdam, 2012.