Free-Surface Simulations with Consideration of Moving Objects, Sea Waves and Scour Based on the Reynolds-Averaged Navier-Stokes Equations

Dissertation

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

(Dr.-Ing.)

der Technischen Fakultät

der Christian-Albrechts-Universität zu Kiel

vorgelegt von

Janek Hartmut Meyer (M.Eng.)

Kiel, 2021

Prüfungskommission:

1. Gutachter:	Prof. Dr. Thomas Slawig
2. Gutachter:	Prof. Dr. Michael Hinze
3. Prüfer:	Prof. Dr. Thomas Meurer
Vorsitz:	Prof. Dr. Reinhard Koch

Disputationstermin: 05. November 2021

Zusammenfassung

Diese Arbeit beschreibt ein numerisches Verfahren zur Simulation beweglicher Segelyachten an der freien Wasseroberfläche basierend auf den *Reynolds-gemittelten Navier-Stokes Gleichungen* und der *Volume-of-Fluid* Gleichung, welche mit der *Finite-Volumen* Methode, einem impliziten Zeitschrittverfahren sowie einem *SIMPLE* ähnlichem Algorithmus gelöst werden. Dabei werden Simulationen mit Glattwasser sowie Seegang berücksichtigt. Des Weiteren wird das Verfahren durch eine Methode zur Simulation von Auskolkungsvorgängen an Offshore-Fundamenten erweitert.

Das vorgestellte Simulationsverfahren wird in das quell-offene Softwarepaket *OpenFOAM* implementiert. Dabei werden nicht nur die in OpenFOAM vorhandenen Methoden, sondern auch weitere, ergänzend implementierte, bereits bestehende oder eigene, neu entwickelten Methoden genutzt. Im Vergleich zu dem OpenFOAM eigenen Simulationsverfahren für freie Wasseroberflächen *interFoam*, löst das in dieser Arbeit neu erarbeitete Verfahren zahlreiche Probleme auf Seiten der numerischen Stabilität, bei zugleich optimierter Qualität der Ergebnisse.

Beschrieben werden die gewählten Erhaltungsgleichungen sowie die Finite-Volumen Methode zur Diskretisierung der räumlichen Terme. Ergänzend wird eine bekannte Diskretisierungs-Methode zur Berücksichtigung der durch die scharfe Wasseroberfläche entstehenden Diskontinuitäten implementiert.

Die Methoden zur Diskretisierung der instationären Terme werden mit einer neuen, stabilisierten Methode zweiter Ordnung ergänzt. Für die Diskretisierung des konvektiven Terms der Volume-of-Fluid Gleichung werden bekannte High-Resolution Schemata implementiert und untersucht. Zudem wird ein eigener Ansatz zur Unterdrückung der verbleibenden numerischen Ventilation präsentiert. Zur Turbulenzmodellierung wird das Zwei-Gleichungsmodell *k-ω-SST* aus OpenFOAM mit bekannten Anpassungen modifiziert. Die vorhandene Methode zur Simulation der Starr-Körper-Bewegung wird mit einem bekannten Verfahren zur Stabilisierung des expliziten Zeitschrittverfahrens erweitert. Zudem wird ein Ansatz zur Abschätzung relevanter Segelkräfte implementiert. Für die Erzeugung des Seeganges wird die *waves2Foam* Bibliothek eingearbeitet. Anschließend wird auf die benötigte Wellendämpfung eingegangen und sowohl eine weitere bekannte, wie auch eine neue, eigene Dämpfungsmethode implementiert.

Simulationsergebnisse für die *Rayleigh Taylor Instabilität* werden erzeugt und anhand analytischer Gleichungen bewertet. Des Weiteren werden Ergebnisse für den Testfall *Sysser 60* aus der *Delft Systematic Yacht Hull Series* präsentiert und qualitativ mit aus *Star-CCM+* sowie interFoam resultierenden Ergebnissen verglichen. Zudem werden die wichtigsten Fälle einer Segelyacht-Studie der *Sailing Yacht Research Foundation* simuliert und die Ergebnisse mit denen aus Tankversuchen und Simulationen quantitativ verglichen. In allen Fällen bieten die neuen Resultate erhebliche Verbesserungen zu interFoam.

Das Verfahren wird durch eine Methode zur Simulation von Auskolkungsvorgängen erweitert. Zur Simulation des Bodens und der Suspension wird ein modifiziertes *Bingham-Modell* implementiert. Dabei wird der benötigte relative Druck durch ein eigenes Verfahren abgeschätzt. Die in OpenFOAM vorhandenen Wandfunktionen des Turbulenzmodells werden mit einem selbst entwickelten Ansatz auf die Domain-interne Sedimentwand übertragen. Zur Berücksichtigung von Boden-Abschrutschungen wird ein neu entwickeltes Sliding Modell implementiert. Es wird gezeigt, dass bei der Verwendung eines einzigen Geschwindigkeitsfeldes für alle Phasen Probleme, durch die Vermischung der Boden- und Wassergeschwindigkeiten, in Grundnähe auftreten. Diese Ungenauigkeit wird durch einen eigenen Algorithmus bei der Diskretisierung unterdrückt. Des Weiteren wird gezeigt, dass die Umwandlung des Bodens in die Suspension noch unzureichend dargestellt wird. Abschließend werden die Simulationsergebnisse für Auskolkungsvorgänge hinter einer Kante sowie um einen vertikalen Zylinder mit und ohne Mudplate präsentiert und mit experimentellen Ergebnissen verglichen, wobei eine gute Übereinstimmung festgestellt werden kann. Durch die Nutzung der Volume-of-Fluid Methode ist das hier vorgestellte Verfahren im Vergleich zu den meisten anderen auch auf beliebig geformte Geometrien anwendbar. Zugleich bleibt der Berechnungsaufwand, durch die Nutzung des relativ einfachen Bingham-Modells sowie der übertragenen Wandfunktionen, gering.

Abstract

This works describes a numerical method for the simulation of moving sailing yachts at the free water surface based on the *Reynolds-averaged Navier-Stokes Equations* and the *Volume-of-Fluid* equation, which are solved using the *Finite Volume* method, an implicit marching method and a *SIMPLE* like algorithm. Thereby, simulations with smooth water and swell are considered. Moreover, the method is complemented by the possibility to simulate scour at offshore groundings.

The presented simulation method is implemented into the open-source software package *OpenFOAM*. Not only the methods available in OpenFOAM, but also other, supplementary implemented, already existing or own, newly developed methods are used. Compared to OpenFOAM's own simulation method for free water surfaces *interFoam*, the method newly developed in this thesis solves numerous problems on the part of numerical stability, while at the same time optimizing the quality of the results.

The selected conservation equations and the Finite Volume method for discretizing the spatial terms are described. In addition, a well-known discretization method is implemented to take into account the discontinuities caused by the sharp water surface.

The methods for discretizing the unsteady terms are supplemented with a new, stabilized second-order method. Well-known high-resolution schemes are implemented and investigated for the discretization of the convective term of the Volume-of-Fluid equation. In addition, a newly developed approach to suppress the remaining numerical ventilation is presented. For the turbulence modeling, the two-equation model *k-ω-SST* from OpenFOAM is modified with well-known adjustments. The existing method for simulating rigid body motion is supplemented by a known method for stabilizing the explicit time discretization. In addition, an approach for estimating relevant sailing forces is implemented. The *waves2Foam* library is incorporated to generate the swell. Then the required wave damping is discussed and a further known, as well as a new, own damping method implemented.

Simulation results for the *Rayleigh Taylor Instability* are generated and evaluated using analytical equations. Furthermore, results for the test case *Sysser 60* from the *Delft Systematic Yacht Hull Series* are presented and compared qualitatively with results from *Star-CCM+* and interFoam. In addition, the most important cases of a sailing yacht study by the *Sailing Yacht Research Foundation* are simulated and the results are quantitatively compared with those from tank tests and simulations. In all cases, the new results offer significant improvements to interFoam.

The simulation model is extended by a method for simulating scouring processes. A modified *Bingham* model is implemented to simulate the soil and the suspension. Thereby the required relative pressure is estimated using a newly developed procedure. The wall functions of the turbulence model in OpenFOAM are transferred to the domain-internal sediment wall using a self-developed approach. A newly developed sliding model is implemented to take account of soil sliding. It is shown that when using a single velocity field for all phases, problems due to the mixing of the soil and water velocities, occur close to the ground. This inaccuracy is suppressed by a newly developed algorithm during the discretization. Furthermore it is shown that the transformation of the soil into the suspension is still insufficiently represented. Finally, the simulation results for scouring processes behind an apron and around a vertical cylinder with and without a mudplate are presented and compared with experimental results, whereby a good agreement can be found. By using the Volume-of-Fluid method, the model presented here can, compared to most of the others, also be applied to geometries of any shape. At the same time, the computational effort remains low, due to the use of the relatively simple Bingham model as well as the transferred wall functions.

Contents

Nomenclature

Latin letters

$\mathbf{A}$	Coefficient matrix
a	Matrix coefficient
a_d	Matrix coefficient of the main diagonal
a_n	Matrix coefficient of a secondary diagonal
C	Central cell in the Normalized Variable Approach
C_0	Scheme parameter of the angle correction
C_{limit}	Limiting Courant-Friedrichs-Lewy Number
C_{o}	Courant-Friedrichs-Lewy Number
C_{oLower}	Lower Limit of the Courant-Friedrichs-Lewy Number blending
C_{oUpper}	Upper Limit of the Courant-Friedrichs-Lewy Number blending
C_{R}	Reflection coefficient
D	Downwind cell
$\mathbf{d}$	Distance vector from the owner to the neighbor cell center
$\mathbf{F}$	Force vector
f	face
f_{x}	Interpolation factor
$\mathbf{g}$	Gravity vector
g	Magnitude of gravity vector
H	Wave height
$\mathbf{I}$	Inertia tensor
k	Turbulent kinetic energy
L_{os}	Lenght over surface
$\mathbf{M}$	Moment vector
m	Mass of the object
N	Neighbor cell
N_{v}	Numerical ventilation
$\mathbf{n}$	Surface normal vector
P	Owner cell
p	Pressure
p_{d}	Dynamic pressure
P_k	Turbulent production rate
$\mathbf{q}$	Arbitrary source term
q	SIMPLE iteration

r	relaxation factor
R_e	Reynolds number
S	Stability domain
$\mathbf{s}$	Face area vector
$\mathbf{s}_\text{non-ortho}$	Non-orthogonal part of the face area vector
$\mathbf{s}_\text{ortho}$	Orthogonal part of the face area vector
s	Face area
T	Wave period
$\mathbf{T}_{zyx}$	Transformation tensor to transform from the body to the global coordinate system
t	Time
U	Far upwind cell in the Normalized Variable Approach
$\mathbf{u}$	Velocity vector
$\mathbf{u}_b$	Local grid velocity vector
u_c	Velocity magnitude of the current
V	Volume
$\mathbf{v}$	Translational velocity vector of the ship
X	X-axis of the global coordinate system
x	x-axis of the body coordinate system
Y	Y-axis of the global coordinate system
y	y-axis of the body coordinate system
Z	Z-axis of the global coordinate system
z	z-axis of the body coordinate system

Superscripts

CoG	With respect to the the center of gravity
T	Transposed

Subscripts

BCS	With respect to the body coodinate system
CoE	With respect to the center of effort
C	Central cell in the Normalized Variable Approach
D	Downwind cell
d	main diagonal
f	face
GCS	With respect to the global coodinate system
i	Ith cell or row of the matrix
N	Neighbor cell or neighbor element in the matrix
n	Secondary diagonal
P	Owner cell

U	Far upwind cell in the Normalized Variable Approach

Greek symbols

α	Volume fraction
α_w	Volume fraction of the water phase
β_{BICS}	Scheme parameter of the Blended Interface Capturing Scheme
β_{GDS}	Scheme parameter of the Gamma Differencing Scheme
β_{IGDS}	Scheme parameter of the Inter Gamma Differencing Scheme
Γ_ϕ	Diffusivity
Δt	Time step size
ϵ	Turbulent dissipation rate
θ	Angle between face normal and free-surface normal
θ_{p}	Pitch Angle
λ	Wave length
μ_{e}	Effective dynamic viscosity
μ_ℓ	Laminar dynamic viscosity
μ_{t}	Dynamic eddy viscosity
ν_{e}	Effective kinematic viscosity
ν_ℓ	Laminar kinematic viscosity
ν_{t}	Kinematic eddy viscosity
ρ	Density
ρ_w	Density of the water
ρ_a	Density of the air
$\hat{\rho}$	Reverse interpolated face value of the density
τ	Viscous stress tensor
Φ	Face flux
ϕ	Generic quantity
$\tilde{\phi}$	Normalized generic quantity in the Normalized Variable Approach
ϕ_{r}	Roll Angle
ψ	Yaw Angle
Ω	Angular velocity vector of the object
ω	Specific turbulent dissipation rate

Operators

$\mathscr{D}$	Divergence operator
$\mathscr{H}$	Discretization operator using a higher order scheme
$\mathscr{I}$	Interpolation operator
$\mathscr{L}$	Discretization operator using a lower order scheme
$\mathscr{R}$	Reconstruct volume field operator
∇	Nabla Operator

Abbreviations

BC	Boundary condition
BCG	Body coordinate system
BDF	Backward Differentiation Formula
BICS	Blended Interface Capturing Scheme
BRICS	Blended Interface Capturing Scheme with Reconstruction
CBC	Convective Boundedness Criterion
CDS	Central Differencing Scheme
CFL	Courant-Friedrichs-Lewy Number
CFLC	Courant-Friedrichs-Lewy Condition
CICSAM	Compressive Interface Capturing Scheme for Arbitrary Meshes
DDS	Downwind Differencing Scheme
DNS	Direct Numerical Simulation
GCS	Global coordinate system
GDS	Gamma Differencing Scheme
HRIC	High-Resolution Interface Capturing
IGDS	Inter-Gamma Differencing Scheme
LES	Large Eddy Simulation
MULES	Multi-Dimensional Limiter for Explicit Solution
NV	Numerical Ventilation
NVA	Normalized Variable Approach
NVD	Normalized Variable Diagram
OF	OpenFOAM
PISO	Pressure Implicit with Splitting of Operator
SHM	SnappyHexMesh
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
UDS	Upwind Differencing Scheme
VOF	Volume-of-Fluid
VPP	Velocity Prediction Program
WL	Water Line

List of Figures

List of Tables

1. Introduction

1.1. Motivation

In the recent decades the computational fluid dynamics (CFD) have become a widely used method in the naval architecture and ocean engineering. In the development phase of a vessel or sailing yacht it is often applied to estimate important properties. Besides basic parameters like the resistance, lift and floating position, also more complex predictions are more and more requested, for example the roll damping coefficient or the sea keeping behavior in irregular waves.

Beyond doubt, computational fluid dynamics have established itself at least as an addition to towing tank trials and have become an inherent part of naval architecture. With increasing accuracy, but also computational costs the applied numerical methods reach from potential flow solvers, over RANSE[1]-solvers to LES[2]-methods. Although the first two methods have become a cheaper alternative to towing tank tests, commercial software packages for numerical flow simulations are still expert tools with license costs easily reaching 100,000€ per year.

An alternative to commercial software is the free, open-source software package OpenFOAM[3] contributed under the GNU GPLv3 license. OpenFOAM already comes with a huge amount of functionalities. It allows to simulate one-phase or multi-phase flow including a lot of different modeling approaches for the turbulence. Furthermore it holds different grid motion techniques and an extensive - *but difficult to handle* - grid generator, called *snappyHexMesh*. The source code is written in C++ and publicly available. Almost all of its numerical methods are based on the Finite Volume method. For the solution of the equation systems, different iterative approaches are available, including multi grid methods. For many applications OpenFOAM delivers tools, which are able to keep up with commercial codes. Also a solver for *free-surface*[4] simulations is available, called *interFoam*. But, at least this solver is not competitive, as shown in this work, as it is missing the necessary stability for industrial applications. This has been confirmed by Claughton 2015a, who compared the results of different flow solvers, Star-CCM+, FINE/Marine, OpenFOAM, FlowLogic and Shipflow, to towing tank tests in the field of use of hydrodynamic sailing yacht investigations.
For the user, the instabilities seems to lead to an arbitrary behavior and diverging simulations. Although OpenFOAM comes at no costs, this problem makes the free-surface solver uneconomic. The here presented work closes this gap by developing and implementing a new free-surface solver into OpenFOAM. The main field of use are hydrodynamic simulations of sailing yachts in calm water or in swell. Compared to commercial codes the new solver can be passed internally to other users or developers without any restrictions. Furthermore, the complete code of the solver is accessible avoiding the

[1] Reynolds-averaged Navier-Stokes Equations
[2] Large Eddy Simulations
[3] Open Source Field Operation and Manipulation
[4] In this work *free-surface* stands for a sharp interface between two phases, for example water and air.

typical *black-box* character of commercial tools.

The last-mentioned advantages are also the reason why the newly developed free-surface solver can be extended for the simulation of scouring processes within this work. Such a procedure does not currently exist in the standard OpenFOAM packages. Furthermore, many methods are based on the deformation of the domain boundaries to represent the sediment erosion. This prevents the application on geometries of any shape, e.g. on a monopile foundation with a mounted mudplate. Since the procedure developed in this thesis is based on the Volume-of-Fluid method, such simulations are also possible.

1.2. Focus and Contribution of this Work

The first focus of this work is the development of a state-of-the-art flow solver for the simulation of sailing yachts in calm water and sea waves. All developments are implemented into the open source simulation package OpenFOAM. For the first task OpenFOAM already comes with a solver called *interFOAM*. Indeed, its reliability and quality of the results are not competitive compared to commercial solvers like Star-CCM+ or FINE/Marine (Claughton 2015a). Hence, the given work should close this gap based on the given numerical methods delivered with OpenFOAM and additionally implemented known and newly developed numerical methods.

The second focus is the extension of this flow solver with a method to simulate scouring processes around arbitrarily shaped offshore-foundations. Such method is not available in the standard OpenFOAM packages.

The main contributions of this work onto free-surface simulations of sailing yachts with OpenFOAM are

- the implementation of three known high-resolution discretization schemes for the convective term of the Volume-of-Fluid equation,
- implementation of a known discretization method, treating the pressure and density discontinuity at the free-surface in a numerical stable way,
- development and implementation of a new method to suppress wrong numerical ventilation [5] below the yacht,
- application of a known method to circumvent the inherently given non-physical turbulent decay between the flow-inlet and the yacht,
- implementation of a known method to stabilize the explicit time marching method of the simulation of the motion of the yacht,
- implementation of a typicall approach to estimate the relevant forces induced by the sails of the yacht

[5] Here, *numerical ventilation* means the phenomenon, where wrong air is convected under the hull due to the used numerical methods. Hence, the submerged grid-cells directly below the hull may contain more air than water. This is a serious problem, as the non-physical air reduces the drag significantly.

- incorporation of the *waves2Foam* library to generate sea waves,
- development and implementation of a stabilized second order backward differentiation formula scheme required for the simulation of sea waves,
- implementation of a known wave damping method to damp sea waves,
- development and implementation of a new wave damping method with less needed wave specific parameter adjustment,
- and the implementation of known modifications for the stabilization of the used k-ω-SST turbulence model to avoid non-physical growing of the turbulence intensity in sea wave simulations and further problems.

The main contributions of this work onto scour simulations with OpenFOAM are

- all above listed contributions for free-surface simulations (except the parts for the yacht motion)
- the implementation of a known Bingham model to simulate the soil and the suspension,
- development and implementation of a new approach to estimate the required relative pressure,
- development and implementation of a new model to simulate soil sliding,
- development and implementation of a way to transfer OpenFOAMs wall functions onto the domain internal sediment wall,
- and the development and implementation of a method to prevent wrong sediment creeping while using only one shared velocity field for all relevant phases (soil, suspension and water).

1.3. Development Process

The main intention has been to create an OpenFOAM based, state of the art in-house solver for basic free-surface flow simulations of sailing yachts coupled with rigid body motion. This solver should be competitive to the until then used, leading commercial codes, but without the recurring license costs. Later the development work has become more project driven and the possibility to simulate sea waves, sediment and scouring was added during a two year research project supported by the *Ministerium für Energiewende, Landwirtschaft, Umwelt und ländliche Räume des Landes Schleswig-Holstein*.

The methodical basic idea has been to develop a RANSE solver based on the Finite Volume method. First, starting from OpenFOAMs *interFoam* solver the new development has become a completely new solver only sharing OpenFOAM‘s basic numerical methods.

The unique MULES algorithm, responsible for the calculation of the free-surface in interFoam was replaced by solving the standard Volume-of-Fluid transport equation in combination with a high-resolution discretization method. This is also the common way in other commercial flow codes like Siemens Star-CCM+, NUMECA FINE/Marine or Ansys Fluent.

The newly implemented high-resolution schemes guarantee a sharp resolution of the free-surface. With the fluid-velocity initialized with the desired boat speed, this results in unwanted, but correct flow details like splash waves at the bow in the beginning of the simulation. Therefore, it was decided to accelerate the flow from a small velocity to the target value in the very beginning. Now, avoiding such temporary flow phenomena, the final quasi steady state solution is reached much earlier.

Originally, the jump behavior of the density at the free-surface was unattended and its smearing results in the potential of infinite growing velocities in the lighter phase above the free-surface. Based on Patrick Queutey and Michel Visonneau 2007 a suitable discretization method considering this jump behavior was implemented improving the stability and reliability of the solver significantly.

Based on this improvement it has been possibly to do state of the art hydrodynamic simulations of sailing yachts, without the motion of the yacht. Simulations with free rigid body motion especially of wide and light sailing yachts were failing mostly. After fruitless testing of different explicit time discretization methods, the solution was found by using an *added mass* approach (Söding 2001 and Shigunov, Söding, and Zhou 2001). Such a method introduces an additional mass, which reduces the error of the explicit time stepping method and finally stabilizes the method.

As it is common practice doing the hydrodynamic simulations of sailing yachts without the sail, but with free pitch and heave, it is necessary to estimate the sail forces and moments to achieve a correct equilibrium of the floating position of the yacht. Therefore, a suitable but simple “sail-model” was implemented, which derives the forces for the free axis of motion from the flow. With these methods the code was ready for sailing yacht simulations with rigid body motion in flat water.

The basic capability to simulate sea waves, was incorporated by implementing the *waves2Foam* library (Jacobsen, Fuhrman, and Fredsøe 2012). The library provides the possibility to generate sea waves belonging to various wave theories based on potential flow theory. For example boundary conditions based on first-, second- and fifth-order Stokes wave theory or cnoidal wave theory are given for the velocity and water height. Boundary conditions for the pressure are not available. This is also the reason, why it is necessary to use a relaxation zone at the wave inlet of the domain. In this zone the given potential solution is smoothly blended over to the pure solution of the RANSE solver, by using an explicit relaxation. As a first modification, this explicit relaxation was replaced by an implicit method guaranteeing an undisturbed convergence. Still, some major parts for high quality wave simulations were left open at this point.

A major issue is the partial reflection of the simulated wave at the outlet. To suppress such reflections a wave damping method, applied infront of the outlet, was implemented. First, the sponge layer method, presented by Choi and Yoon 2009 was incorporated. This approach is also used in Star-CCM+. As the method requires a parameter adjustment depending on the given wave properties (R. Perić and Abdel-Maksoud 2016), a new damping method was developed (see Meyer, Graf, and Slawig 2017 attached in B) achieving the same quality of damping, but without the necessity of a wave depended parameter adjustment.

The correct simulation of the wave propagation requires at least a second order discretization scheme for the time derivatives. Standard, 2nd order accurate schemes are already given in the OpenFOAM

library. Indeed, these schemes have a tendency to produce oscillations, especially due to the high density jump at the free-surface. Decreasing the time step size may help but is impracticable. At best the oscillations are barely visible at the free-surface and will not influence the relevant results of the simulation, but at its worst, the oscillations may build up and destroy the free-surface completely. Therefore, a stabilized, 2nd order scheme was developed, which is presented in this work without a final mathematical proof of it's stability properties.

A further problem was induced by the turbulence model. The typically used models, like the k-ω-SST or k-ϵ model have the characteristic to produce infinite turbulence in the wave regions of the simulation. A few wave length behind the inlet, the waves have lost half of their height due to the way too big turbulence. First, the turbulent production term was modified based on the suggestion given by Mayer and Madsen 2000. As the modeling approach is not based on physical arguments, it was replaced later by the method presented by Larsen and Fuhrman 2018. The second approach introduces an additional limitation for the turbulent viscosity based on a mathematical stability analysis of common turbulence models. The analysis also shows, that all investigated unmodified models produces wrong turbulence in the potential flow area of sea waves.

As previously mentioned the content of the given work was influenced by given projects. Therefore, the last part leaves the field of naval architecture and goes over to ocean engineering. With the previously described progress for the free-surface solver it was obvious that this work can be the fundament for the simulation of scouring[6] processes at offshore structures by the addition of a third sediment phase.

The sediment is simulated with the help of a Bingham model, which basically models a solid sediment behavior by introducing a very high viscosity. The relative pressure used by the Bingham model is estimated with a new approach based on the solution of a Poisson equation. To keep the typical wall characteristics without demanding a fine grid, the common wall functions have been transferred to the domain internal sediment walls. Furthermore, additional modifications have been applied to model a solid sediment wall inside the solution domain. Only the combination of all these methods made it possible to simulate the sediment correctly.

1.4. OpenFOAM Versions and Naming

The code development started using OpenFOAM 1.6-extend. The major work was done using the OpenFOAM distributions 2.3.x and 2.4.x. While the final scour solver implementation has been frozen at version 2.4.x, all other parts underlie continuous development in which a downward compatibility is not intended. At this time, the solver is based on OpenFOAM 1906. The complete code is kept in-house and part of a nearly complete automation procedure for yacht simulations, called *Naval Simulation Tech (NST)*, also developed and implemented by the author. The final flow solver has been named *NSTFoam*.

[6] Scour is the erosion of sediment on the seabed near offshore structures

1.5. Applications Outside this Work

The presented solver was used for several scientific and industrial projects including clients in the professional sailing sport. The projects have ranged from small dinghies, over innovative planning motor yachts and foiling catamarans to sailing yachts with measures over 100ft.
In Graf, Renzsch, and Meyer 2016 the solver was applied to estimate the hydrodynamic forces of foiling catamarans and the results were used by a newly developed *Velocity Prediction Program* [7] (VPP).
In Graf, Meyer, et al. 2017 the solver was used successfully for general resistance calculations of a yacht hull, sailing yacht simulations in head sea, motor yacht simulations at high Froude numbers and the simulation of a yacht surfing in a following sea.
In Graf and Freiheit 2020 the solver was used one more time to deliver the hydrodynamic values for a VPP. This time, to optimize the trimming parameters of the Olympic Nacra-17 foiling catamaran.
Although not part of this work, a lot of effort was spent into the meshing procedure using snappyHexMesh. This finally leads to an almost complete automation of complete test matrices for standard sailing yacht simulations, including meshing, running and post-processing, while accomplishing competitive simulation time, accuracy and reliability. Since three years this setup is applied for several commercial projects in the field of sailing and motor yachts.

1.6. Outline

This work can be divided into two parts. First the method for the simulation of a moving sailing yacht in waves, among other things, is described in chapters 2 to 11. Based on this, the simulation method has been extended to simulate scouring around arbitrary offshore foundations. This second part is given in the publication attached in appendix C. The dissertation has the following structure:

- In Chapter 2 the used formulation of the incompressible, unsteady Reynolds-averaged Navier-Stokes Equations is presented. It is also compared to some other variants.
- In Chapter 3 the Finite Volume method used for the spatial discretization is described. This includes the basic principle and the most important discretization methods for unstructured grids.
- In Chapter 4 the temporal discretization methods are described. Standard 1st and 2nd order schemes are given (*forward Euler*, *backward Euler*, *Crank-Nicolson* and *BDF(2)*). After a discussion on the stability, a new stabilized 2nd order scheme is derived and its results are compared to the standard schemes using a 2D simulation with sea waves.
- In Chapter 5 the governing equation for the Volume-of-Fluid method is given. Furthermore different high-resolution discretization schemes, *HRIC*, *BICS* and *BRICS*, used for the discretization of the convective term are presented.
- In Chapter 6 the k-ω-SST turbulence model and the here used modifications are described. The modifications consider the density treatment, wrong wave damping due to infinite growing of the turbulence intensity in the waves and the inherently given turbulent decay.

[7] A *Velocity Prediction Program* determines the different boat speeds and driving conditions for different wind and course conditions for a sailing yacht. Such programs are used by yacht designers to find the best possible combination from different hull and sail geometries. Usually, the hull and sails are considered separately beforehand for the required input. Their properties can be determined by simple empirical formulas, by experimental investigations or by CFD simulations.

- In Chapter 7 the rigid body motion method, especially for a sailing yacht is given. A simple sail model is presented, which allows to include the sail forces into the motion of the yacht without simulating the real sail. Furthermore a stabilization approach is presented, which is required as the solution of the motion equation is based on an explicit 2nd order time discretization. Finally, the results are validated with experimental results and CFD results from Star-CCM+ using a cone falling into water.
- In Chapter 8 the solution procedure used to solve the equation system is given. It is mainly based on the *Semi-Implicit Method for Pressure Linked Equations* (SIMPLE).
- In Chapter 9 the here chosen way for the treatment of the density jump at the free-surface is given. The Finite Volume method uses the Gauss-Theorem to convert volume integrals into surface integrals. This theorem is only valid for continuous variables. For the discontinuous density its application leads to stability problems with very high velocities, especially in the air phase making a special jump treatment essential.
- In Chapter 10 the simulation method for sea waves is given. Artificial wave damping is necessary in front of the outlet to suppress wrong wave reflection and the main part of this chapter describes the derivation of a new wave damping method. Also a comparison to another well known wave damping method is given.
- In Chapter 11 the solver is applied onto several test cases. First the *Rayleigh Taylor Instability* is simulated and compared to analytical results. Afterwards, the *Sysser 60* hull of the *Delft Systematic Yacht Hull Series* is simulated and the results are compared qualitatively to the results of Star-CCM+. Finally, several simulation results for the *Wide and Light* test case of the *Sailing Yacht Research Foundation* are given and compared to different other commercial CFD methods and towing tank results.
- In Chapter 12 a short introduction to the newly developed scouring simulation method is given. The complete description of the method is provided in appendix C.
- Some results of Chapters 5 and 9 and Section 11.2 have been presented previously in Meyer, Renzsch, et al. 2016. This work is attached in appendix A.
- The content of Chapter 10 has been presented in Meyer, Graf, and Slawig 2017 and is attached in appendix B.
- The whole simulation method for the scour solver has been presented in Meyer, Graf, and Slawig 2021. The basic numerical method is based on the approaches described in Chapters 2 to 6, 8 and 9. All additional methods belonging to the scour simulation are described only in the publication attached in appendix C.

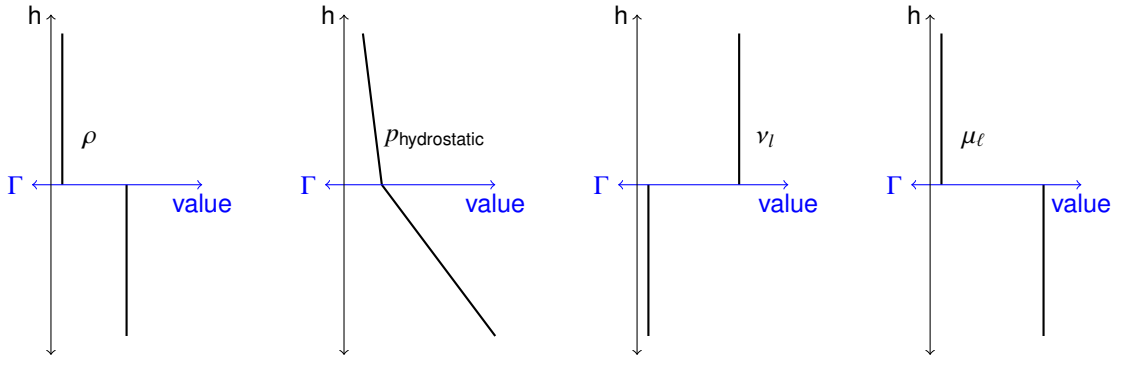
2. Governing Equations

This chapter gives an overview and discussion about the relevant governing equations. Section 2.1 will first described the governing equations in the form used in the present work, afterward alternative formulations for the momentum equation are given in Section 2.2. All versions require special treatments for the discretization. Therefore, knowledge about the characteristics of the flow variables, the ensuing discretization method and the solution method is absolutely necessary for a discussion of the different formulations. Hence, this chapter has to anticipate some information.

Figure 2.1 shows the characteristics of the main flow variables at the free-surface in macroscopic view. The water-surface is represented by the interface Γ. Air is assumed to be above this interface and water below. At a sharp surface the density has a jump at the interface. This leads to a kink in the slope of the hydrostatic pressure and furthermore to a jump in the gradient of the hydrostatic pressure. Additionally the laminar kinematic and dynamic viscosities have a jump at the interface. The velocity $\mathbf{u}$ is not shown in the figure, but it is assumed to be continuous at the free-surface.

The solution domain will be discretized by an arbitrary unstructured grid with cell-centered variables. All equations will be discretized using the Finite Volume method. Hence, the equations will be integrated over the volume and afterward, the general Gauss theorem is applied to the terms containing spatial derivatives, converting the volume integrals into surface integrals. This leads to a face-based method, requiring proper interpolation schemes to interpolate the cell-centered variables to the cell-faces. The application of the Gauss theorem is only valid if the slope of the variable is continuous inside the control-volume.

The governing equations are solved with an SIMPLE like algorithm. This algorithm derives a pressure equation with the help of the semi-discretized momentum equation and the continuity equation. At this point it is important to mention, that this pressure equation is a Laplace equation containing



(a) density **(b)** hydrostatic pressure **(c)** laminar kinematic viscosity **(d)** laminar dynamic viscosity

Figure 2.1.: Characteristics of problematic flow variables at the water-air interface Γ

the laplacian of the pressure. As previously mentioned the gradient of the pressure has a jump at the free-surface leading to an undefined behavior for the laplacian if not using any special treatment. This topic is covered in Chapter 9.

2.1. Used Formulation

For the calculation of the flow the incompressible, unsteady Reynolds-averaged Navier-Stokes equations are solved using the Finite Volume method. The Volume-of-Fluid (VOF) method introduced by Hirt and Nichols 1981 is used for the calculation of the free-surface. The momentum conservation equation, the continuity equation and the conservation equation for the transport of the volume fraction α are defined as

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \nabla \cdot \mu_{\mathrm{e}} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) = -\nabla p + \rho \mathbf{g} + \mathbf{q} \tag{2.1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.2}$$

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\alpha_i \mathbf{u}) = 0 \tag{2.3}$$

with the density ρ, the velocity vector $\mathbf{u}$, the pressure p, the gravity vector $\mathbf{g}$, the effective dynamic viscosity $\mu_{\mathrm{e}} = \mu_\ell + \mu_{\mathrm{t}}$ consisting of the laminar dynamic viscosity μ_ℓ and the dynamic eddy viscosity μ_{t}, and with the volume fraction α_i for the ith phase. Optional source terms are represented by the vector $\mathbf{q}$. For instance, in the beginning of the simulation an additional horizontal velocity acceleration, $\rho \mathbf{a}_{\mathrm{flow}}$, is used to accelerate the flow from zero to the target velocity. In the remaining work this source term is not presented further. The fluid properties are calculated by

$$\rho = \sum_i \rho_i \alpha_i \,, \quad \mu = \sum_i \mu_i \alpha_i \quad \text{with} \quad 1 = \sum_i \alpha_i \,. \tag{2.4}$$

The dynamic eddy viscosity μ_{t} is calculated with a modified version of OpenFOAMs implementation of the k-ω-SST two equation model Menter, Ferreira, and Esch 2003, see Chapter 6.

For all equations the Volume-of-Fluid method is used to discretize the spatial derivatives. After integrating the equations over the volume, the resulting volume integrals are converted to surface integrals using the Gauss theorem. This leads to a face based method, requiring interpolation schemes, which interpolate the cell-centered variables to the faces.

2.2. Alternative Formulations

The total pressure p of (2.1) consist of the static pressure $\rho g h$ and the dynamic pressure p_{d}

$$p = \rho g h + p_{\mathrm{d}} \tag{2.5}$$

with g as the magnitude of the gravity vector $\mathbf{g}$ and the z-coordinate h. Substituting the total pressure of (2.1) with (2.5) leads to a momentum equation depending on the dynamic pressure

$$\frac{\delta \rho \mathbf{u}}{\delta t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \nabla \cdot \mu_{\mathrm{e}} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) = -\nabla p_{\mathrm{d}} - gh\nabla \rho \tag{2.6}$$

This form of the momentum equation is used for the free-surface solver *interFoam* delivered with OpenFOAM. In this solver the Gauss theorem is applied without any special jump treatment, although the variables are discontinuous. As shown in chapter 11.2 and appendix A this leads to errors, reducing the stability of the solver especially for fine grids.

A different formulation for the momentum equation can be achieved assuming a piece-wise constant density. Under this assumption it is possible to exclude the density from the derivatives in (2.1), which leads to the following formulation

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) - \nabla \cdot \nu_{\mathrm{e}} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) = -\frac{1}{\rho} \nabla p + \mathbf{g} \quad . \tag{2.7}$$

As the derivation is based on the assumption of a piece-wise constant density it is not allowed to use (2.4) for the determination of the density. A valid approach is to reconstruct the free-surface inside the cell and adopt the Gauss-Theorem to this reconstruction. This approach is the basic principle of the so called Ghost Fluid Method (Huang, Carrica, and Stern 2007). A first implementation of this method using the Volume-of-Fluid method on unstructured grid was presented by Vukčević, Hrvoje Jasak, and Gatin 2017.
Although using (2.4) permits (2.7) it is used in some codes in this combination without any correction. Therefore this combination has been investigated in the beginning of this work. Simulating flow around a sailing yacht and initializing the fluid with the final velocity typically leads to a big splash wave at the bow in the beginning of the simulation. Using (2.1) including the density in the derivations, resolves this wave well. While neglecting the derivations of the density without proper correction this wave is completely absent, showing that the introduced error is not negligible. Due to this reasons the equations in Section 2.1 are used in this work.

3. Finite Volume Discretization

This chapter describes the Finite Volume method (FVM), which is applied to convert the spatial terms of the governing equations into the equation system. The treatment of time derivatives is explained in the next chapter. The Finite Volume method is very suitable for computational fluid dynamics due to its inherently given conservative properties. A detailed description of the Finite Volume discretization applied in OpenFOAM is given by Hrvoje Jasak 1996. In the following the Finite Volume method and the most important discretization schemes are explained based on the standard transport equation for a generic quantity ϕ:

$$\frac{\partial \rho_\phi \phi}{\partial t} + \nabla \cdot \left(\rho_\phi \mathbf{u} \phi\right) - \nabla \cdot \left(\Gamma_\phi \nabla \phi\right) = q_\phi \tag{3.1}$$

First, the equation is integrated over the volume leading to

$$\int_V \frac{\partial \rho_\phi \phi}{\partial t} dV + \int_V \nabla \cdot \left(\rho_\phi \mathbf{u} \phi\right) dV - \int_V \nabla \cdot \left(\Gamma_\phi \nabla \phi\right) dV = \int_V q_\phi dV \quad . \tag{3.2}$$

Afterwards, the generalized form of the Gauss' theorem,

$$\int_V \nabla \cdot \mathbf{a} dV = \int_S \mathbf{a} \cdot d\mathbf{S} = \int_S \mathbf{a} \cdot \mathbf{n} dS \quad , \tag{3.3}$$

$$\int_V \nabla \phi dV = \int_S \phi d\mathbf{S} = \int_S \phi \mathbf{n} dS \quad , \tag{3.4}$$

$$\int_V \nabla \mathbf{a} dV = \int_S \mathbf{a} d\mathbf{S} = \int_S \mathbf{a} \mathbf{n} dS \quad , \tag{3.5}$$

is applied to transform the volume integrals into surface integrals resulting in

$$\frac{\partial \rho_\phi \phi}{\partial t} V + \int_S \left(\rho_\phi \mathbf{u} \phi\right) \cdot \mathbf{n} dS - \int_S \left(\Gamma_\phi \nabla \phi\right) \cdot \mathbf{n} dS = q_\phi V \quad . \tag{3.6}$$

It is important to notice, that the Gauss' theorem requires Lipschitz continuity. As shown later, this is not the case in all relevant governing equations and might require special adopted methods to overcome this issue. Transposing the equation gives the final integral form for the Finite Volume method as used in OpenFOAM

$$\underbrace{\frac{\partial \rho_\phi \phi}{\partial t}}_{\text{temporal derivative}} + \underbrace{\frac{1}{V} \int_S \left(\rho_\phi \mathbf{u} \phi\right) \cdot \mathbf{n} dS}_{\text{convection term}} - \underbrace{\frac{1}{V} \int_S \left(\Gamma_\phi \nabla \phi\right) \cdot \mathbf{n} dS}_{\text{diffusion term}} = \underbrace{q_\phi}_{\text{source term}} \quad . \tag{3.7}$$

To make this equation solvable it has to be transformed into an algebraic equation system. First, the spatial solution domain is divided into a finite number of small control volumes, called cells. Together, these cells build the (unstructured) grid. The cells can be of nearly arbitrary polyhedral shape, like

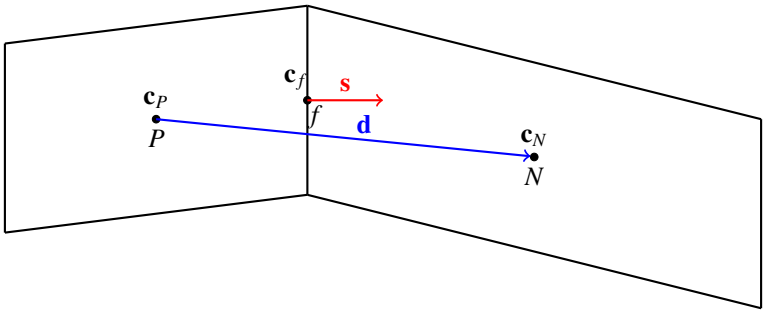


Figure 3.1.: Owner and Neighbor Cell

tetrahedral, prism, hexahedral and other shapes. Indeed, they have to fulfill geometrical criteria like convexity. The flat sides of a cell are called faces and have to be even. Faces defined by three points are inherently even. Faces with more points have to be even too, but usually for complex grids this criterion is no fulfilled exactly introducing a small error. In the given work overlapping cells are not considered, therefore all faces, except boundary faces, belong to two cells. In OpenFOAM these two cells are subdivided into the owner cell P and the neighbor cell N, see Figure 3.1. For their further definition the surface vector $\mathbf{s}$ is used, where $\mathbf{s}$ is perpendicular to the face and has a magnitude equal to the area of the face. This surface vector shows out of the owner cell and shows into the neighbor cell completing the owner/neighbor definition.

The variables are solved only at the cell centers (cell-centered nodes). Such discretization converts the field functions into grid functions. In the following two important mesh quality criteria are presented followed by the description of the treatment of all relevant terms.

3.1. Two Important Mesh Quality Criteria

To further describe the quality of the individual cells or faces, different quality characteristics exist. In this work two important characteristics are relevant:

3.1.1. Skewness

With reference to Figure 3.2 the skewness is defined as

$$Q_{\text{Skewness}} = \frac{|\mathbf{d}|}{|\mathbf{m}|} \quad . \tag{3.8}$$

The skewness can be used to quantify how good a linear interpolation between the two nodes P and N will represent the value at the center of the face. Hence, this characteristics shows how valid the application of the mid-point rule is while using a linear interpolation. The smaller the value, the better the approximation.

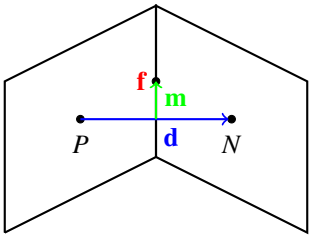


Figure 3.2.: Skewness

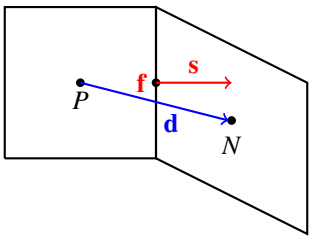


Figure 3.3.: Non-Orthogonality

3.1.2. Orthogonality

With reference to Figure 3.3 the orthogonality is defined as

$$Q_{\text{Orthogonality}} = \frac{\mathbf{s} \cdot \mathbf{d}}{|\mathbf{s}|\,|\mathbf{d}|} \quad . \tag{3.9}$$

The orthogonality describes how good the gradient between the two nodes P and N approximates the surface normal gradient at the midpoint of the face.

3.2. Discretization of the Convection Term

Having in mind, that the faces are assumed to be flat, the surface integrals of (3.7) can be converted into sums. For the convective term this leads to

$$\int_V \nabla \cdot \left(\rho_\phi \mathbf{u}\phi\right) dV = \int_S \left(\rho_\phi \mathbf{u}\phi\right) \cdot \mathbf{n} dS = \sum_f \left(\rho_\phi \mathbf{u}\phi\right)_f \cdot \mathbf{s} \quad , \tag{3.10}$$

where $\mathbf{s}$ is the surface vector as defined previously. The subscript $_f$ marks, that the values have to be evaluated at the face. As mentioned in the previous chapter, the variables are stored at the cell center. Hence, proper interpolation methods between the cell values can be used to get the values at the face. For this the midpoint rule is applied. It assumes that the value at the center of the face is valid for the whole face area. In OpenFOAM the stencil is restricted to the cell and its direct neighbors, therefore only the two values of the cells sharing the same face can be used to create an implicit interpolation scheme, which contributes to the coefficient matrix.

In the following different standard interpolation schemes will be given. Most are based on the flux at the face, which is defined as

$$\Phi = \left(f_\mathrm{x} \mathbf{u}_P + (1 - f_\mathrm{x})\, \mathbf{u}_N\right) \cdot \mathbf{s} \tag{3.11}$$

with the interpolation factor f_x being the ratio of the two distances $\left|\mathbf{c}_N - \mathbf{c}_f\right|$ and $|\mathbf{d}|$ according to Fig. 3.1

$$f_x = \frac{\left|\mathbf{c}_N - \mathbf{c}_f\right|}{|\mathbf{d}|} \quad . \tag{3.12}$$

Central Differencing Scheme (2nd order):

$$\phi_{f\mathrm{CD}} = f_x \phi_P + (1 - f_x)\, \phi_N \tag{3.13}$$

Upwind Differencing Scheme (1st order):

$$\phi_{f\mathrm{UD}} = \begin{cases} \phi_P, & \text{for } \Phi \geq 0 \\ \phi_N, & \text{for } \Phi < 0 \end{cases} \tag{3.14}$$

Downwind Differencing Scheme (1st order):

$$\phi_{f\mathrm{DD}} = \begin{cases} \phi_N, & \text{for } \Phi \geq 0 \\ \phi_P, & \text{for } \Phi < 0 \end{cases} \tag{3.15}$$

Linear Upwind Differencing Scheme (2nd order):

$$\phi_{f\mathrm{LU}} = \phi_{f\mathrm{UD}} + \boxed{(\nabla\phi)_{\mathrm{UD}} \cdot \left(\mathbf{c}_f - \mathbf{c}_P\right)} \tag{3.16}$$

where the gradient $(\nabla\phi)_{\mathrm{UD}}$ is calculated at the center of the upwind cell using the solution of the previous iteration. Hence, the boxed term is treated explicitly and only contributes to the right hand side of the equation system. It is important to notice that the Linear Upwind Differencing Scheme is the only scheme including a correction for skewness.

All schemes, except the upwind scheme, may lead to an unbounded solution and may cause oscillations.

3.3. Discretization of the Diffusion Term

For the diffusion term the surface integrals are converted to surface sums, as before

$$\int_V \nabla \cdot \left(\Gamma_\phi \nabla\phi\right) dV = \int_S \left(\Gamma_\phi \nabla\phi\right) \cdot \mathbf{n} dS = \sum_f \left(\Gamma_\phi \nabla\phi\right)_f \cdot \mathbf{s} \quad . \tag{3.17}$$

Again the subscript $_f$ marks, that the values have to be evaluated at the face. The diffusion variable Γ_ϕ can be obtained using an interpolation scheme as described in previous section. Here, the central differencing scheme is used. For the gradient different approaches are possible. One way is to evaluate the gradient at the cell centers and interpolate it to the faces.

$$(\nabla\phi)_f = f_x\, (\nabla\phi)_P + (1 - f_x)\, (\nabla\phi)_N \tag{3.18}$$

with

$$(\nabla\phi)_C = \frac{1}{V}\sum_f \phi_f \mathbf{s} \tag{3.19}$$

as the gradient at the cell center. Equation (3.18) is a 2nd order approximation also on non-orthogonal meshes, but requires a big computational molecule. Another way is to calculate the product $(\nabla\phi)_f \cdot \mathbf{s}$ directly at the face, which requires a smaller computational molecule.

If the mesh is orthogonal

$$(\nabla\phi)_f \cdot \mathbf{s} = \frac{\phi_N - \phi_P}{|\mathbf{d}|}|\mathbf{s}| \tag{3.20}$$

is a 2nd order approximation.

If the mesh is non-orthogonal an additional correction is required. As described in Hrvoje Jasak 1996, the product $(\nabla\phi)_f \cdot \mathbf{s}$ is split into two parts:

$$(\nabla\phi)_f \cdot \mathbf{s} = (\nabla\phi)_f \cdot \mathbf{s}_{\text{ortho}} + \boxed{(\nabla\phi)_f \cdot \mathbf{s}_{\text{non-ortho}}} \tag{3.21}$$

where the two vectors $\mathbf{s}_{\text{ortho}}$ and $\mathbf{s}_{\text{non-ortho}}$ have to fulfill the following condition

$$\mathbf{s} = \mathbf{s}_{\text{ortho}} + \mathbf{s}_{\text{non-ortho}} \tag{3.22}$$

and $\mathbf{s}_{\text{ortho}}$ has to be parallel to $\mathbf{d}$. This allows to use (3.20) on the orthogonal contribution and (3.18) only at the non-orthogonal part. Furthermore, the orthogonal part contributes to the coefficient matrix, while the boxed, non-orthogonal part is evaluated with the results of the previous iteration and only contributes to the right hand side of the equation system. In Hrvoje Jasak 1996 three methods for the decomposition of $\mathbf{s}_{\text{ortho}}$ and $\mathbf{s}_{\text{non-ortho}}$ are discussed.

As shown in figure 3.4 for the *minimum correction approach* $\mathbf{s}_{\text{ortho}}$ and $\mathbf{s}_{\text{non-ortho}}$ build a right angle

$$\mathbf{s}_{\text{ortho}} = \frac{\mathbf{d}\cdot\mathbf{s}}{\mathbf{d}\cdot\mathbf{d}}\mathbf{d} \quad , \tag{3.23}$$

for the *orthogonal correction approach* $\mathbf{s}$ and $\mathbf{s}_{\text{ortho}}$ have the same length.

$$\mathbf{s}_{\text{ortho}} = \frac{\mathbf{d}}{|\mathbf{d}|}|\mathbf{s}| \tag{3.24}$$

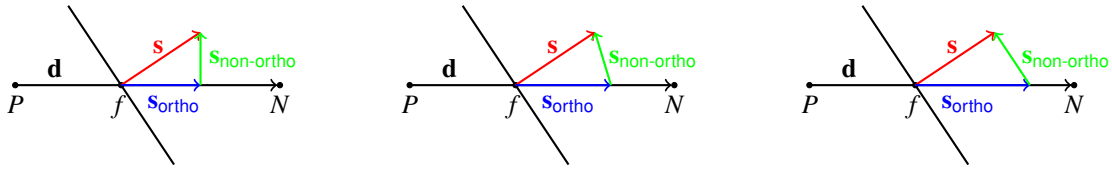
and for the *over-relaxed correction* approach $\mathbf{s}$ and $\mathbf{s}_{\text{non-ortho}}$ build a right angle.

$$\mathbf{s}_{\text{ortho}} = \frac{\mathbf{d}}{\mathbf{d}\cdot\mathbf{s}}|\mathbf{s}|^2 \quad . \tag{3.25}$$

Due to it's good convergence and stability properties the over-relaxed approach is used in OpenFOAM (Hrvoje Jasak 1996).

3.4. Discretization of the Source Term

The final form of the source term is already given in (3.7). For the sake of clarity it should be repeated here, that in OpenFOAM the equation is divided by the cell volume V after integrating over volume.



(a) Minimum correction **(b)** Orthogonal correction **(c)** Over-relaxed correction

Figure 3.4.: Non-orthogonal correction approaches

Therefore the contribution of the source term to the right hand side of the equation system is just

$$q_\phi \quad . \tag{3.26}$$

4. Temporal Discretization

This chapter describes the used marching methods. First the standard 1st and 2nd order methods of OpenFOAM are described. Afterwards the stability of theses methods is categorized and a newly developed modification, used to stabilize one of OpenFOAMs 2nd order scheme is presented. In the end of this chapter, the results of the new discretization scheme are compared to the results of the standard schemes by applying these schemes on different sea wave simulations.

For the discretization of the temporal derivatives the standard methods for initial value problems can be applied. First the time is divided into discrete timesteps with the timestep size Δt. From a given initial value one can march successively from one timestep to next. Such methods are called *Marching Methods* and can be subdivided into *Multistep* (Dahlquist 1956 and Dahlquist 1961) and *Runge-Kutta* (Runge 1895 and Kutta 1901) methods. Multistep methods use the solution of several previous timesteps, whereas Runge-Kutta methods use additional points between the last and the new timestep.

Methods which use the values of the last and the new timestep are called two-level methods, methods which use three points are called three-level methods and so forth.

For most sailing yacht or ship simulations, for example the prediction of drag, only the final, quasi-steady solution is of interest. The whole transient behavior between the initial values and the final values are unimportant. For such cases a stable 1st order method is the right choice.

For transient simulations, like cases with sea waves, a 2nd order scheme or higher is of utmost importance. Only in that case a correct propagation of the sea waves is guaranteed.

In the following the standard time discretization schemes of OpenFOAM are presented. All belong to the Multistep methods. Afterwards a discussion onto the stability of these schemes is given. Based on this, a modified 2nd order scheme is presented, which shows good stability characteristics as presented in the end of this chapter.

4.1. Standard Marching Methods of OpenFOAM

For the derivation it is sufficient to start with the first order ordinary differential equation with an initial condition

$$\frac{\mathrm{d}\phi(t)}{\mathrm{d}t} = f(t, \phi(t)); \qquad \phi(t_0) = \phi^0 \quad . \tag{4.1}$$

Integrating this equation from t to $t_{n+\Delta t}$ with the step size Δt leads to

$$\int_{t_n}^{t_{n+\Delta t}} \frac{\mathrm{d}\phi(t)}{\mathrm{d}t} \mathrm{d}t = \int_{t_n}^{t_{n+\Delta t}} f(t, \phi(t))\, \mathrm{d}t \quad . \tag{4.2}$$

The left hand side can be expressed by $\int_{t_n}^{t_{n+\Delta t}} \frac{\mathrm{d}\phi(t)}{\mathrm{d}t}\mathrm{d}t = \phi^{n+1} - \phi^n$, which is exact. The right hand side can be approximated by numerical quadrature. Using only the known values from the last time step leads to the 1st order accurate *explicit Euler* method (Euler 1768), also known as *forward Euler*

$$\phi^{n+1} = \phi^n + f\left(t_n, \phi^n\right)\Delta t \quad . \tag{4.3}$$

Using only the unknown values of the current time leads to the 1st order accurate *implicit Euler* method, also known as *backward Euler*

$$\phi^{n+1} = \phi^n + f\left(t_{n+1}, \phi^{n+1}\right)\Delta t \quad . \tag{4.4}$$

Approximating the area of the integral by a trapezoid gives the implicit, 2nd order accurate *Crank-Nicolson* method (Crank and Nicolson 1947)

$$\phi^{n+1} = \phi^n + 0.5\left(f\left(t_n, \phi^n\right) + f\left(t_{n+1}, \phi^{n+1}\right)\right)\Delta t \quad . \tag{4.5}$$

Another 2nd order accurate method can be obtained from (4.1) by approximating $\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}$ with a 2nd order polynomial which is constructed by the current and two old points

$$\frac{1}{2\Delta t}\left(3\phi^{n+1} - 4\phi^n + \phi^{n-1}\right) = f\left(t_{n+1}, \phi^{n+1}\right) \quad . \tag{4.6}$$

This method belongs to the group of linear multi-step methods, in this case to the subgroup of *Backward Differentiation Formulas* (BDF) (Curtiss and Hirschfelder 1952 and Gear 1971). As it uses two steps it is named BDF(2). Besides some other special discretization schemes for steady state problems, *backward Euler*, *Crank-Nicolson* and *BDF(2)* are available in OpenFOAM[1].

4.2. Stability

For the application of temporal discretization techniques it is important to discuss their stability behavior. Here, two stability definitions, A-stable (Dahlquist 1963) and L-stable (Ehle 1973) are of importance:

Definition 4.2.1 (A-stable) *A method whose stability domain S satisfies*

$$S \subset \mathbb{C}^- = \{z; \quad Re\, z \leq 0\} \tag{4.7}$$

is called A-stable.

The stability domain S of a numerical method is the set of complex numbers for which the numerical method delivers a monotonically decreasing sequence of approximations when solving the *Dahlquist test equation*, $y' = \lambda y, y(0) = 1, \lambda \in \mathbb{C}$, with fixed steps. The exact analytical solution of the test equation is $y(t) = e^{\lambda t}$. If A-stability is given, the stability domain contains the complete left half-space $\mathbb{C}^-$. When using an A-stable method, the numerical solution has the same asymptotic properties as the analytical solution of the equation without oscillations. For some applications A-stability is already a too strict criterion and weaker formulations like the A_0- or the A_α stability are sufficient. The latter criteria are of no further importance here.

[1] In OpenFOAM these schemes are called *Euler*, *CrankNicolson* and *backward*.

Definition 4.2.2 (L-stable) *A method is called L-stable if it is A-stable and if in addition*

$$\lim_{z\to-\infty} R(z) = 0 \quad . \tag{4.8}$$

Here, $R(z)$ is the stability function. L-stability is given, if the stability function equals zero, if z tends to $-\infty$ on the real axis. If L-stability is given, oscillations will be damped. The backward Euler method is L-stable. Crank-Nicolson and BDF(2) are only A-stable and may give unbounded solutions. Especially for sea waves the lacked L-Stability is an important issue. Hence, a newly developed, stabilized version of BDF(2) is presented in the next section.

4.2.1. New Stabilized 2nd Order Backward Differentiation Formula

For transient simulations, especially simulations with sea waves, 2nd order methods are desired. Therefore it is of high interest to have a reliable, stabilized 2nd order method. If the matrix of the equation system is a M-matrix (see definition 4.2.3) as defined in Hackbusch 2005, p. 46, oscillations may occur only if the right hand side of the equation system is negative (Milovan Perić and Ferziger 2002, pp. 150–151).

Definition 4.2.3 (M-matrix) *A is called M-matrix, if A is non-singular,*

$$a_{\alpha\alpha} > 0 \text{ for all } \alpha,\beta \in I, \tag{4.9}$$

$$a_{\alpha\beta} \leq 0 \text{ for all } \alpha \neq \beta, \tag{4.10}$$

and

$$\mathbf{A}^{-1} \geq \mathbf{0}, \tag{4.11}$$

with $\mathbf{0}$ being the zero matrix.

Assuming that the applied spatial discretization methods will give a M-matrix and positive right hand side, one can see from (4.6), that it is not guaranteed to keep a positive right hand side for the final equation system if the BDF(2) method is applied. This is caused by the sign of the coefficient of ϕ^{n-1}, which may lead to a negative right hand side for large ϕ^n. Under consideration of the given field of use one can guess, that this is the case at wave tails. Due to the large density jump, a negative right hand side can be expected in this region, when applied onto the momentum equation (2.1). For the cell i the BDF(2) contribution to the right hand side of the x-component of the momentum equation is

$$\text{rhs}_{\text{BDF(2),x-momentum},i} = \frac{2\rho_i^n u_{\text{x},i}^n V_i^n}{\Delta t} - \frac{\rho_i^{n-1} u_{\text{x},i}^{n-1} V_i^{n-1}}{2\Delta t} \quad . \tag{4.12}$$

A similar behavior may occur for the Volume-of-Fluid equation (2.3)

$$\text{rhs}_{\text{BDF(2),VOF},i} = \frac{2\alpha_i^n V_i^n}{\Delta t} - \frac{\alpha_i^{n-1} V_i^{n-1}}{2\Delta t} \quad , \tag{4.13}$$

where α_i equals unity, if the cell is filled with water and zero, if filled with air.

Table 4.1.: Test case parameters

	Case 1	Case 2	Case 3
H (wave height)	0.16m	0.16m	0.0016m
λ (wave length)	4.0m	4.0m	0.04m
T (wave period)	1.6s	1.6s	0.16s
u_c (velocity magnitude of current)	$0.0\frac{m}{s}$	$0.5C_p$[1]	$0.0\frac{m}{s}$
Δt (time step size)	0.004s	0.0004s	0.0004s

[1] C_p = phase speed of the wave

This problem is not given for the implicit Euler method. Knowing this and assuming that the problem occurs only in a few cells at the free-surface a new *non oscillating BDF(2)* method was constructed

$$\text{non oscillating BDF(2)}_i = \begin{cases} \text{BDF(2)}_i & \text{if } \left(4\phi_i^n - \phi_i^{n-1}\right)\phi_i^n \geq 0 \\ \text{implicit Euler}_i & \text{else} \end{cases} . \tag{4.14}$$

As shown in (4.14) the scheme locally switches to the first order implicit Euler scheme, if the right hand side belonging to cell i would become negative using the BDF(2) method.

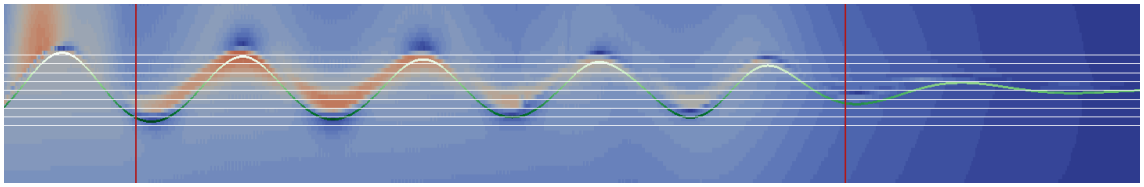
A similar behavior is given for the Crank-Nicolson scheme as $-0.5f\left(t_n, \phi^n\right)\Delta t$ is part of its contribution to the right hand side. This method has not been further investigated in this work.

4.2.2. Demonstration of the Stability Improvements

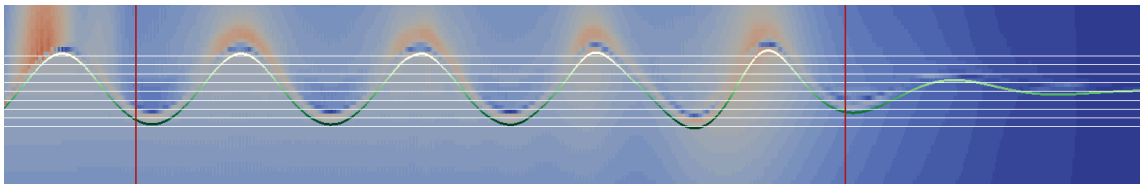
To demonstrate the non oscillating behavior and accuracy of the new non oscillating BDF(2) scheme, the different discretization schemes were applied on a 2D-simulation with sea waves. The domain has a length of 7 wave lengths and a height of 4.5 wave lengths with a water depth of 4 wave lengths. First order Airy waves are generated on the left side of the domain using adequate boundary conditions and an implicit relaxation zone with a length of one wave length for the generation of the waves. To avoid wave reflections, a wave damping zone with a size of two wave lengths is located in front of the outlet on the right side of the domain. In the region of the free-surface the cells size is chosen to resolve the wave length with 100 and the wave height with 20 cells, which is a common standard. Three cases were investigated, see Tab. 4.1.

Fig. 4.1 to 4.3 show the results of the simulations. The free-surface is represented by the green-white line, the results are stretched in the vertical direction. The blue-red background color shows the magnitude of the fluid velocities. The vertical red lines mark the limits of the relaxation zones.
The implicit Euler scheme does not have any stability issues, but in all cases the wave height is damped significantly as the wave energy is preserved poorly, only.
The backward scheme shows good results for Case 1 and 3. For Case 2 very high velocities occur above the free surface, especially at the wave trails as expected. It should be mentioned, that in other simulations - for example some simulations with a sailing yacht, where much smaller cells are located around the hull - the problem was much larger so that the simulations failed completely.

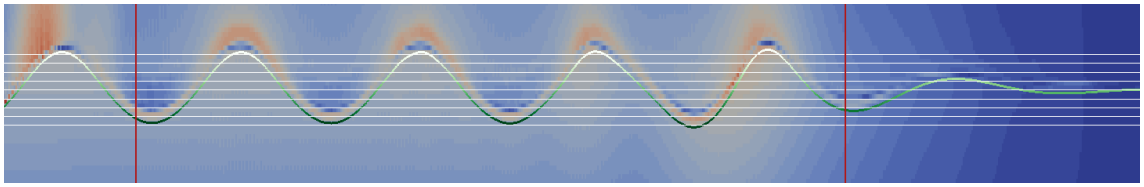
The Crank-Nicolson scheme leads to good results for Case 1 and 2, but completely fails for case 3. Case 3 has been simulated again with a smaller time step. Under this conditions, the Crank-Nicolson scheme gives good results, too. Nevertheless, this results in a significant increase of simulation time and is not practical for an end-user.
Thus, none of the standard schemes deliver good and reliable results for all three cases.
On the other hand, the new non oscillating BDF(2) scheme shows good results for all cases. The wave height is not damped, as the first order method is applied only for a few cells. The figures also show the cells, where the first order method is used (black cells in subfigures (e)). Again, one can see, that this is done only for a few cells at the wave trails, mainly.



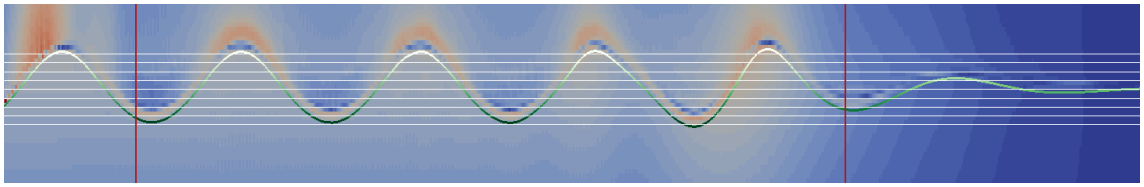
(a) Euler



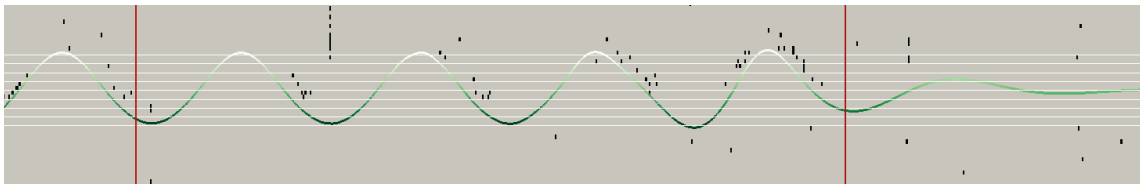
(b) Crank-Nicolson



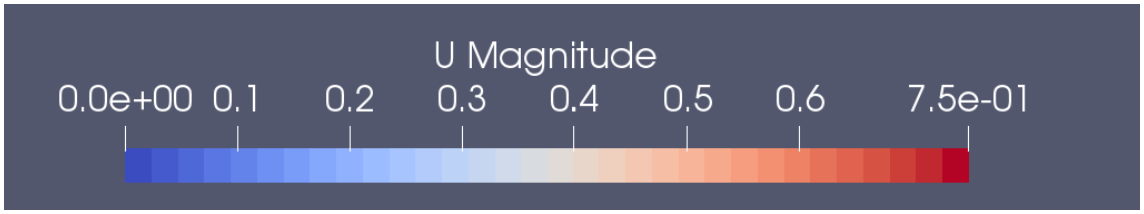
(c) BDF(2)



(d) non oscillating BDF(2)

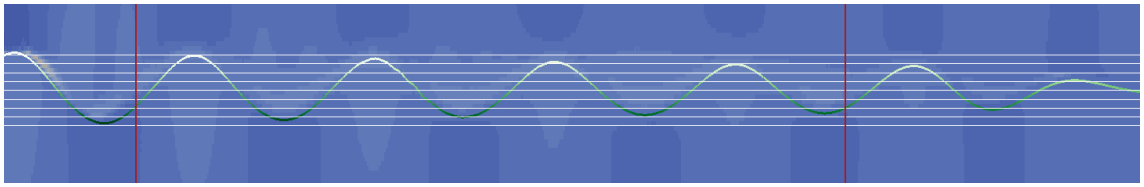


(e) non oscillating BDF(2): black cells are switching to implicit Euler

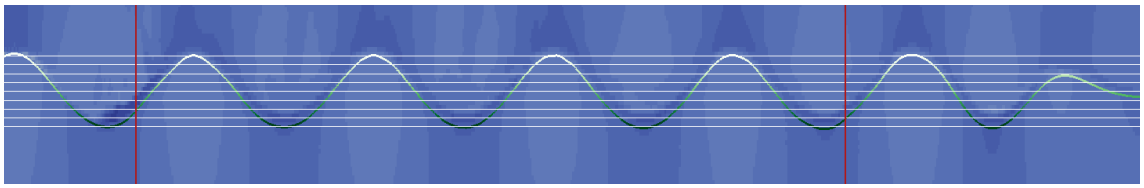


(f) legend [m]

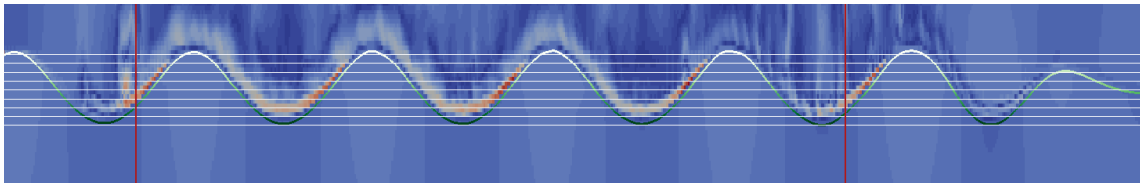
Figure 4.1.: Comparison of different time discretization schemes: Case 1



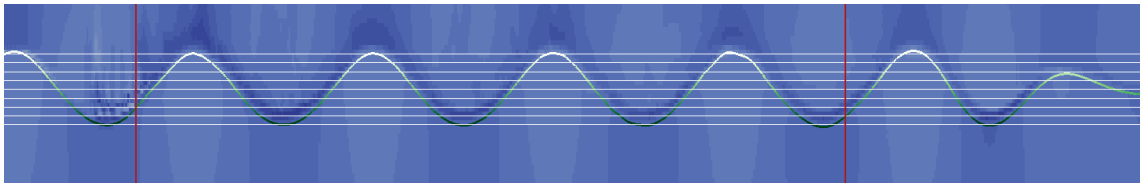
(a) Euler



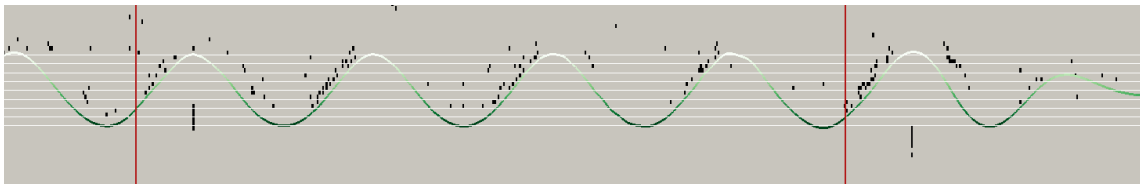
(b) Crank-Nicolson



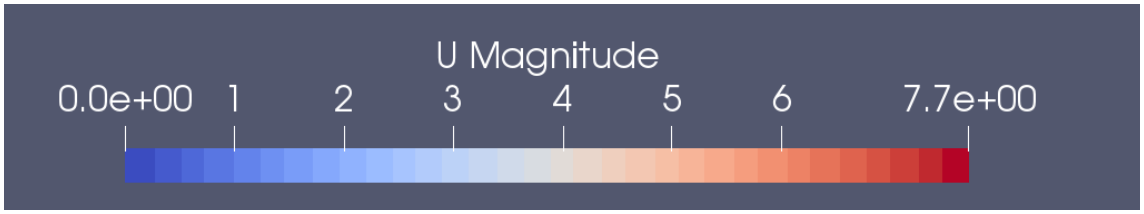
(c) BDF(2)



(d) non oscillating BDF(2)

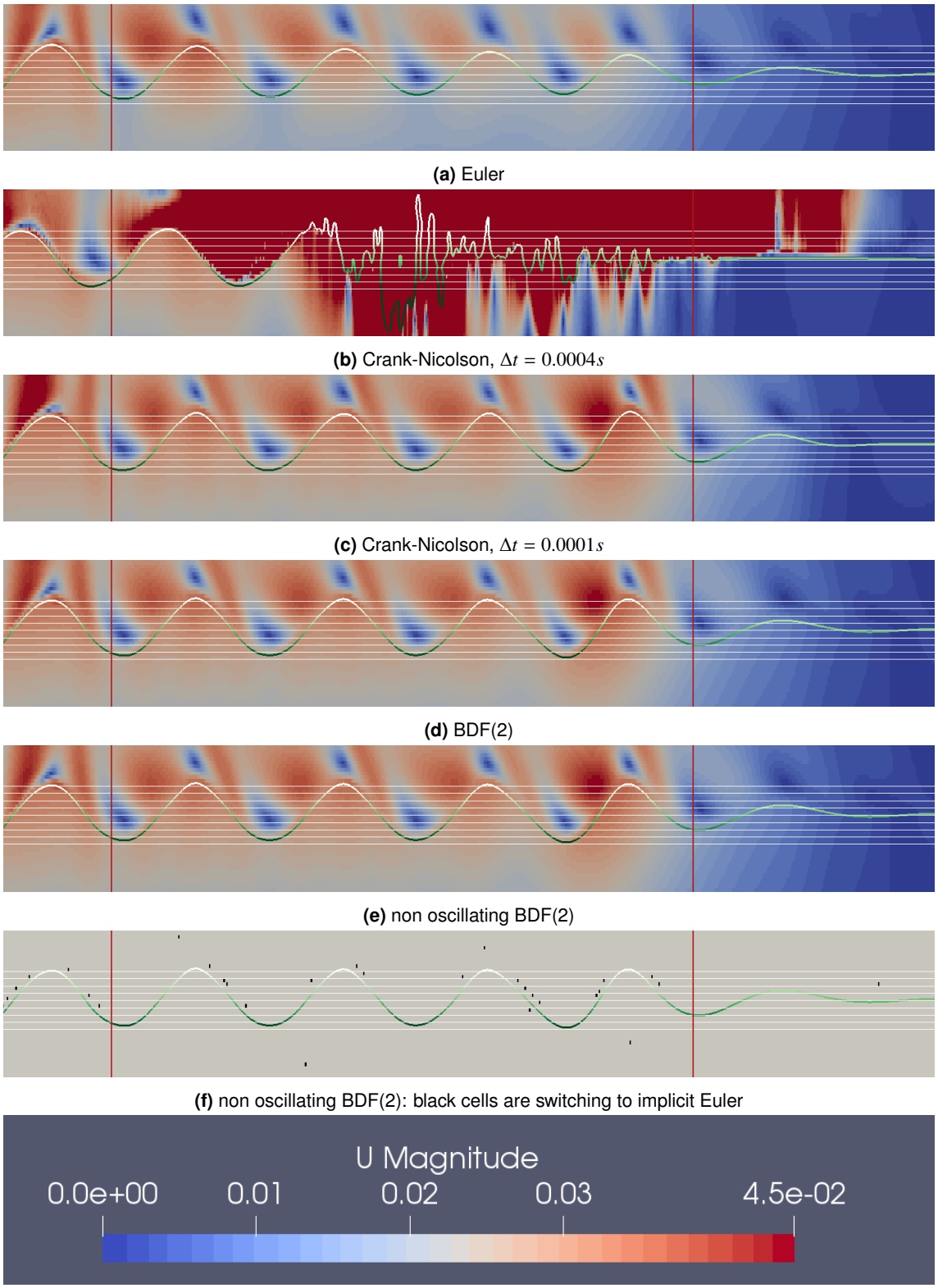


(e) non oscillating BDF(2): black cells are switching to implicit Euler



(f) legend [m]

Figure 4.2.: Comparison of different time discretization schemes: Case 2



(a) Euler

(b) Crank-Nicolson, $\Delta t = 0.0004s$

(c) Crank-Nicolson, $\Delta t = 0.0001s$

(d) BDF(2)

(e) non oscillating BDF(2)

(f) non oscillating BDF(2): black cells are switching to implicit Euler

(g) legend [m]

Figure 4.3.: Comparison of different time discretization schemes: Case 3

5. Volume-of-Fluid Method

For the consideration of the two phases, water and air, the *Volume-of-Fluid* (VOF) method is applied. This chapter describes the Volume-of-Fluid method. First a short description of OpenFOAMs VOF-approach is given. Aftwards the newly implemented approach based on known high-resolution schemes is presented. In the end of this chapter a newly developed method for the suppression of the remaining numerical ventilation is given.
In ship hydrodynamics the correct calculation of the waves generated by the ship is an important requirement on the numerical simulation. These generated waves lead to the wave resistance which is one main part of the total resistance of the ship. Ship waves result in different requirements on the simulation which are difficult to solve. For a racing yacht with a high Froude-Number most of the free-surface has to have a sharp border between water and air. On the other hand on the bow it may come to breaking waves and spray. Getting these different properties into one method also under consideration of the conservation of mass and an acceptable calculation time is a tough task and part of current research.

A lot of methods exist trying to solve these phenomenons. The methods could be divided into *Interface Capturing Methods*, *Interface Tracking Methods* and *Hybrid Methods* which are a mixture of former. A more obvious way to distinguish between these methods suggested in Wackers 2007 and Wackers et al. 2011 is to divide them into *Fitting Methods*, *Capturing Methods with Reconstruction* and *Capturing Methods without Reconstruction*. None of these methods is able to fulfill the requirements sufficiently.

Fitting Methods treat the free-surface as an sharp interface between two immiscible fluids with their own initial boundary conditions. The grid has to be morphed accordingly to the interface. Advantages of these methods are the possibility to solve the two phases separately or only the water phase which avoids problems like the density jump in the equation-system and results in a fast calculation time. On the other hand it is hard to impossible to resolve overturning waves, extreme wave shapes and spray.

For capturing-methods with reconstruction the grid is not necessarily deformed. Instead they use marker particles or convected continuous functions to get the water surface definition for example. Methods of this kind like the *Level-Set Method* (Sussman, Smereka, and Osher 1994) give a well defined interface and also allow overturning waves. Though these methods are not able to resolve spray. Furthermore the conservation of mass is critical and requires a re-initialization of the function from time-to-time.

Capturing-methods without reconstruction like the *Volume-of-Fluid* (VOF) method (Hirt and Nichols 1981) use a volume-fraction equation to determine the amount of each fluid in the cells. The local fluid

properties like the density are set as a mixture of all fluids considering the local volume-fraction. This results in the same equations like for single phase flow but with discontinuities in the equation-system at the surface for example. Additionally one has to solve an transport equation for the volume-fraction. Capturing-methods allow complex wave shapes including breaking waves and spray while guaranteeing mass conservation. However it is hard to obtain a sharp interface and boundedness. This requires additional effort and still results in interface-smearing over one to three cells. The conservation equations becomes harder to solve due to the discontinuities.

A detailed description of the methods and there advantages and disadvantages, respectively is given by Böhm 2014. It is also shown that the VOF method is the current state-of-the-art with respect to the requirements of ship especially yacht hydrodynamics. Due to this reasons the VOF method has been chosen for the solver explained here.

In Marić, D. B. Kothe, and Bothe 2020 the VOF methods are devided into *algebraic* and *geometric* VOF-methods. The algebraic methods invoke continuum-based partial differential equations for the convection of the volume fraction, like the Volume-of-Fluid equation (5.2) shown in the next chapter. The geometric methods rely on geometrical operations to approximate the convection of the volume fraction. In Marić, D. B. Kothe, and Bothe 2020 it is tried to give a complete overview of classical, as well as contemporary geometrical VOF methods, while focusing primarily on unstructured meshes and three dimensional calculations.

5.1. Volume-of-Fluid Transport Equation

The Volume-of-Fluid method introduces one new field variable α for each fluid phase indicating the amount of each phase in the cell. Here α_w represents the volume fraction of the water and α_a the volume fraction of the air. Each volume fraction variable can take values within the interval $[0, 1]$. For example $\alpha_w = 1$ stands for a cell filled with water and $\alpha_w = 0$ for a cell completely filled with air. Intermediate values indicate, that the free-surface is located in the given cell. The sum of all volume-fractions has to be unity at each control volume to guarantee volume conservation

$$\sum \alpha_i = 1 \quad . \tag{5.1}$$

Therefore, if n_{phases} is the number of fluid phases, one has to calculate the transport of $n_{\text{phases}} - 1$ volume fractions and the last one can be calculated explicitly with (5.1). In the following only two phases water and air are considered. Usually one estimates the volume fraction of the heavy fluid, here water, and calculates the volume fraction of the air by applying (5.1). This is also done here and in the following the generic volume fraction α stands for the water phase.

The original Volume-of-Fluid method (Hirt and Nichols 1981) requires the solution of at least one additional partial differential equation to estimate the convection of the volume-fraction. Here, the volume-fraction is calculated with following transport-equation

$$\frac{\delta \alpha}{\delta t} + \nabla(\alpha \mathbf{u}) = 0 \quad . \tag{5.2}$$

There exist different approaches for the calculation of the flow (Milovan Perić and Ferziger 2002). In the original VOF method (Hirt and Nichols 1981) Eq. (5.2) is solved for the hole domain but the continuity- and momentum-equations are only solved for the water. Another approach (Kawamura and H. Miyata 1995) uses the calculated volume-fraction to get the free-surface and uses this interface as a new boundary. This allows to calculate the two phases separated from each other. Furthermore it is possible to consider the two phases as one fluid what has been done for the given solver. In this case both phases are sharing the same velocity field. The fluid properties like the density are calculated accordingly to the volume-fraction

$$\rho = \rho_w \alpha + \rho_a (1 - \alpha) \tag{5.3}$$

where the subscripts *w* and *a* denote the water and air phase, respectively.

Special methods are required to achieve a sharp interface. Still, the resulting free-surface is usually smeared over one to three cells. This induces a refined mesh in the region of the interface for a satisfying sharpness.

5.2. Original Implementation in OpenFOAM

The standard free-surface solver *interFoam* contributed with OpenFOAM also implements the Volume-of-Fluid method. In its VOF transport equation an additional compressive term is used to guarantee a sharp interface. The transport equation is solved using the *Multi-Dimensional Limiter for Explicit Solution* (MULES). It uses an implicit predictor based on solving a modified version of (5.2) using upwind differencing for the convective term. Afterwards, the solution is corrected with an explicit corrector using the 2nd order van Leer (van Leer 1979) discretization scheme, which belongs to the *Total Variation Diminishing* (TVD) (Sweby 1984) scheme family. Although the additional interface compression term is used, the interface tends to be smeared. Furthermore, as shown in 11.2 the compression term seems to flatten the waves and the results do not have the same quality as commercial solvers like Star-CCM+.

The newest OpenFOAM versions offer additional ways for the calculation of the volume-fraction. Besides the MULES algorithm it is possible to chose between the *Iso-Advector* (Roenby, Bredmose, and Hrvoje Jasak 2016) or the *Piecewise-Linear Interface Calculation* (PLIC) (see Brackbill, D. Kothe, and Zemach 1992 for example) approach. Both approaches do not apply the VOF transport equation. Instead, they first reconstruct the free-surface from the given volume-fraction field. Than, the free-surface is convected using an explicit marching method. Finally, the new solution for the volume-fraction is derived from the new position of the free-surface. Such methods, deliver a very sharp interface, nevertheless, due to the application of an explicit marching method, they might suffer from stability problems while using large time steps.

Both isoAdvector and PLIC have not been investigated any further within this work.

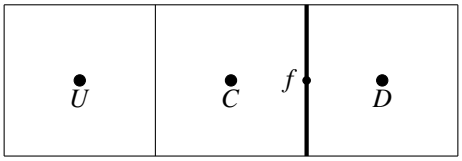


Figure 5.1.: Cell naming used in the Normalized Variable Approach

5.3. New Implementation Using High-Resolution Schemes

As mentioned above the Volume-of-Fluid method requires special methods for a sharp interface. One way is to add additional terms to the equations for an artificial interface-compression. Another possibility is to use a high-resolution scheme for the discretization of the convective term in Eqn. (5.2). Such schemes should transform every gradient into a step function to guarantee sharpness. The solution has to be bounded, which means that the volume-fraction α has to lie between zero and unity. Additionally, the scheme should allow a Courant-Friedrichs-Lewy-Number (CFL) larger unity to decrease computational costs. Furthermore the scheme should prevent numerical oscillation. These requirements result in a large family of discretization schemes. The part of these schemes treated here is based on the *Normalized Variable Diagram* (NVD).

This section first describes the principle of the NVD. Furthermore, an introduction to the main principle for boundedness and the avoidance of numerical oscillation is given. Afterwards the newly implemented high-resolution schemes *High-Resolution Interface Capturing Scheme* (HRIC), *Blended Interface Capturing Scheme* (BICS) and *Blended Interface Capturing Scheme with Reconstruction* (BRICS) are described. These schemes are used for the convective term. Using high-resolution schemes requires additional effort for the solution of the VOF equation. This is described at the end of this section.

Another well-known high resolution scheme is the *Compressive Interface Capturing Scheme for Arbitrary Meshes* (CICSAM) (Ubbink 1997). Due to its strict Courant number limitation it was not investigated any further within this work.

5.3.1. Normalized Variable Approach

The *Normalized Variable Approach* (B. P. Leonard 1988) is a common way to define convection schemes. Considering three neighboring cells the scheme for a generic quantity ϕ can be formulated as

$$\phi_f = f(\phi_U, \phi_C, \phi_D) \quad . \tag{5.4}$$

Here the subscripts U, C and D stands for the far upwind, central and downwind cell and depend on the flow direction. The subscript f stands for the face between the central and downwind cell, as shown in Fig. 5.1. On structured grids the NVA defines the normalized variable $\tilde{\phi}$ as

$$\tilde{\phi} = \frac{\phi - \phi_U}{\phi_D - \phi_U} \quad . \tag{5.5}$$

Based on this formula one can calculate the normalized values for the three cells with

$$\tilde{\phi}_U = 0 \quad , \tag{5.6}$$

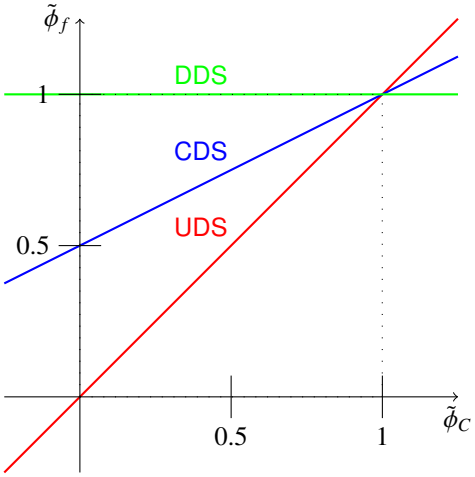


Figure 5.2.: Normalized Variable Diagram

$$\tilde{\phi}_C = \frac{\phi_C - \phi_U}{\phi_D - \phi_U} \tag{5.7}$$

and

$$\tilde{\phi}_D = 1 \quad . \tag{5.8}$$

The basic idea of the NVA is to define a scheme depending on $\tilde{\phi}_C$. Now the definition is simplified to

$$\tilde{\phi}_f = f(\tilde{\phi}_C) \tag{5.9}$$

but still considers the three neighboring cells. Figure 5.2 exemplifies the upwind differencing scheme (UDS), downwind differencing scheme (DDS) and the central differencing scheme (CDS) in the *Normalized Variable Diagram* (B. Leonard 1991). The diagram plots the resulting normalized face value $\tilde{\phi}_f$ over the normalized cell value $\tilde{\phi}_C$ of the central cell. The final face value is obtained by denormalizing $\tilde{\phi}_f$

$$\phi_f = \tilde{\phi}_f(\phi_D - \phi_U) + \phi_U \quad . \tag{5.10}$$

This principle can be implemented easily for structured grids. For arbitrarily unstructured meshes it is not obvious how to determine the far upwind cell and the concept becomes quite complicated. The calculation of the upwind cell is explained later in Section 5.3.6.

5.3.2. Boundedness and Oscillations (Convection Boundedness Criterion)

To guarantee boundedness the *Convection Boundedness Criterion* (CBC) (Gaskell and Lau 1988) is a sufficient criterion. This criterion says that a numerical approximation is bounded if one bound $\tilde{\phi}_f$ as follows

$$\tilde{\phi}_f := \begin{cases} \tilde{\phi}_C & \text{if } \tilde{\phi}_C < 0 \\ [\tilde{\phi}_C, 1] & \text{if } 0 \leq \tilde{\phi}_C \leq 1 \\ \tilde{\phi}_C & \text{if } 1 < \tilde{\phi}_C \end{cases} . \tag{5.11}$$

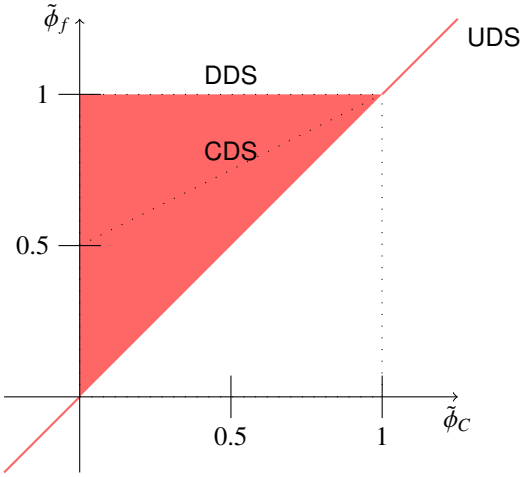


Figure 5.3.: Visualization of the Convective Boundedness Criterion

Figure 5.3 shows the valid regions of the CBC in the NVD. The CBC is valid for the red area and the line $\tilde{\phi}_f = \tilde{\phi}_C$. For $0 \leq \tilde{\phi}_C \leq 1$ the valid schemes lies between the upwind and the downwind scheme. For $\tilde{\phi}_C < 0$ and $\tilde{\phi}_C < 1$ only the the upwind scheme fulfills the CBC. Concise the CBC leads to a blend between the UDS and the DDS. Many schemes use the CBC to guarantee boundedness. As shown in Tao 2000 and B. Yu et al. 2001 this is only a sufficient condition and not a necessary one. The CBC region could be extended to the *Extended Convective Boundedness Criterion* (ECBC) region, for example. Here, the ECBC is not handled any further.

Higher-Order schemes tend to nonphysical oscillation. One example is the 2nd order accurate CDS. The oscillation can be decreased if the scheme satisfies the CBC. Additionally the scheme should be based on a continuous function. Otherwise, small changes of $\tilde{\phi}_C$ may lead to nonphysical oscillation and may also lead to numerical instability.

5.3.3. Stability in Time (CFL-Condition)

For the stability in time the *Courant-Friedrichs-Lewy Condition* (CFLC) (Courant, Friedrichs, and Lewy 1928) is a necessary condition for numerical stability. The CFLC is defined as

$$C_\mathrm{o} \leq C_\mathrm{limit} \tag{5.12}$$

with C_o as the CFL of the local cell calculate by

$$C_\mathrm{o} = \frac{\Delta t}{V} \sum_f \mathbf{u} \cdot \mathbf{s} \tag{5.13}$$

and C_limit as the limiting CFL. The limiting CFL depends on the discretization method used for the discretization of the time derivatives. If an explicit method is used a typical value is $C_\mathrm{limit} = 1$. Implicit methods often allows a higher CFL. Furthermore the order of the discretization scheme influences the maximal allowed CFL. In most cases an higher order reduces C_limit.

The following sections describe three well known high-resolution schemes, which have been imple-

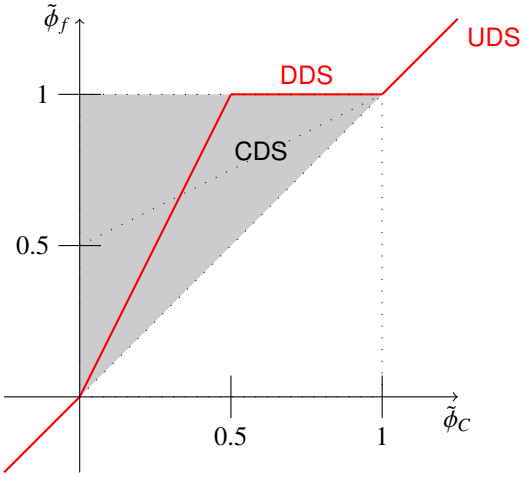


Figure 5.4.: HRIC base function

mented in OpenFOAM during this work. All of them fulfill the previously defined criteria.

5.3.4. High-Resolution Interface Capturing Scheme (HRIC)

The High-Resolution Interface Capturing scheme (HRIC) (Muzaferija and M. Perić 1997, Muzaferija and M. Perić 1999 and Muzaferija, M. Perić, et al. 1999) is a blend of the upwind and downwind differencing schemes and is based on the NVD. In the given field of use the generic quantiy ϕ represents the volume fraction α and its normalized value is $\tilde{\phi}$. The construction of this schemes starts with the HRIC base function

$$\tilde{\phi}_f = \begin{cases} \tilde{\phi}_C & \text{if } \tilde{\phi}_C < 0 \\ 2\tilde{\phi}_C & \text{if } 0 \leq \tilde{\phi}_C < 0.5 \\ 1 & \text{if } 0.5 \leq \tilde{\phi}_C < 1 \\ \tilde{\phi}_C & \text{if } 1 \leq \tilde{\phi}_C \end{cases} . \tag{5.14}$$

The aim of the function is to use the compressive DDS while fulfilling the CBC. Figure 5.4 shows that $\tilde{\phi}_f$ is a continuous function and fulfills the CBC. The value from the base function is than corrected by the local CFL

$$\tilde{\phi}_f^* = \begin{cases} \tilde{\phi}_f & \text{if } C_\text{o} < C_\text{oLower} \\ \tilde{\phi}_C + (\tilde{\phi}_f - \tilde{\phi}_C)\frac{C_\text{oUpper} - C_\text{o}}{C_\text{oUpper} - C_\text{oLower}} & \text{if } C_\text{oLower} \leq C_\text{o} < C_\text{oUpper} \\ \tilde{\phi}_C & \text{if } C_\text{oUpper} \leq C_\text{o} \end{cases} . \tag{5.15}$$

This correction implies that $\tilde{\phi}_f^*$ is used for a local CFL less than C_oLower and upwind differences for a local CFL greater or equal to C_oUpper. Between these limiting numbers a blend of upwind and downwind differences is used. Usually, the values 0.5 and 1 are chosen for C_oLower and C_oUpper. This results in pure upwind differencing for a CFL greater than 1 which increases stability. If stability is not the top priority higher values could be set. As shown in Böhm and Graf 2014 this is recommended for quasi-steady yacht simulations to reduce the numerical ventilation. Therefore, if one is interested in a steady state solution one may set both values, C_oLower and C_oUpper, to a value higher than the maximum local CFL. This will deactivate the CFL blending completely and may lead to less numerical ventilation.

In the last step a final correction regarding the interface angle θ is applied.

$$\tilde{\phi}_f^{**} = \tilde{\phi}_f^{*}(\cos\theta_f)^{C_0} + \tilde{\phi}_C[1-(\cos\theta_f)^{C_0}] \tag{5.16}$$

Here, C_0 is the scheme parameter, its default value is 0.05. The angle θ_f is the angle between the face normal and the interface normal. It is calculated by

$$\cos(\theta_f) = \frac{|\vec{\nabla}\phi \cdot \vec{CD}|}{||\vec{\nabla}\phi||||\vec{CD}||} \tag{5.17}$$

This last correction reduces the tendency of the interface to align with the grid. Finally the convected cell-face value of the volume fraction can be determined as:

$$\phi_f^{\text{HRIC}} = \tilde{\phi}_f^{**}(\phi_D - \phi_U) + \phi_U \,. \tag{5.18}$$

5.3.5. Blended Interface Capturing Scheme (BICS)

The *Blended Interface Capturing Scheme* (BICS) (Wackers et al. 2011) is similar to the HRIC scheme. The main difference between these schemes, is that the HRIC scheme is blending to the first order upwind scheme, whereas the BICS scheme is blending to the second order accurate *Gamma Differencing Scheme* (GDS) (H. Jasak, Weller, and Gosman 1999).
The construction of the BICS scheme starts with the Inter-Gamma Differencing Scheme (IGDS) (H. Jasak, Weller, and Gosman 1999)

$$\tilde{\phi}_{\text{IGDS}} = \begin{cases} \tilde{\phi}_C & \text{if } \tilde{\phi}_C < 0 \\ -\frac{\tilde{\phi}_C^2}{\beta_{\text{IGDS}}} + (1+\frac{1}{\beta_{\text{IGDS}}})\tilde{\phi}_C & \text{if } 0 \le \tilde{\phi}_C < \beta_{\text{IGDS}} \\ 1 & \text{if } \beta_{\text{IGDS}} \le \tilde{\phi}_C \le 1 \\ \tilde{\phi}_C & \text{if } 1 < \tilde{\phi}_C \end{cases} . \tag{5.19}$$

The IGDS is very similar to the HRIC base function. It fulfills the convective boundedness criterion and is a continuous function, see Fig. 5.5. The scheme parameter β_{IGDS} is usually set to 0.5 to guarantee stability. The IGDS is limited to a local Courant Number of 0.3. Therefore the BICS scheme blends to the GDS, depending on the local Courant-Friedrichs-Lewy number. The major part of the *Gamma* scheme is described by

$$\tilde{\phi}_{\text{GDS}} = \begin{cases} \tilde{\phi}_C & \text{if } \tilde{\phi}_C < 0 \\ -\frac{\tilde{\phi}_C^2}{2\beta_{\text{GDS}}} + (1+\frac{1}{2\beta_{\text{GDS}}})\tilde{\phi}_C & \text{if } 0 \le \tilde{\phi}_C < \beta_{\text{GDS}} \\ 0.5\tilde{\phi}_C + 0.5 & \text{if } \beta_{\text{GDS}} \le \tilde{\phi}_C \le 1 \\ \tilde{\phi}_C & \text{if } 1 < \tilde{\phi}_C \end{cases} . \tag{5.20}$$

As one can see in Fig. 5.6 the GDS is second order accurate like the CDS but blends to upwind differences to fulfill the CBC. It has no CFL limitations. Usually the scheme parameter β_{GDS} is set to 0.1. Lower values may lack stability and higher values will reduce the accuracy. The CFL dependency

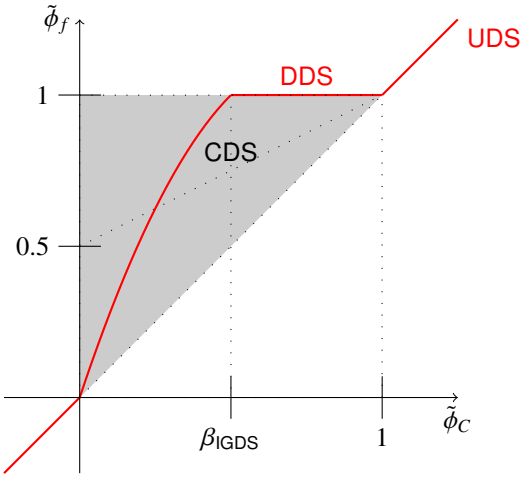


Figure 5.5.: Inter-Gamma Differencing Scheme

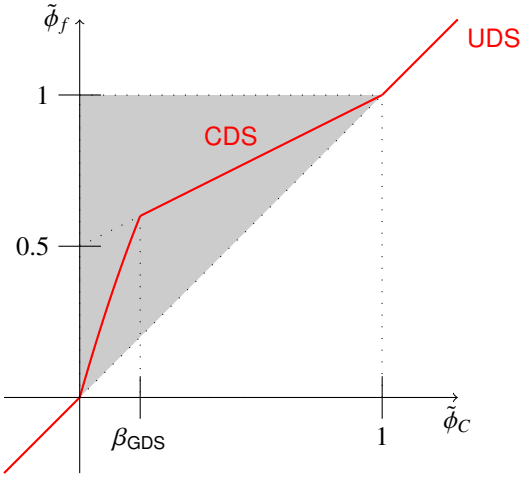


Figure 5.6.: Gamma Differencing Scheme

of the BICS scheme is given in following equation

$$\tilde{\phi}_{\text{BICS}^*} = \begin{cases} \tilde{\phi}_C & \text{if } \tilde{\phi}_C < 0 \\ -\frac{1-p_{\text{BICS}}}{\beta_{\text{BICS}^2}}\tilde{\phi_C}^2 + (p_{\text{BICS}} + \frac{2(1-p_{\text{BICS}})}{\beta_{\text{BICS}}})\tilde{\phi}_C & \text{if } 0 \leq \tilde{\phi}_C < \beta_{\text{BICS}} \\ p_{\text{BICS}}\tilde{\phi}_C + (1 - p_{\text{BICS}}) & \text{if } \beta_{\text{BICS}} \leq \tilde{\phi}_C \leq 1 \\ \tilde{\phi}_C & \text{if } 1 < \tilde{\phi}_C \end{cases} \tag{5.21}$$

with the slope p_{BICS} depending on the local Courant number C_o, see (5.22), and the scheme parameter β_{BICS} depending on the slope p_{BICS}, see (5.25). The scheme is using a quadratic variation for the part below β_{BICS} and a linear variation for the region above β_{BICS}. The slope p_{BICS} is calculated by

$$p_{\text{BICS}}(C_o) = \alpha_p(C_o)p_{\text{IGDS}} + (1 - \alpha_p(C_o))p_{\text{GDS}} \tag{5.22}$$

with

$$\alpha_p(C_o) = \begin{cases} 1 & \text{if } C_o \leq 0.3 \\ \frac{C_o - 0.3}{e^{(C_o-0.3)}-1} & \text{if } 0.3 < C_o \end{cases} \tag{5.23}$$

and

$$p_{\text{GDS}} = 0.5; \quad p_{\text{IGDS}} = 0\,. \tag{5.24}$$

The scheme parameter β_{BICS} is calculated by

$$\beta_{\text{BICS}}(p_{\text{BICS}}) = a_0 + a_1 p(C_o) \tag{5.25}$$

with

$$a_1 = \frac{\beta_{\text{GDS}} - \beta_{\text{IGDS}}}{p_{\text{GDS}} - p_{\text{IGDS}}}; \quad a_0 = \beta_{\text{IGDS}} - a_1 p_{\text{IGDS}} \tag{5.26}$$

with

$$\beta_{\text{GDS}} = 0.1; \quad \beta_{\text{IGDS}} = 0.5 \tag{5.27}$$

as the standard parameters. Finally the classical angle correction from HRIC is applied. Here, the difference between BICS and HRIC is that BICS reduces to the GDS instead of the upwind scheme for the angle correction:

$$\tilde{\phi}_{\text{BICS}} = \tilde{\phi}_{\text{BICS}^*}(\cos\theta)^{C_0} + \tilde{\phi}_{\text{GDS}}[1 - (\cos\theta)^{C_0}]\ . \tag{5.28}$$

Furthermore a value of 0.5 is used for C_0 instead of 0.05 as in the HRIC implementation. For the calculation of the interface angle θ see (5.17).

5.3.6. Calculation of the Upwind Cell and the Blended Interface Capturing Scheme with Reconstruction (BRICS)

For arbitrarily unstructured grids the calculation of the upwind value U is not straight forward, as the upwind cell is not known explicitly. This section describes three ways for the calculation of the upwind value.

The classical approach is to extrapolate to an imaginary upstream node U by the use of the gradient

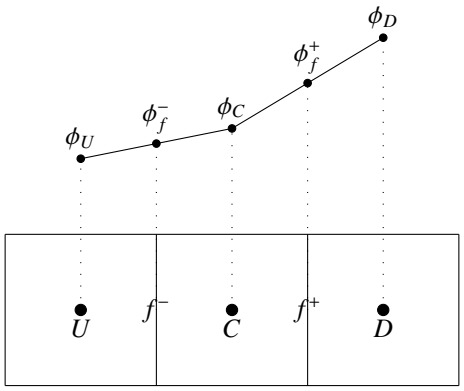


Figure 5.7.: Modified approach for the NVD criterion

projection method (Patrick Queutey and Michel Visonneau 2007).

$$\phi_U = \phi_C - \vec{CU} \cdot \vec{\nabla}\phi_C, \quad with \quad \vec{CU} = -\vec{CD}. \tag{5.29}$$

Another approach renounces the upwind value completely and uses a modification of the NVD criterion (H. Jasak, Weller, and Gosman 1999). Instead of using the upwind and downwind cell, one uses the upwind and downwind face, see Fig. 5.7. Therefore, the new definition of $\tilde{\phi}_C$ is

$$\tilde{\phi}_C = \frac{\phi_C - \phi_f^-}{\phi_f^+ - \phi_f^-} = 1 - \frac{\phi_f^+ - \phi_C}{\phi_f^+ - \phi_f^-} \quad . \tag{5.30}$$

After some transformation which is explained in detail in H. Jasak, Weller, and Gosman 1999 one gets the final equation

$$\tilde{\phi}_C = 1 - \frac{\phi_D - \phi_C}{2(\nabla\phi)_C \cdot \vec{CD}} \quad . \tag{5.31}$$

For both shown methods the extrapolation may lead to values outside the interval [0, 1]. Therefore the values will be bounded to the given interval once they have been calculated. The methods are satisfactory for continuous quantities. For the discontinuous volume fraction these approaches may lead to big errors and stability problems as both are based on the gradient.

Another way getting the upstream node U is based on a reconstruction. It is introduced as an extension of the BICS scheme in Wackers 2007. The resulting scheme is called *Blended Interface Capturing Scheme with Reconstruction* (BRICS). The difference to the original BICS scheme lies in the reconstruction of the upstream node U as explained below.

The BRICS scheme first searches the cell UU containing the imaginary point U, see Fig. 5.8. The search consist of a loop over the neighboring cells of C. The cells UU can be pre-computed. If the mesh is not changing the search algorithm has to be called once, only. After the cell UU is identified the value for the point U can be calculated with a weighted interpolation based on the cell UU and its neighboring cells UU_{nb}

$$\phi_U = \sum_{N=UU,UU_{\text{nb}}} \frac{\phi_N}{\|\vec{NU}\|} \Big/ \sum_{N=UU,UU_{\text{nb}}} \frac{1}{\|\vec{NU}\|} \quad . \tag{5.32}$$

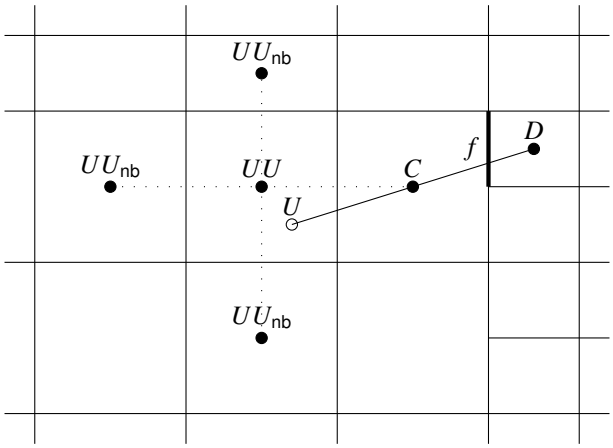


Figure 5.8.: Reconstruction of the upstream node U

All three schemes, HRIC, BICS and BRICS have been implemented and applied during this work. The quality of the results seems to be equally good. Due to its conclusive discontinuity treatment the BRICS scheme was finally chosen and all results presented in this work are based on this scheme.

5.3.7. Deferred Correction

The implementation of a high-resolution scheme is not straightforward. For example the far upwind cell U is not directly available on unstructured meshes. Usually unstructured solvers, like OpenFOAM, only allow to consider the contribution of the direct cell neighbors implicitly in the coefficient matrix. Hence the contribution of cell U can't be treated implicitly. Furthermore an explicit treatment of the convective term would destabilize the solution. Therefore, the high-resolution schemes are implemented as a *Deferred Correction* scheme (Rubin and Khosla 1977):

Terms with an approximation of higher order are treated explicitly and set to the *right hand side* (rhs). Additionally one adds the same term with a lower approximation on both sides of the equation system. This leads to following form for the VOF transport equation:

$$\frac{\delta\alpha}{\delta t} + \mathscr{L}(\alpha\mathbf{u}) = \mathscr{L}(\alpha\mathbf{u}) - \mathscr{H}(\alpha\mathbf{u}) \tag{5.33}$$

where $\mathscr{H}$ and $\mathscr{L}$ are the operators for the approximations of higher and lower order, respectively. Terms on the left hand side are treated implicitly, terms on the right hand side are treated explicitly. The difference on the rhs is small and should not produce additional difficulties (Milovan Perić and Ferziger 2002). On convergence both terms of lower order cancel out and the solution has the accuracy of the higher order scheme. This method requires more iterations but less memory space and less computational cost per iteration.

5.4. New Approach for the Suppression of Numerical Ventilation

The usage of the previously described high-resolution schemes already reduces the numerical ventilation. Yet, the numerical ventilation, where wrong air is convected under the hull, is high enough to

noticeably influence the shear stress, especially, if the stem of the ship is not piercing the free-surface (see Section 11.2). In Viola, Flay, and Ponzini 2012 an iterative approach to circumvent this problem is presented. It removes the wrong air by introducing an air-sink, which amount is iteratively increased. Here, a different approach will be introduced.

The new method first selects the underwater region at the ship and than implicitly relaxes the volume fraction to 1.0. The key point is the correct selection of the underwater region. Knowing that the pressure p is zero at the free-surface in calm water condition one can derive an estimation of the underwater region based on the pressure and the distance to the hull. The Relaxation is applied on cells, which fulfills following criteria:

(1) $p \geq 0.008 \rho_w g L_{\text{os}}$,

(2) $y \leq 0.003 L_{\text{os}}$ and

(3) $\alpha_w > 0.6$

with the pressure p, the water density ρ_w, the magnitude of the gravity acceleration g, the length over surface of the ship L_{os}, the wall distance y and the volume fraction of the water phase α_w. To prevent numerical instabilities due to oscillations a linear blending is introduced for the pressure dependent part. This leads to the final equation for the relaxation factor

$$r_{\text{suppr.}} := \begin{cases} 10^{-6} & \text{if } p \geq p_{\text{suppr.}}, y \leq y_{\text{suppr.}} \text{ and } \alpha_w > 0.6 \\ \max\left(1.0 - \frac{p - 0.1 p_{\text{suppr.}}}{0.9 p_{\text{suppr.}}}, 10^{-6}\right) & \text{if } p_{\text{suppr.}} > p \geq 0.1 p_{\text{suppr.}}, y \leq y_{\text{suppr.}} \text{ and } \alpha_w > 0.6 \\ 1.0 & \text{else} \end{cases} \tag{5.34}$$

with

$$p_{\text{suppr.}} = 0.008 \rho_w g L_{\text{os}} \tag{5.35}$$

and

$$y_{\text{suppr.}} = 0.003 L_{\text{os}} \quad . \tag{5.36}$$

The two factors 0.008 and 0.003 were set by experience and are used consequently in all simulations without any adjustments. The relaxation is based on

$$\frac{1}{r} a_{\text{VOF d}} \alpha_{\text{d}}^{q+1} + \sum_n a_{\text{VOF n}} \alpha_{\text{n}}^{q+1} = s_{\text{VOF}} + \frac{1 - r}{r} a_{\text{VOF d}} \alpha^{\text{t}} \tag{5.37}$$

with the diagonal elements $a_{\text{VOF d}}$ and neighbor elements $a_{\text{VOF n}}$ of the coefficient matrix of the Volume-of-Fluid transport equation. The term s_{VOF} contains all possible source terms or contributions to the right hand side. The target volume fraction α^{t} is 1.0 to represent a cell fully filled with water. The approach significantly reduces the numerical ventilation. Only in the region straight below the free-surface some ventilation is left.

Forcing the volume fraction to 1.0 can be interpreted as an artificial water source, which obviously defects the mass conservation. Nevertheless, the numerical ventilation is also nonphysical and its impact onto the result is much higher. For the simulations presented in Section 11.3 the here described

suppression method was applied and Figure 11.16 shows the resulting volume fraction on the bottom of the hull with very little numerical ventilation.

6. Turbulence Modeling

This chapter describes the used form of the k-ω-SST turbulence model. First the original version contributed with OpenFOAM is presented. Afterwards possible known modifications are discussed and the final, newly implemented version of the k-ω-SST turbulence model is presented.

6.1. OpenFOAMs k-ω-SST Turbulence Model

Although calculation power is growing exponentially it is still not affordable to use *Direct Numerical Simulations* (DNS) for industrial applications on turbulent flow. While the direct simulation of large eddies, *Large Eddy Simulation* (LES), has been used in few cases it is still common practice to completely average the turbulent fluctuations. The most applied methods are based on the *Reynolds-averaged Navier-Stokes Equations* (RANSE). Turbulence models based on averaging usually include empirical terms which are only valid under specific circumstances. Therefore choosing such model always means to find the best compromise for the given application. A good compromise for yacht simulations used for many industrial applications is the two-equation turbulence model *k-ω-SST* (Menter, Ferreira, and Esch 2003). This two-equations-model uses the *k-ϵ* (Launder and Sharma 1974) model for the far field region and blends to the *k-ω* (Wilcox 1988, Wilcox 2006 and Wilcox 2008) model in the near wall region. The equations of OpenFOAMs implementation of the k-ω-SST turbulence model can be written as follows

$$\frac{\partial k}{\partial t} + \nabla \cdot (\mathbf{u}k) - \nabla \cdot ((\nu_\ell + \alpha_k \nu_\mathrm{t}) \nabla k) = P_k - \beta^* k\omega \tag{6.1}$$

and

$$\begin{aligned} &\frac{\partial \omega}{\partial t} + \nabla \cdot (\mathbf{u}\omega) - \nabla \cdot ((\nu_\ell + \alpha_\omega \nu_\mathrm{t}) \nabla \omega) \\ &= \gamma S_2 - \beta \omega^2 + 2\,(1 - F_1)\,\frac{\alpha_{\omega 2}}{\omega}\,(\nabla k \cdot \nabla \omega) \end{aligned} \tag{6.2}$$

with

$$S_2 = \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) : \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) \tag{6.3}$$

where : stands for the double inner product, k for the turbulent kinetic energy and ω for the specific turbulent dissipation rate. The production rate for the kinetic energy (the production of turbulence due to shear) is

$$P_k = \min\left(G, 10.0\beta^*\omega\right) \tag{6.4}$$

with

$$G = \nu_\mathrm{t} S_2 \quad . \tag{6.5}$$

The closure coefficients are $\alpha_{\omega 1} = 0.5$, $\alpha_{\omega 2} = 0.856$, $\alpha_{k1} = 0.85$, $\alpha_{k2} = 1.0$, $\beta_1 = 0.075$, $\beta_2 = 0.0828$, $\gamma_1 = \frac{5}{9}$, $\gamma_2 = 0.44$, $\beta^* = 0.09$ and $a_1 = 0.31$. The values for α_ω, α_k, β and γ are than calculated with the following generalized blending function

$$\phi_{\text{blend}} = F_1 \phi_1 + (1 - F_1)\, \phi_2 \tag{6.6}$$

with the blending factor

$$F_1 = \tanh\left(\left(\min\left(\min\left(F^*, \frac{4\alpha_{\omega 2} k}{\text{CD}^+_{k\omega} y^2}\right), 10\right)\right)^4\right) \tag{6.7}$$

with the wall distance y,

$$F^* = \max\left(\frac{k^{0.5}}{\beta^* y \omega}, \frac{500\nu_l}{y^2 \omega}\right) \quad , \tag{6.8}$$

and

$$\text{CD}^+_{k\omega} = \max\left(\text{CD}_{k\omega}, 10^-10\right) \tag{6.9}$$

with

$$\text{CD}_{k\omega} = \frac{2\alpha_{\omega 2}}{\omega} \nabla k \cdot \nabla \omega \quad . \tag{6.10}$$

F_1 blends between the k-ϵ and k-ω models. The blending value F_2 is calculated by

$$F_2 = \tanh\left(\left(\min\left(F^*, 100\right)\right)^2\right) \quad . \tag{6.11}$$

Finally, the kinematic and the dynamic eddy viscosities are calculated by

$$\nu_t = \frac{a_1 k}{\max\left(a_1 \omega, F_2 S_2^{0.5}\right)} \tag{6.12}$$

and

$$\mu_t = \rho \nu_t \quad . \tag{6.13}$$

6.2. Applied Modifications

Using this Version of the k-ω-SST model leads to a few problems for free-surface simulations. These problems and possible solutions are described below:

(1) As explained in Chapter 2 it is necessary to include the density into the derivations as (2.4) is used to calculate the fluid properties. Otherwise the derivations are not consistent.

(2) As shown in Spalart and Rumsey 2007 and Eq. (6.3) to (6.5) the turbulent production depends on the velocity gradient, see (6.4). Therefore, using this model, a decay of the turbulence is inherently given in the free-stream region between the inlet and the object of interest. Equations for the rate of decay are given in Spalart and Rumsey 2007, showing that the turbulent viscosity decays much slower than the turbulent kinetic energy and the specific dissipation rate. Nevertheless it is recommended to reach the correct values at the object of interrest for all three variables, otherwise important separations might be undetected. In a few situations it is possbile to adjust the turbulent values at the inlet using

the decay equations and the desired values at the object. Indeed, the equations are not solvable for all kind of problems.

To remedy this problem it is suggested in Spalart and Rumsey 2007 and Rumsey and Spalart 2009 to use ambient values in the far field region by introducing constant sustaining terms $\beta^* \omega_{\text{amb}} k_{\text{amb}}$ and $k_{\text{amb}} \omega_{\text{amb}}^2$ into the turbulent governing equations. As explained in Rumsey and Spalart 2009 these terms are significantly smaller than the free-stream turbulence levels inside the boundary layer. The ambient values k_{amb} and ω_{amb} do not equal the values typically used at the inlet. In Rumsey and Spalart 2009 it is suggested to estimate these values with $k_{\text{amb}} = 10^{-6} u^2$ and $\omega_{\text{amb}} = 5.0 \frac{u}{l_{\text{chord}}}$, where l_{chord} is the chord length of the object.

The approach is already implemented in the newest versions of OpenFOAM and the method needs to be activated only.

(3a) Applying the model onto free-surface simulations with sea waves leads to an over-prediction of the turbulent kinetic energy and turbulent viscosity at the free-surface. This high amount of wrong turbulence leads to wrong wave damping reducing the wave height to a small fraction after a few wavelengths. For the first simulations and the simulations presented in Meyer, Graf, and Slawig 2017 and appendix B a modified production term described by Mayer and Madsen 2000 was used. In this modified version, the production term is based on the mean rotation rather than the strain rate by using $S_2 = (\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{u})$. This term has been used also by Jacobsen, Fuhrman, and Fredsøe 2012 and suppresses the wrong growth of the turbulence successfully. As stated out in Mayer and Madsen 2000 and Larsen and Fuhrman 2018 this approach has several fundamental deficiencies. Calculating the turbulent production based on the mean rotation has no valid physical background, thus this approach was only used as an interim solution within this work.

(3b) Another approach to avoid wrong wave damping is given by Devolder, Rauwoens, and Troch 2017. It uses a sink based on a buoyancy term $G_b = -\frac{\nu_t}{\sigma_t} \nabla \rho \cdot \mathbf{g}$ in the turbulent kinetic energy equation, which is based on the density gradient at the free-surface. The idea to use a buoyancy term is adopted from Maele and Merci 2006, where such an approach is successfully applied onto fire simulations. This is an important difference, as fire flows have a smooth density transition by nature and sharp free-surface flows do not. In Devolder, Rauwoens, and Troch 2017 it is argued to use this approach for free-surface simulations, as the density has a smooth gradient due to the smearing from the Volume-of-Fluid method. Nevertheless, this function seems to be unphysical at the free-surface. First, the free-surface will be smeared over 1 to 5 cells due to the VOF-method. This number of cells is nearly independent of the cell size. Therefore, for smaller cells, the density gradient increases and the buoyancy term increases too. Furthermore if the free-surface is located directly at the face between two cells, usually no smearing occurs. Due to this reason, the buoyancy term is highly grid dependent. A second point of criticism is, that the method suppresses the turbulence in all three spatial direction at the free-surface. It seems to be plausible to treat the free-surface as a slip-wall. As the free-surface represents a sharp interface, the turbulent vorticities should decrease at the free-surface, because they can not go through it. Nevertheless, this must not be an isotropic phenomenom and anistropic vorticities parallel to the free-surface might still occur. The buoyancy term has not been activated in this work, due to above reasons and it is only mentioned for the sake of completeness.

(3c) A new valid approach to avoid exponentially growing of turbulence in waves is given by Larsen and Fuhrman 2018. It is shown that almost all commonly used two-equation turbulence closure models are unconditionally unstable in regions of nearly potential flow with finite strain. Especially sea waves have potential flow characteristics. The solution is a new formulation of the closure, which limits the kinematic viscosity in such regions, but blends to the original formulation in all other regions, like uniform boundary layer flows.

Modifying the k-ω-SST model with above methods leads to the following version:

$$\frac{\partial \rho k}{\partial t} + \nabla \cdot (\rho \mathbf{u} k) - \nabla \cdot (\rho (\nu_\ell + \alpha_k \nu_\mathrm{t}) \nabla k) = \rho P_k + G_b - \rho \beta^* k \omega + \rho \beta^* \omega_\mathrm{amb} k_\mathrm{amb} \tag{6.14}$$

and

$$\begin{aligned} &\frac{\partial \rho \omega}{\partial t} + \nabla \cdot (\rho \mathbf{u} \omega) - \nabla \cdot (\rho (\nu_\ell + \alpha_\omega \nu_\mathrm{t}) \nabla \omega) \\ &= \rho \gamma S_2 - \rho \beta \omega^2 + 2 (1 - F_1) \rho \frac{\alpha_{\omega 2}}{\omega} (\nabla k \cdot \nabla \omega) + \rho \beta \omega_\mathrm{amb}^2 \end{aligned} \tag{6.15}$$

with the buoyancy modification

$$G_b = -\frac{\nu_\mathrm{t}}{\sigma_t} \nabla \rho \cdot \mathbf{g} \tag{6.16}$$

with a turbulent Prandtl number $\sigma_t = 0.85$. The last terms in (6.14) and (6.15) enforce the ambient values of k and ω in the far field region around the object. The modified k-ω closure which stabilizes the model in nearly potential flow regions leads to following equation for the turbulent kinematic viscosity

$$\nu_\mathrm{t} = \frac{a_1 k}{\max\left(a_1 \omega, F_2 S_2^{0.5}, a_1 \lambda_2 \frac{\beta}{\beta^* \gamma} \frac{S_2}{p_\Omega} \omega\right)} \tag{6.17}$$

with

$$p_\Omega = \left(\nabla \mathbf{u} - (\nabla \mathbf{u})^T\right) : \left(\nabla \mathbf{u} - (\nabla \mathbf{u})^T\right) \tag{6.18}$$

and the stress limiter coefficient $\lambda_2 = 0.05$, which defines the threshold of $\frac{p_\Omega}{S_2}$ identifying a region as effectively potential flow.

7. Rigid Body Motion

Ships driving through the water will change their floating position from the hydro-static to the hydro-dynamic position. The hydrodynamic floating position depends on the fluid forces and vice versa. Additionally in sea waves the whole floating position is unsteady. This makes it necessary to include the motion of the vessels into the numerical simulation. In Azcueta 2001 the complete integration of the ship motion into the RANS equations is derived.

After describing basic definitions like the frame of reference, translational and rotational motions, this chapter gives an overview of the equations of motion, mesh motion techniques and the newly implemented sail forces model. Furthermore the stability of the time marching procedure is discussed and a newly implemented stabilized version is explained. Finally, the new motion method is validated with experimental results and results from the commercial solver Star-CCM+.

7.1. Frame of Reference

Two coordinate systems (see Fig. 7.1) are used to describe the rigid body motion:

Global Coordinate System (GCS):
A non-rotating, non-accelerating, right-handed coordinate system, which moves with the mean velocity of the ship.

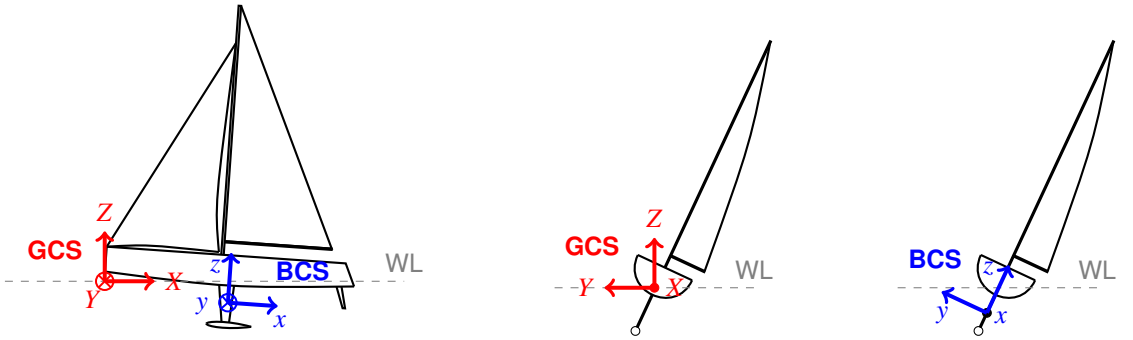
Body Coordinate System (BCS):
A body-fixed, right-handed coordinate system, which has its origin at the center of gravity **CoG** of the ship. The x-axis is pointing from bow to stern, the y-axis is pointing from port to starboard and the z-axis is pointing upwards.

7.2. Translational Motions

In ship hydrodynamics the translational motions of a vessel are subdivided into *heave*, *sway* and *surge* motions.

Surge:
The surge motion describes the transversal motion around the longitudinal axis (x-axis) of the vessel.

Sway:
The sway motion describes the transversal motion around the transverse axis (side to side, y-axis) of



(a) coordinate systems **(b)** Global coordinate system **(c)** Body coordinate system

Figure 7.1.: Coordinate systems

the vessel.

Heave:
The heave motion describes the transversal motion around the vertical axis (z-axis) of the vessel.

7.3. Rotational Motions

The rotational motion of a ship can be described by three angles. If the first and the third rotations are around the same axis, the three angles are called *Euler Angles*, if the rotations are around three different axis, they are called *Tait-Bryan Angles*. Here, the rotations are applied around three different axis using the three angles *roll*, *pitch* and *yaw*.

Roll Angle:
The roll angle ϕ_r is the rotation around the longitudinal axis of the vessel (x-axis). The deviation from normal on this axis is referred to as *heel*.

Pitch Angle:
The pitch angle θ_p is the rotation around the transverse axis (side to side, y-axis) of the vessel. The deviation from normal is called *trim*.

Yaw Angle:
The yaw angle ψ is the rotation around the vertical axis of the vessel (z-axis). With respect to the magnetic north pole, the deviation from normal is called *heading*.

Based on this angles the rotation tensors can be described as follows:
Rotation around inital x-axis:

$$\mathbf{T}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_\mathrm{r} & -\sin\phi_\mathrm{r} \\ 0 & \sin\phi_\mathrm{r} & \cos\phi_\mathrm{r} \end{bmatrix} \tag{7.1}$$

Rotation around initial y-axis:

$$\mathbf{T}_y = \begin{bmatrix} \cos\theta_\mathrm{p} & 0 & \sin\theta_\mathrm{p} \\ 0 & 1 & 0 \\ -\sin\theta_\mathrm{p} & 0 & \cos\theta_\mathrm{p} \end{bmatrix} \tag{7.2}$$

Rotation around initial z-axis:

$$\mathbf{T}_z = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.3}$$

As the three rotations are not commutable it is required to agree on a fixed order. A common order is the Roll-Pitch-Yaw convention, which is used here:

Roll-Pitch-Yaw convention in extrinsic order: $x - y - z$

Roll-Pitch-Yaw convention in intrinsic order: $z - y' - x''$

In case of the extrinsic order all three rotations are around the initial axis x, y and z. In case of the intrinsic order the rotations are around the rotated axis. This means, the first rotation around the initial z-axis will lead to a new right handed coordinate system with the axis x', y' and z'. The second rotation is around the new y'-axis and leads to another coordinate system with the axis x'', y'' and z''. The last rotation is around the new x''-axis. Both, exstrinsic and instinsic order, will lead to the same final rotation of the vessel. The roll-pitch-yaw rotations lead to a transformation tensor which allows to transform from the body-system to the global system

$$\begin{aligned} \mathbf{T}_{zyx} &= \mathbf{T}_z \mathbf{T}_y \mathbf{T}_x \\ &= \begin{bmatrix} \cos\theta_\mathrm{p}\cos\psi & \sin\phi_\mathrm{r}\sin\theta_\mathrm{p}\cos\psi - \cos\phi_\mathrm{r}\sin\psi & \cos\phi_\mathrm{r}\sin\theta_\mathrm{p}\cos\psi + \sin\phi_\mathrm{r}\sin\psi \\ \cos\theta_\mathrm{p}\sin\psi & \sin\phi_\mathrm{r}\sin\theta_\mathrm{p}\sin\psi + \cos\phi_\mathrm{r}\cos\psi & \cos\phi_\mathrm{r}\sin\theta_\mathrm{p}\sin\psi - \sin\phi_\mathrm{r}\cos\psi \\ -\sin\theta_\mathrm{p} & \sin\phi_\mathrm{r}\, cos\theta_\mathrm{p} & \cos\phi_\mathrm{r}\cos\theta_\mathrm{p} \end{bmatrix}. \end{aligned} \tag{7.4}$$

The transposed tensor $\mathbf{T}_{zyx}^T$ is used to transform from the global system into the body-system.

7.4. Motion Equations

The equation for translational motion can be written as:

$$\mathbf{F}_\mathrm{GCS} = \frac{d}{dt}\left(m\mathbf{v}_\mathrm{GCS}\right) \tag{7.5}$$

with the mass of the ship m, the velocity of the ship $\mathbf{v}_\mathrm{GCS}$ in the global coordinate system and the total external forces $\mathbf{F}_\mathrm{GCS}$ also given in the global coordinate system. The equation for rotational motion can be written as:

$$\mathbf{M}_\mathrm{BCS}^\mathbf{CoG} = \frac{d}{dt}\left(\mathbf{I}_\mathrm{BCS}^\mathbf{CoG}\boldsymbol{\Omega}_\mathrm{BCS}^\mathbf{CoG}\right) \tag{7.6}$$

with the inertia tensor of the body with respect to the center of gravity $\mathbf{I}_{\text{BCS}}$, the angular velocity of the ship $\mathbf{\Omega}_{\text{BCS}}$ and the total external moment $\mathbf{M}_{\text{BCS}}$. All values are given in the body coordinate system with respect to the center of gravity.
Equation (7.6) is converted into the global coordinate system using the previously defined transformation tensor. Under the assumption of a constant inertia tensor in the BCS, the equation for rotational motion can than be written as:

$$\mathbf{T}_{zyx}^{-1}\mathbf{M}_{\text{GCS}}^{\text{CoG}} = \mathbf{I}_{\text{BCS}}^{\text{CoG}}\frac{d}{dt}\left(\mathbf{T}_{zyx}^{-1}\mathbf{\Omega}_{\text{GCS}}^{\text{CoG}}\right) \tag{7.7}$$

After following transformation steps

$$\mathbf{T}_{zyx}^{-1}\mathbf{M}_{\text{GCS}}^{\text{CoG}} = \mathbf{I}_{\text{BCS}}^{\text{CoG}}\left(\mathbf{T}_{zyx}^{-1}\frac{d}{dt}\left(\mathbf{\Omega}_{\text{GCS}}^{\text{CoG}}\right) + \frac{d}{dt}\left(\mathbf{T}_{zyx}^{-1}\right)\mathbf{\Omega}_{\text{GCS}}^{\text{CoG}}\right) \tag{7.8}$$

and

$$\mathbf{T}_{zyx}^{-1}\mathbf{M}_{\text{GCS}}^{\text{CoG}} = \mathbf{I}_{\text{BCS}}^{\text{CoG}}\left(\mathbf{T}_{zyx}^{-1}\frac{d}{dt}\left(\mathbf{\Omega}_{\text{GCS}}^{\text{CoG}}\right) + \mathbf{\Omega}_{\text{GCS}}^{\text{CoG}} \times \mathbf{T}_{zyx}^{-1}\mathbf{\Omega}_{\text{GCS}}^{\text{CoG}}\right) \tag{7.9}$$

one gets the final form for the equation of rotational motion

$$\mathbf{M}_{\text{GCS}}^{\text{CoG}} = \mathbf{T}_{zyx}\mathbf{I}_{\text{BCS}}^{\text{CoG}}\mathbf{T}_{zyx}^{-1}\frac{d}{dt}\left(\mathbf{\Omega}_{\text{GCS}}^{\text{CoG}}\right) + \mathbf{\Omega}_{\text{GCS}}^{\text{CoG}} \times \mathbf{T}_{zyx}\mathbf{I}_{\text{BCS}}^{\text{CoG}}\mathbf{T}_{zyx}^{-1}\mathbf{\Omega}_{\text{GCS}}^{\text{CoG}} \quad . \tag{7.10}$$

The inertia tensor is

$$\mathbf{I}_{\text{BCS}}^{\text{CoG}} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \tag{7.11}$$

with the moments of inertia

$$I_{xx} = \int_m \left(y^2 + z^2\right) dm; \quad I_{yy} = \int_m \left(x^2 + z^2\right) dm; \quad I_{zz} = \int_m \left(x^2 + y^2\right) dm \tag{7.12}$$

and the moments of deviation

$$I_{xy} = I_{yx} = -\int_m (x + y)\, dm; \quad I_{xz} = I_{zx} = -\int_m (x + z)\, dm; \quad I_{yz} = I_{zy} = -\int_m (y + z)\, dm \quad . \tag{7.13}$$

7.5. Movement of the Body

The movement of the body can be achieved with several different approaches.

Grid Motion:
This methods moves the entire grid according to the motion of the object. The grid itself is not manipulated and keeps its mesh quality. For free-surface simulations the user has to ensure, that the free-surface always lies within the dedicated refinement-zone. For large motion amplitudes or especially angular motions this requirement might lead to large cell numbers.

Mesh Morphing:
A mesh morphing method deforms the mesh to enable the motion of the object. The complete amount

of motion is applied to a few cells around the object. With increasing distance to the object the amount deformation is decreased. For free-surface simulations it is only necessary to add additional cells in the region next to the object. Hence the cell-size can be kept small. On the other hand, the deformation of the mesh might reduce the cell-quality (e.g. non-orthogonality, skewness).

Sliding Interface:
For the sliding interface method a rotating grid around the object (e.g. propulsor, ship) has to be defined. For angular motions this sub-grid is rotated. Translational movements are applied to both grids. Special interpolation techniques have to be applied for the interface between the main grid and the sub-grid. Especially for pitch rotations the sub-grid has to be quite large, as it needs to be larger then the ship length.

Overset Grid:
The overset grid method uses two grids. one background grid which contains the refinement for the free-surface and one overlapping grid, which contains the object and its near field refinements. This technique allows arbitrary and large ship motions.

All methods have in common that the whole mesh or at least some cells are moved. This introduces an additional velocity field $\mathbf{u}_\mathrm{b}$ which represents the local grid velocities. The grid velocity has to be considered in the convective term of all governing equations. For the momentum conservation equation (2.1) this leads to

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} (\mathbf{u} - \mathbf{u}_\mathrm{b})) - \nabla \cdot \mu_\mathrm{e} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) = -\nabla p + \rho \mathbf{g} \quad . \tag{7.14}$$

In Böhm 2014 the roughly described methods have been investigated with respect to their application on free-surface simulations with sailing yachts. Here, the mesh morphing method of OpenFOAM is used.

7.6. External Forces and Moments

The external forces and moments acting on the object can be separated in fluid, gravity and other forces

$$\mathbf{F}_\mathrm{GCS} = \mathbf{F}_\mathrm{fluid} + \mathbf{F}_\mathrm{gravity} + \mathbf{F}_\mathrm{other} \quad . \tag{7.15}$$

7.6.1. Forces and Moments Induced by the Fluid and Gravity

The fluid forces consist of pressure and viscous parts

$$\mathbf{F}_\mathrm{fluid} = \mathbf{F}_\mathrm{pressure} + \mathbf{F}_\mathrm{viscous} \quad . \tag{7.16}$$

Both can be calculated by integrating over the surface of the body

$$\mathbf{F}_\mathrm{pressure} = \int_s -p \mathbf{n} ds = \sum_{j=1}^{n} -p_j \mathbf{s}_j \quad , \tag{7.17}$$

$$\mathbf{F}_{\text{viscous}} = \int_s \tau_j \cdot \mathbf{n} ds = \sum_{j=1}^{n} \tau_j \cdot \mathbf{s}_j \tag{7.18}$$

with the viscous stress tensor

$$\tau = \mu_e \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \quad . \tag{7.19}$$

The gravity force is

$$\mathbf{F}_{\text{gravity}} = m\mathbf{g} \quad . \tag{7.20}$$

The same approach leads to Eq. (7.21) to (7.24) for the external moments. As the point of reference is set to the center of gravity, the moment of the gravity force is zero.

$$\mathbf{M}_{\text{GCS}}^{\text{CoG}} = \mathbf{M}_{\text{fluid}} + \underbrace{\mathbf{M}_{\text{gravity}}}_{0} + \mathbf{M}_{\text{other}} \tag{7.21}$$

$$\mathbf{M}_{\text{fluid}} = \mathbf{M}_{\text{pressure}} + \mathbf{M}_{\text{viscous}} \tag{7.22}$$

$$\mathbf{M}_{\text{pressure}} = \int_s -\left(\mathbf{x} - \mathbf{x}_g\right) \times p\mathbf{n} ds = \sum_{j=1}^{n} -\left(\mathbf{x}_i - \mathbf{x}_g\right) \times p_j \mathbf{s}_j \tag{7.23}$$

$$\mathbf{M}_{\text{viscous}} = \int_s \left(\mathbf{x} - \mathbf{x}_g\right) \times \tau_j \cdot \mathbf{n} ds = \sum_{j=1}^{n} \left(\mathbf{x}_i - \mathbf{x}_g\right) \times \tau_j \cdot \mathbf{s}_j \tag{7.24}$$

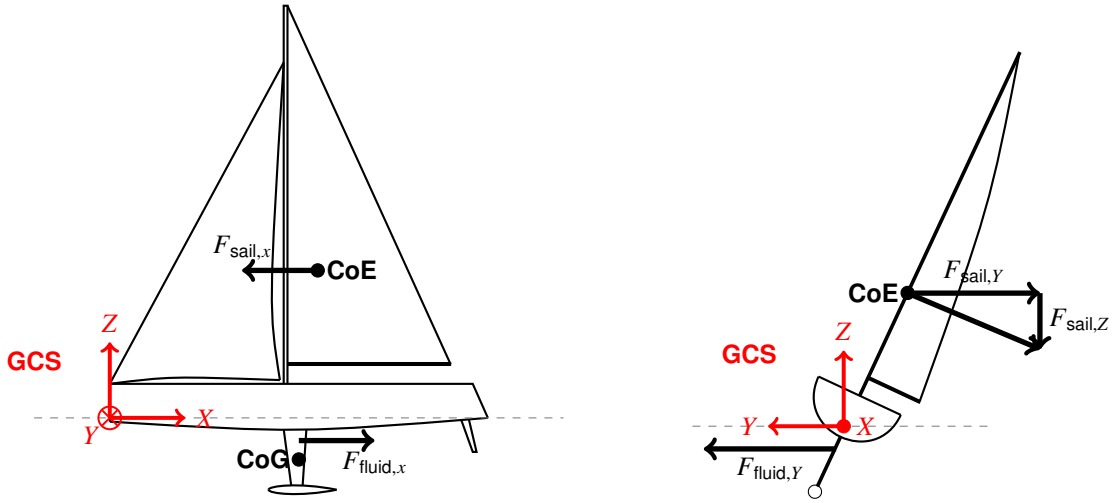
7.6.2. Forces and Moments Induced by the Sail

The *other forces*, F_{other}, may result from arbitrary sources. In the case of sailing yachts design it is common practice to consider the forces resulting from the sail as these forces influence the orientation of the hull. For hydrodynamic simulations, it is common practice to simulate the hull without a geometric representation of the sails. Therefore, the sail forces and moments do not result from the fluid simulation and a proper estimation of the sail forces has to be applied. Typical simulations of sailing yachts are free in pitch and heave, while all other degrees of freedoms are blocked. Hence, only the forces and moments of the sail contributing to the heave and pitch motions have to be considered.

The pitching moment of the sail results from the X-component of the driving force, which has to equal the negative hydrodynamic drag (F_{fluid}) of the yacht.

$$\mathbf{M}_{\text{sail}} = -\mathbf{r}_{\text{CoE}} \times (\mathbf{T}_{x1} \cdot F_{\text{fluid}}) \tag{7.25}$$

with $\mathbf{r}_{\text{CoE}}$ being the the distance vector from the center of gravity to the center of effort of the sail force and $\mathbf{T}_{x1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ being a tensor which extracts the x-component of the force. The simulations are run with a predefined drift angle. To make sure that the boat has its equilibrium condition at this drift angle the Y-component of the sail-force has to match the Y-component of the fluid-force. Assuming that the total force of the sail is perpendicular to the boats z-axis, one can derive the vertical contribution of



(a) sail moment **(b)** vertical sail force

Figure 7.2.: Sail forces

the sail-force as shown in Fig. 7.2 with

$$F_{\text{sail},z} = -f_{z,\text{sail}} F_{\text{fluid},y} * \tan(\phi_r) \tag{7.26}$$

where $f_{z,\text{sail}}$ is an adjustment parameter, which allows to manipulate the vertical contribution by the user.

7.7. Integration and Improvements onto the Stability

The rigid body motion equations and the flow equations are solved in a segregated way. To calculate the motion, the Eq. (7.5) and (7.10) have to be integrated in time. As described in Chapter 4 implicit methods like backward Euler, Crank-Nicolson or BDF(2) might be a preferred choice, due to the given A-stability. Indeed, such methods can not be implemented implicitly, as the forces and moments depend on the flow-equations.

Especially, simulations with light sailing yachts show convergence issues, due to this problem. Decreasing the time step or using a relaxation factor may help, but such approaches increase the simulation time and depend highly on a trial and error investment by the user. In Shigunov, Söding, and Zhou 2001 and Söding 2001 a reliable method based on an added mass is given, which will be described in the following.

For the new time step

$$M_{ij}\dot{v}_j = f_i \tag{7.27}$$

is valid. Where M_{ij} is the mass matrix, $\dot{v}_j$ holds the translational and rotational accelerations (with $i = 1...6$) and f_i holds the exact forces and moments of the current time step. Instead of the exact force only an approximation $\bar{f_i}$ is given. It is now possible to describe the exact force with its approximation

and an additional correction term

$$f_i = \bar{f}_i - A_{ij}\dot{v}_j \tag{7.28}$$

where A_{ij} is the unknown added mass matrix. As shown later in 8.1 the flow and the resulting fluid force f_i is determined several times at each time step.
In Shigunov, Söding, and Zhou 2001 an exact, but slow approach for the calculation of A_{ij} is given, in Söding 2001 a faster, approximate solution is added. Both ways are explained in the following.

Exact determination of the added mass (Shigunov, Söding, and Zhou 2001):
For the determination of the exact added mass, the acceleration of the object is set to zero for the first iteration at each time step. An additional iteration is done for each degree of freedom, where $\dot{v}_j = \delta_{jk}$ is used, with

$$k = 1...\text{number of degrees of freedom} \quad . \tag{7.29}$$

Subtracting $[f_i]_k$ from $[f_i]_0$ allows to calculate each component of the added mass matrix

$$[f_i]_0 - [f_i]_k = A_{ij}\left(\delta_{jk} - 0\right) = A_{ik} \quad . \tag{7.30}$$

Approximation of the added mass (Söding 2001):
This approach also sets the acceleration of the object to zero within the first iteration at the beginning of each time step. For the second iteration the acceleration is set to the solution of the last time step for all components. Again, based on the difference of the first two iterations one can calculate the added mass

$$[f_i]_0 - [f_i]_k = A_{ij}\dot{v}_{jk} \quad . \tag{7.31}$$

From the third iteration onward the acceleration is set to the solution of the previous iteration. Considering the old iterations into the determination of the added mass one gets an over-determined equation system for A_{ij}, which is solved with the weighted least squares method leading to

$$\sum_k \left(g\dot{v}_l\dot{v}_j\right)_k A^T_{ji} = -\sum_k (g\dot{v}_l\Delta f_i)_k \tag{7.32}$$

with $\Delta f_i = f_{ik} - f_{j0}$ and the weight factor g_k. As described by Söding 2001 it proved to appropriate to use weights g_k according to the relation $\frac{g_{k-1}}{g_k} = 0.9$ leading to the final equation for the added mass

$$\sum_{k=1}^{K} \left(0.9^{K-k}\dot{v}_l\dot{v}_j\right)_k A^T_{ji} = -\sum_{k}^{K} \left(0.9^{K-k}\dot{v}_l\Delta f_i\right)_k \quad . \tag{7.33}$$

In the given work, the second way has been implemented. Since this implementation, no instabilities due to the motion appear proving that the approximate way is stable enough.

7.8. Validation

The implementation of the added mass method has been validated with an in-house experiment made at the *Yacht Research Unit Kiel* presented by Singh 2012. The experiment uses a blunt cone vertically falling into water. The trajectory of the cone was measured and was originally compared to the CFD

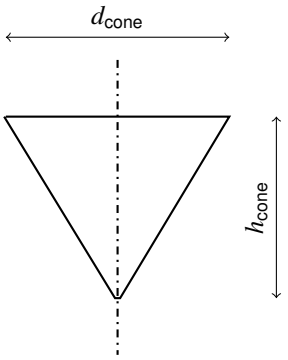


Figure 7.3.: Blunt Cone

Table 7.1.: Blunt Cone measurements

parameter	value
h_{cone}	0.160m
d_{cone}	0.197m
$d_{\text{cone-tip}}$	0.0045m
m	0.6148kg

results of Star-CCM+.
Figure 7.3 shows the shape of the cone and its measurements are given in Tab. 7.1. For the initial position of the cone the tip is placed directly at the free surface. For the experimental investigation a string going through the body keeps the cone in position and allows only a vertical movement.

The domain has a length and width of 1.7m and a height of 1.2m. The water has a depth of 0.65m. The coarsest cells have an edge length of 0.05m. The grid is refined at the free surface and next to the cone, the finest cells have a size of 0.0625m. A slip boundary condition is applied at the domain boundaries. The timestep is set to 0.0003s.

Figure 7.4 shows the results of the experiment, the Star-CCM+ simulation and the here described new OpenFOAM solver. All three results are in a very good alignment, which shows that the new stabilized implementation of the motion solver can be used for such unsteady problems.

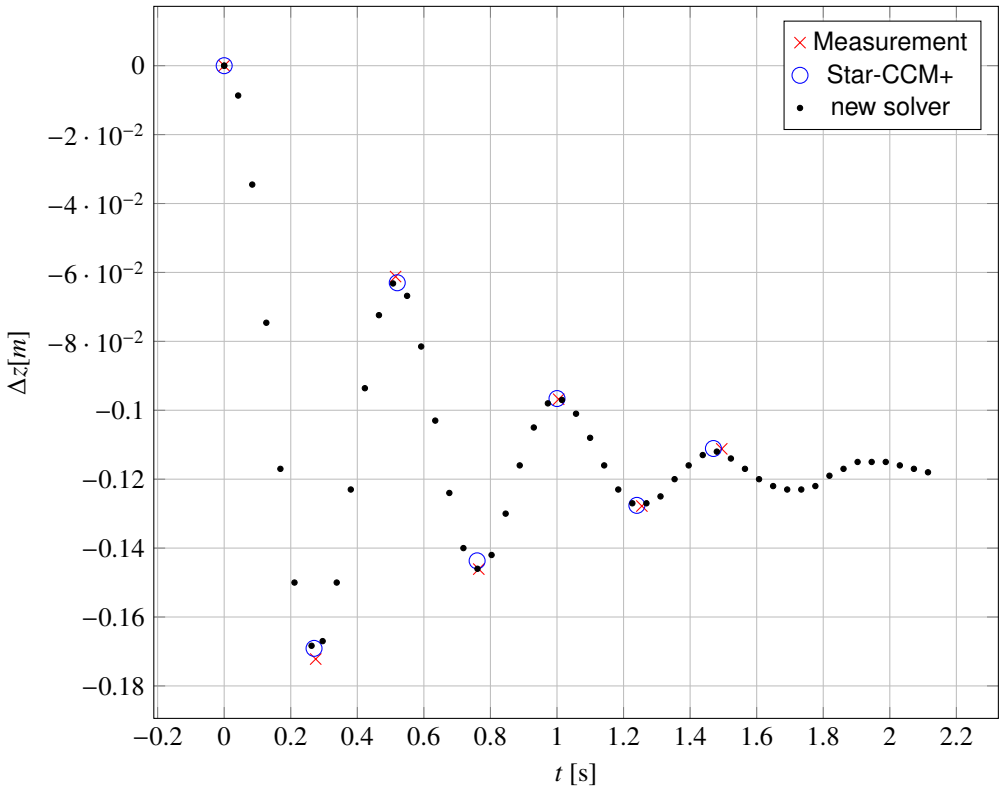


Figure 7.4.: Comparison of the experimental and numerical rigid body motion results

8. Solution Procedure

This chapter describes the solution method for the complete equation system. The system is solved using a segregated solution algorithm. First the derivation of the pressure- and velocity equations is given. Finally the iterative solution algorithm is given.

8.1. Derivation of the Pressure- and Velocity-Equations

To solve the coupled equation system, consisting of the equations (2.1), (2.2), (2.3), (6.14), (6.15) , (7.5) and (7.10) a segregated solution algorithm very similar to the algorithms *Pressure-Implicit with Splitting of Operators* (PISO) (Issa 1986) and *Semi-Implicit Method for Pressure-Linked Equations* (SIMPLE) (Patankar 1981) is used.

Using the implicit Euler discretization scheme exemplary, the semi-discretized, linearized momentum equation yields

$$\begin{aligned} &\frac{\rho}{\Delta t}\mathbf{u}_d + \frac{1}{V}a_d\mathbf{u}_d + \frac{1}{V}\sum_n a_n\mathbf{u}_n \\ &= \mathbf{q} + \rho\mathbf{g} - \mathscr{R}((\nabla p)_f \cdot \mathbf{s}_f) + \frac{\rho}{\Delta t}\hat{\mathbf{u}}_d \end{aligned} \tag{8.1}$$

Where a_d is the matrix coefficient of the main diagonal for cell c and a_n are the respective off diagonal elements resulting from the neighbor cells of the implicit part of the convective and diffusive terms. The subscript f indicates a value at the face. The time discretization needs the velocity of the old time step $\hat{\mathbf{u}}_d$. The source term $\mathbf{q}$ contains all other explicit parts. For the *Rhie-Chow* interpolation (Rhie and Chow 1982) generally used in OpenFOAM, the pressure gradient is calculated at the cell-faces. Then, the *Reconstruct-Volume-Field-Operator* $\mathscr{R}()$ reconstructs the required volume field from the face flux field, where $\mathbf{s}_f$ is the face area vector

$$\mathscr{R}(\phi_f) = \left(\sum_f \frac{\mathbf{s}_f}{|\mathbf{s}_f|}\mathbf{s}_f\right)^{-1} \cdot \left(\sum_f \frac{\mathbf{s}_f}{|\mathbf{s}_f|}\phi_f\right) \quad . \tag{8.2}$$

Rearranging (8.1) to $\mathbf{u}_d$ yields the velocity equation

$$\begin{aligned} \mathbf{u}_d = &\left(\frac{\rho}{\Delta t} + \frac{1}{V}a_d\right)^{-1} \\ &\left(\mathbf{q} + \rho\mathbf{g} - \mathscr{R}((\nabla p)_f \cdot \mathbf{s}_f) + \frac{\rho}{\Delta t}\hat{\mathbf{u}}_d - \frac{1}{V}\sum_n a_n\mathbf{u}_n\right) \end{aligned} \quad . \tag{8.3}$$

Substituting (8.3) into the continuity equation (2.2) and rearranging yields a Poisson equation for the pressure

$$\nabla \cdot \left(\frac{1}{\alpha_f}\nabla p\right) = \nabla \cdot \left(\frac{1}{\alpha_f}\left(h_{tf} + h_{sf}\right)\right) \tag{8.4}$$

with

$$\alpha_f = \left(\frac{\mathscr{I}(\rho)}{\Delta t} + \mathscr{I}\left(\frac{1}{V} a_d \right) \right) \quad , \tag{8.5}$$

$$h_{tf} = -\frac{\mathscr{I}(\rho)}{\Delta t} \mathscr{I}(\hat{\mathbf{u}}_d) \quad , \tag{8.6}$$

$$h_{sf} = -\mathscr{I}(\rho)\mathbf{g} + \mathscr{I}\left(\frac{1}{V} \sum_n a_n \mathbf{u}_n - \mathbf{q} \right) \tag{8.7}$$

and the interpolation operator $\mathscr{I}()$ which interpolates a face field from a volume field using a central differencing interpolation scheme. To avoid a steady state solution depending on the timestep Δt the term for the time is excluded from the interpolation and built on the face directly (Patrick Queutey and Michel Visonneau 2007). Therefore each term has to be interpolated on its own. This is not done in the original interFoam solver, where all terms are interpolated mutually. After solving the pressure equation the fluxes can be updated with

$$\Phi = \frac{\mathbf{s}_f}{\alpha_f} \cdot \left(h_{tf} + h_{sf} - (\nabla p)_f \right) \tag{8.8}$$

8.2. Iterative Solution Procedure

The equations are solved using a segregated algorithm which key points are shown in Fig. 8.1. Unless otherwise described, the momentum equation is relaxed implicitly with a relaxation factor of 0.7 and the pressure is relaxed explicitly with a relaxation factor of 0.3. Consequently the choice of the relaxation factors is in accordance to Milovan Perić 1985 who pointed out that optimal convergence is achieved, if the sum of both relaxation factors is about unity. The pressure relaxation is done after updating the flux and before updating the velocity. The shown non-orthogonal correction loop is used to update the explicitly treated part of (3.21). In the early development phase of this solver one additional iteration for the non-orthogonal correction was used. Later, test simulation showed, that additional loops are not required for steady-state simulations.

The individual equations itself are solved using OpenFOAMs standard linear equation solvers. Table 8.1 gives an overview of the selected solvers. The selected settings should guarantee a fast solution speed while giving the highest possible stability on complex grids with some bad cells.

For the Volume-of-Fluid equation and the pressure equation the *Geometric Agglomerated Algebraic Multigrid* (GAMG) solver is applied, which is a solver using a multigrid accelerator. The coarsest level is solved using a *preconditioned conjugate gradient solver* for symmetric matrices or using a *preconditioned bi-conjugate gradient stabilized solver* for asymmetric matrices. Furthermore for the VOF-equation the *DILUGaussSeidel* smoother is applied. According to the OpenFOAM manual this is a combined DILU/GaussSeidel smoother for asymmetric matrices in which DILU smoothing is followed by GaussSeidel smoothing to ensure that any "spikes" created by the DILU sweeps are smoothed-out. The *DILU* smoother is a simplified diagonal-based incomplete LU smoother for asymmetric matrices. For the Pressure equation the *DICGaussSeidel* smoother is applied, which is a Combined DIC/GaussSeidel smoother for symmetric matrices in which DIC smoothing is followed by GaussSeidel to ensure that any "spikes" created by the DIC sweeps are smoothed-out. The *DIC* smoother is a

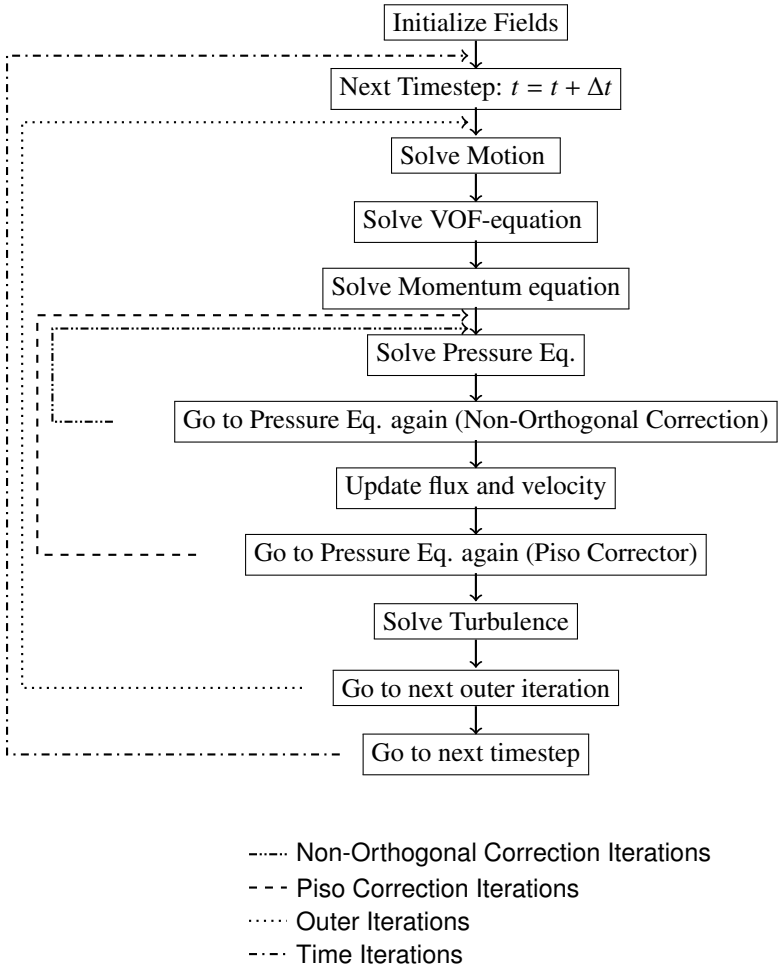


Figure 8.1.: Solution Algorithm

Table 8.1.: Linear Equation Solvers

Equation	Solver	Smoother	Absolute Tolerance	Relative Tolerance
VOF equation	GAMG	DILUGaussSeidel	1e-40	0.01
Momentum equation	smoothSolver	DILUGaussSeidel	1e-7	0.1
Pressure equation	GAMG	DICGaussSeidel	1e-7	0.1
Turbulence equations	smoothSolver	DILUGaussSeidel	1e-20	0.1

simplified diagonal-based incomplete Cholesky smoother for symmetric matrices. The momentum and the two turbulence equations are solved using OpenFOAMs *smoothSolver*. This is an iterative solver for symmetric and asymmetric matrices which uses an user defined smoother. A further mathematical description of these linear solvers is not given in the manual of OpenFOAM. Hence, the solvers are treated as “black-box” solvers in this work.

All linear solvers are based on an iterative procedure. The iterative process is stopped if at least one stopping criteria is fulfilled: (1) The absolute residual is smaller than the user defined absolute tolerance value. (2) The relative residual is smaller than the user defined relative tolerance value, where the relative residual is the difference of the absolute residual before and after solving the equation system.

9. Discontinuity Reconstruction

This chapter gives the explanation of the numerical problem resulting from the sharp interface between the water and air phase, followed by the description of the newly implemented discontinuity treatment solving this problem.

9.1. Problem Description

As mentioned in previous chapters, the Finite Volume method uses the general Gauss theorem to convert the volume integrals into surface integrals. This method is only valid for continuous variables, which is not the case at the free-surface for the density, viscosity and the density dependent pressure. Still, the unmodified Gauss theorem is applied for such variables in the OpenFOAM solver interFoam. This leads to major stability problems resulting in nonphysical high velocities. Values like $10,000\frac{m}{s}$ were observed, before the simulation aborted. In interFoam this problem is reduced noticeable due to the insufficient Volume-of-Fluid solution approach (MULES), which smears the volume fraction and density consequently leading to a smoother slope of the density at the free-surface.

With the new improved solution method for the Volume-of-Fluid equation less density smearing appears and the described problem becomes significant. Finally, the problem leads to arbitrary unstable simulations, making the solver not applicable in industrial practice.

One solution is to limit the velocities to a user specified value every time it is calculated. This leads to a more stable solver but still does not address the root cause. Hence, the reconstruction method presented in Patrick Queutey and Michel Visonneau 2007 has been implemented to remedy this problem completely. In Patrick Queutey and Michel Visonneau 2007 the method is also used to stabilize a free-surface simulation code (ISIS-CFD) computing off-shore flows or wave-resistance problems (P. Queutey, M. Visonneau, and Ferrant 2004).

9.2. Applied Reconstruction Method

The reconstruction is explained in detail in Patrick Queutey and Michel Visonneau 2007 and only the important parts are described here. The method assumes that the free surface Γ is exactly at the face f. It requires a known function, which is continuous on each subdomain Ω^+ and Ω^- and discontinuous across the interface. Here, it is assumed, that the density fulfills this requirement, as a high-resolution scheme is used for the discretization in the Volume-of-Fluid equation. With respect to Fig. 2.1 it becomes clear, that reconstructing $\left(\frac{\nabla p \cdot \mathbf{n}}{\rho}\right)_f$ instead of $(\mathbf{n} \cdot \nabla p)_f$ leads to a smooth solution. The complete geometrical derivation for unstructured grids is given in Patrick Queutey and Michel

Visonneau 2007. The presented approach leads to

$$\left(\frac{\nabla p \cdot \mathbf{n}}{\rho}\right)_f = \frac{1}{\hat{\rho}} \frac{p_R - p_L}{h} \boxed{+ \frac{\nabla p_L \cdot \mathbf{e}^- + \nabla p_R \cdot \mathbf{e}^+}{\hat{\rho} h}} \tag{9.1}$$

for the surface normal gradient with the explicit distance vectors $\mathbf{e}^+$ and $\mathbf{e}^-$ as shown in Fig. 9.1 for the correction of grid non-orthogonality. The boxed term is treated explicitly. The equation given for the face value is

$$\begin{aligned} p_f &= \frac{h^+\rho^+ p_L + h^-\rho^- p_R}{h^+\rho^+ + h^-\rho^-} \\ &\boxed{+ \frac{\rho^+\rho^-}{\hat{\rho}} \left(\frac{h^-\mathbf{e}^+ - h^+\mathbf{e}^-}{h}\right) \cdot \left(\frac{h^+}{h}\left(\frac{\nabla p}{\rho}\right)_L + \frac{h^-}{h}\left(\frac{\nabla p}{\rho}\right)_R\right)} \end{aligned} \tag{9.2}$$

with the face value $\hat{\rho}$ calculated by a reversed linear interpolation

$$\hat{\rho} = \frac{h^-}{h}\rho^- + \frac{h^+}{h}\rho^+ \quad . \tag{9.3}$$

Again, the boxed term is treated explicitly. In Patrick Queutey and Michel Visonneau 2007 it is not described how the explicit, boxed part of (9.1) is calculated. The pressure gradients can be calculated using the Gauss Theorem $\int_v \nabla\phi dv = \int_s \phi d\mathbf{s}$. The Gauss Theorem requires the face values. Using the linear interpolated face values is possible, but numerical tests showed a small negative influence on to the result. Therefore we use the reconstructed face values calculated with (9.2). Indeed, Equation (9.2) itself requires the cell gradients (calculated with the Gauss Theorem). Thus, it is a coupled system of equations and the implementation is not straightforward. Here it is implemented with the following iterative approach:

1. Calculate p_f^n using ∇p^{n-1}
2. Calcuate ∇p^n using p_f^n
3. Limit ∇p^n
4. Update iterator $n = n + 1$. Go to step 1

The limitation at Step 3 is essential for convergence and is based on OpenFOAMs *cellLimited*[1] gradient scheme. Unfortunately this limitation has a significant influence on the computational time. Numerical tests with different numbers of iterations were done during this work and showed that one iteration only is enough for a steady state result.

The final implementation consist of only one iteration and calculates the reconstructed face value with the old pressure gradient. Since there are outer iterations of the SIMPLE-algorithm there is still an

[1] The *cellLimited* scheme works as follows:
For each cell:

- Calculate the unlimited cell centered gradient
- For each face of the cell:
  - Extrapolate the cell centered value to the face by using the previously calculated gradient
  - Compare the extrapolated face value with the cell centered value of the neighbor cell also sharing this face.
  - If the extrapolated face value is not lying between the value of the neighbor cell or the current cell of interest: Limit the gradient, so that the extrapolated value lies inside this interval.

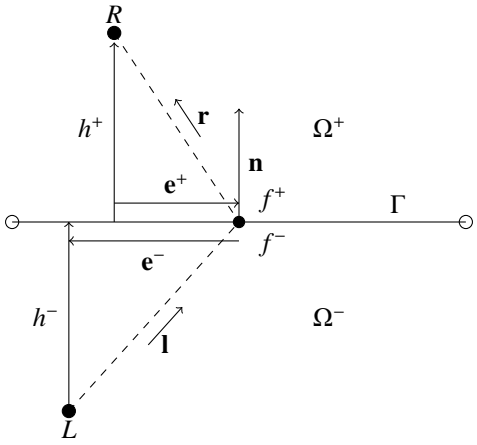


Figure 9.1.: Notations for the discontinuity reconstruction

iterative process updating the face value and cell gradient. It should be mentioned that a method without this gradient reconstruction also requires a limitation of the pressure gradient for sufficient stability. For the reconstruction the flow equations presented in Chapter 8 have to be modified as follows (modifications are highlighted in red):

Momentum equation:

$$\begin{aligned} &\frac{\rho}{\Delta t}\mathbf{u}_d + \frac{1}{V}a_d\mathbf{u}_d + \frac{1}{V}\sum_n a_n\mathbf{u}_n \\ &= \mathbf{q} + \rho\mathbf{g} - \rho\mathscr{R}((\nabla p)_f \cdot \mathbf{s}_f/\hat{\rho}) + \frac{\rho}{\Delta t}\hat{\mathbf{u}}_d \end{aligned} \tag{9.4}$$

Pressure equation:

$$\nabla\cdot\left(\frac{\mathscr{I}(\rho)}{\hat{\rho}}\alpha_f^{-1}\nabla p\right) = \nabla\cdot\left(\alpha_f^{-1}\left(h_{tf}+h_{sf}\right)\right) \tag{9.5}$$

Velocity equation:

$$\begin{aligned} &\mathbf{u}_d = \left(\frac{\rho}{\Delta t} + \frac{1}{V}a_d\right)^{-1} \\ &\left(\mathbf{q} + \rho\mathbf{g} - \rho\mathscr{R}((\nabla p)_f \cdot \mathbf{s}_f/\hat{\rho}) + \frac{\rho}{\Delta t}\hat{\mathbf{u}}_d - \frac{1}{V}\sum_n a_n\mathbf{u}_n\right) \end{aligned} \tag{9.6}$$

Flux:

$$\Phi = \frac{\mathbf{s}_f}{\alpha_f}\cdot\left(h_{tf} + h_{sf} - \frac{\mathscr{I}(\rho)}{\hat{\rho}}(\nabla p)_f\right) \tag{9.7}$$

Since this method has been implemented in the given solver no more velocity overshoots and instabilities were observed. This is also confirmed in the test case presented in Section 11.2.

As mentioned in the beginning of this chapter, also the viscosity has a jump behavior at the free-surface. As the viscosity only occurs inside the diffusive term, this error is not destabilizing the solution. Hence, a small damping of the free-surface waves might occur, as the shear forces of the air phase acting on the water are bigger due to the smearing of the viscosity. Applying special interpolation methods for the viscosity, like the *harmonic mean* instead of central differencing can reduce or remedy this problem. Still, this has not been investigated in the scope of this work.

10. Sea Waves

Predicting the motion of and the flow around yachts in waves using RANSE-solvers is not very common for flow analysis of sailing yachts. One of the problems is the big amount of possible combinations of different waves from different directions leading to a huge test matrix. This results in a lot of computational effort and furthermore makes the evaluation very complicated. If at all applied, such sea wave simulations are usually not used for the hull optimization, but only for the proof that the vessel fulfills special requirements written in the design contract.

Another difficulty in sea wave simulation is the generation and correct propagation of the waves using the here presented numerical methods. The special requirements onto the time discretization method were discussed already in Chapter 4. Also, the sea wave induced problem occurring in typical two-equation turbulence models was addressed earlier in Chapter 6.

Left open is the question of the wave generation at the inlet and the treatment of the sea waves at the outlet. Especially the boundary conditions at the outlet of the flow domain produce unwanted reflections of the waves without the application of additional methods. Different methods suppressing such reflections exist leading from satisfying to unsatisfying results. A suitable solution is to slightly damp the waves in front of the outlet. One group of such damping methods can be classified as *Sponge Layer Methods*. They are based on a damping-zone next to the boundary in which a source term is added to the governing equations. The source term usually weakens the vertical component of the fluid velocity which prevents the wave of moving through this zone. These methods deliver good results with reflections less than 2% but require a parameter adjustment by the user depending on the simulated wave.

Another way using a damping zone is presented in Jacobsen, Fuhrman, and Fredsøe 2012. It explicitly relaxes the velocity to zero and the volume fraction to values of an undisturbed free-surface. This method introduces some numerical problems due to the explicit manipulation of the results of the Reynolds-averaged Navier-Stokes equations.

In Higuera, Lara, and Losada 2013 a method suppressing reflections at the outlet based on active filtering is presented. This method manipulates the outlet boundary condition without using a damping-zone. Indeed the damping quality is not satisfying and reflections up to 15% occur.

Wave damping is not only interesting for ships in sea waves. Also the simulation of free-surface flow around ships in smooth water may benefit of a wave-damping method. Here, the waves generated by the ship might also be reflected from the outlet which inhibits a 100% steady-state solution. Stretching the grid in front of the outlet will prevent reflections, but a proper wave-damping method might reduce the effort for the user. Furthermore simulating offshore structures in waves require an adequate method to prevent wave reflections.

This chapter first describes a common wave damping approach, which delivers good results but

requires a parameter adjustment depending on the wave characteristics. Afterwards, a newly developed wave damping method will be derived. This is followed by the description of the wave generation method. Finally, the results of the new method are compared to the typical used standard method using a 2D test case with waves of different characteristics followed by an application onto a sailing yacht simulation in head waves.

10.1. Common Wave Damping Methods

Two widely used reliable wave damping methods are described in Choi and Yoon 2009 and Park, Kim, and Hideaki Miyata 1999, whereby the first will be described here. The method is based on a sponge layer which can be derived by the typical equations for porous media. The damping is achieved by adding a source term to the momentum equation inside a user-defined damping-zone in front of the outlet. The source term is added to the vertical z-component of the momentum equation and can be written as

$$s_z^d = -\rho \left(f_1 + f_2 |u_z|\right) w u_z \tag{10.1}$$

with the weight-function

$$w = \frac{e^{\kappa} - 1}{e^1 - 1} \tag{10.2}$$

and the character of the blending function

$$\kappa = \left(\frac{x - x_{sd}}{x_{ed} - x_{sd}} \right)^{\zeta} \tag{10.3}$$

with ζ usually set to 3.5. Here ρ is the density of the fluid and u_z is the the vertical velocity component. The parameter f_1 gives the amount of linear damping, f_2 the amount of quadratic damping. The weight factor w depends on the location inside the domain and helps to smoothly fade in the source term in the damping zone. The wave propagation direction is given by x with x_{sd} as the start and x_{ed} as the end x-coordinate of the damping zone.
This method is implemented in commercial codes like STAR-CCM+ or in a slightly different form in ANSYS Fluent. It is deeply investigated in R. Perić and Abdel-Maksoud 2016 showing, that the method delivers a reliable damping with satisfying damping quality. Nevertheless the parameters f_1 and f_2 have to be adjusted for different waves. Scaling laws for adjusting these parameters are given in R. Perić and Abdel-Maksoud 2016 and Robinson Perić et al. 2021. Assuming optimal chosen parameters for a regular and monochromatic wave an adjustment is necessary if the wave changes in its scale, where no adjustment is required if the computational mesh or wave steepness is changed. Also no adjustment is required for different lengths of the damping zone. However, the maximal achievable damping quality depends on this length and at least two wavelengths are recommended.

10.2. Derivation of a New Wave Damping Method

In Jacobsen, Fuhrman, and Fredsøe 2012 wave damping is achieved by relaxing the velocity $\mathbf{u}$ and the volume fraction α explicitly after solving for the volume fraction. Explicit relaxation is done with the

generic equation

$$\phi_{\text{relaxed}} = r\phi + (1-r)\phi^{\text{t}} \quad . \tag{10.4}$$

Here ϕ is a generic quantity, the superscript t signifies the target value. The relaxation factor r depends on previously mentioned damping weight w:

$$r = 1 - w \quad . \tag{10.5}$$

The method can deliver a good damping quality but has significant disadvantages. The volume fraction is relaxed to values which assume an undisturbed flat free-surface at constant height and all three velocity components are relaxed to zero. This forbids additional current or boat speed superposing with the orbital velocity of the waves. Additionally this delivers some kind of Dirichlet boundary condition (BC) at the outlets, whereby a Neumann BC is desirable. Furthermore such an explicit relaxation will prevent the convergence of the SIMPLE-algorithm. Still, this relaxation approach and the sponge layer approach described in Section 10.1 are the inspiration for the new method.

For the derivation of the new method a simplified description of the linearized semi-discretized momentum equation is used. This version lacks the detailed information about the interpolation and reconstruction of the individual terms, for the detailed version see Chapter 8. The simplified, linearized, semi-discretized momentum equation is written as

$$a_{\text{d}}\mathbf{u}_{\text{d}}^{q+1} + \sum_n a_{\text{n}}\mathbf{u}_{\text{n}}^{q+1} = -\nabla p^{q} + \mathbf{s}_{\text{w/o p}} \quad . \tag{10.6}$$

Here, a represents the elements of the coefficient matrix $\mathbf{A}$ and the subscripts d and n mark the main diagonal- and neighbor-elements. All sources and contributions to the right hand side except the pressure gradient are included in $\mathbf{s}_{\text{w/o p}}$. The solution of the current SIMPLE-iteration is marked with $q+1$ and the solution of the last iteration with q. It is not distinguished between the first prediction of the velocity and the corrected velocity of the same iteration. Rearranging (10.6) to $\mathbf{u}_c$ yields the simplified velocity equation

$$\mathbf{u}_{\text{d}}^{q+1} = \frac{1}{a_{\text{d}}}\left(-\nabla p^{q+1} + \mathbf{s}_{\text{w/o p}} - \sum_n a_{\text{n}}\mathbf{u}_{\text{n}}^{q}\right) \quad . \tag{10.7}$$

Substituting (10.7) into the continuity equation (2.2) and rearranging yields the simplified Poisson equation for the pressure

$$\nabla\cdot\left(\frac{1}{a_{\text{d}}}\nabla p^{q+1}\right) = \nabla\cdot\frac{1}{a_{\text{d}}}\left(\mathbf{s}_{\text{w/o p}} - \sum_n a_{\text{n}}\mathbf{u}_{\text{n}}^{q}\right) \quad . \tag{10.8}$$

The main idea of the new wave damping method is to reduce the vertical velocity component by the use of an implicit relaxation included in the momentum equation. Considering the discretized momentum equation $\mathbf{A}\cdot\mathbf{u} = \mathbf{s}$ with the coefficient matrix $\mathbf{A}$, the velocity vector $\mathbf{u}$ and all explicitly treated terms on the right hand side included in the source $\mathbf{s}$, the implicit relaxation of all velocity components can be done with

$$\frac{1}{r}a_{\text{d}}\mathbf{u}_{\text{d}}^{q+1} + \sum_n a_{\text{n}}\mathbf{u}_{\text{n}}^{q+1} = \mathbf{s} + \frac{1-r}{r}a_{\text{d}}\mathbf{u}_{\text{d}}^{\text{t}} \quad . \tag{10.9}$$

Here, the subscripts d marks the main diagonal- and n the neighbor-elements of $\mathbf{A}$. As the limiting case, where the relaxation factor r tends to zero the convergence of the equation system is not obvious. Therefore, L'Hôpital's rule was adopted twice showing that $\mathbf{u}_\mathrm{d}^{\mathrm{q}+1}$ tends to $\mathbf{u}_\mathrm{d}^{\mathrm{t}}$ as aspired.
Modifying equation (10.9) to relax only the vertical velocity component is not straightforward. Using this approach the coefficient matrix $\mathbf{A}$ has to be modified. Indeed it is common practice to reuse this matrix for the solution of all three velocity components. Therefore manipulating $\mathbf{A}$ only for the vertical component will produce additional calculation effort and a lot of programing effort to implement this methods into existing numerical codes like OpenFOAM.

To solve this problem the idea is to implement the implicit relaxation with the help of the deferred correction approach. That means, the product of the modified matrix and the velocity is treated explicitly on the right hand side. Additionally the product of the unmodified matrix and the velocity is added on both, explicit and implicit sides leading to the original unmodified left hand side. If the equation system is converging, the terms with the unmodified matrix are canceling each other out and the solution depends only on the modified matrix. This allows to modify only the right hand side, more precisely only the z-component of the right hand side.

In the following two approaches to apply the deferred correction method are described. The first approach might be the obvious way to go but leads to a diverging equation system, as it will be shown. The second approach leads to our final damping method and a converging equation system.

10.2.1. First Approach (divergent)

In the following equations all terms on the left hand side consider the unknown velocity from the current iteration $q+1$ and all terms on the right hand side use the known values of the last iteration q. For convergence the velocity u^q should tend to u^{q+1}. Starting from the z-component of the vector equation (10.9)

$$\frac{1}{r}a_\mathrm{d}u_\mathrm{zd}^{q+1} + \sum_n a_\mathrm{n}u_\mathrm{zn}^{q+1} = s_\mathrm{z} + \frac{1-r}{r}a_\mathrm{d}u_\mathrm{zd}^{\mathrm{t}} \tag{10.10}$$

and applying $a_\mathrm{d}u_\mathrm{zd}$ on both sides leads to

$$\frac{1}{r}a_\mathrm{d}u_\mathrm{zd}^{q+1} + a_\mathrm{d}u_\mathrm{zd}^{q+1} + \sum_n a_\mathrm{n}u_\mathrm{zn}^{q+1} = s_\mathrm{z} + \frac{1-r}{r}a_\mathrm{d}u_\mathrm{zd}^{\mathrm{t}} + a_\mathrm{d}u_\mathrm{zd}^{q} \quad . \tag{10.11}$$

Putting the term $\frac{1}{r}a_\mathrm{d}u_\mathrm{zd}^{q+1}$ from the left to the right hand side changes its iteration index and leads to

$$a_\mathrm{d}u_\mathrm{zd}^{q+1} + \sum_n a_\mathrm{n}u_\mathrm{zn}^{q+1} = s_\mathrm{z} + \frac{1-r}{r}a_\mathrm{d}u_\mathrm{zd}^{\mathrm{t}} + a_\mathrm{d}u_\mathrm{zd}^{q} - \frac{1}{r}a_\mathrm{d}u_\mathrm{zd}^{q} \quad . \tag{10.12}$$

Simplifying this equation delivers the final equation for the first approach

$$a_\mathrm{d}u_\mathrm{zd}^{q+1} + \sum_n a_\mathrm{n}u_\mathrm{zn}^{q+1} = s_\mathrm{z} + \frac{1-r}{r}\left(a_\mathrm{d}u_\mathrm{zd}^{\mathrm{t}} - a_\mathrm{d}u_\mathrm{zd}^{q}\right) \quad . \tag{10.13}$$

The whole relaxation of the z-component for the velocity is included in one source term on the right hand side. Therefore it is no more necessary to manipulate the coefficient matrix $\mathbf{A}$ and it is possible to include the relaxation of the z-component in the typical vector form of the momentum equation used in codes like OpenFOAM.

10.2.2. Investigation on the Convergence Behavior - 1st Approach

Using the Jacobi method to solve equation (10.13) leads to

$$u_{zd}^{q+1} = \frac{1}{a_d}\left(s_z + \frac{1-r}{r}\left(a_d u_{zd}^{t} - a_d u_{zd}^{q}\right) - \sum_n a_n u_{zn}^{q}\right) \quad . \tag{10.14}$$

For wave damping ($u_{zd}^{t} = 0$) with full relaxation ($r \to 0$) the source term s_z becomes negletable and the solution of the equation is $u_{zd}^{q+1} \to -\infty u_{zd}^{q}$. Therefore no convergence is possible for small r.

10.2.3. Second Approach

For the second approach (10.10) is multiplied with the relaxation factor r before applying the deferred correction

$$a_d u_{zd}^{q+1} + r\sum_n a_n u_{zn}^{q+1} = r s_z + (1-r)\, a_d u_{zd}^{t} \quad . \tag{10.15}$$

In Eq. (10.10) of the first approach the main diagonal elements of $\mathbf{A}$ are multiplied with $\frac{1}{r}$. Now, in (10.15) the neighbor elements of $\mathbf{A}$ and the right hand side term s_z are multiplied with r. Applying the deferred correction method to (10.15) to get rid of the relaxation factor on the left hand side leads to

$$a_d u_{zd}^{q+1} + \sum_n a_n u_{zn}^{q+1} = r s_z + (1-r)\left(a_d u_{zd}^{t} + \sum_n a_n u_{zn}^{q}\right) \quad . \tag{10.16}$$

Here, the original right hand side term s_z is manipulated. This needs to be considered in the derivation of the pressure equation. The chosen solution considering this as easy as possible is described in Subsection 10.2.5.

10.2.4. Investigation on the Convergence Behavior - 2nd Approach

Using the Jacobi method again to solve eq. (10.16) gives

$$u_{zd}^{q+1} = \frac{1}{a_d}\left(r s_z + (1-r)\left(a_d u_{zd}^{t} + \sum_n a_n u_{zn}^{q}\right) - \sum_n a_n u_{zn}^{q}\right) \quad . \tag{10.17}$$

For full relaxation ($r \to 0$) one gets the aspired behavior $u_{zd}^{q} \to u_{zd}^{t}$ where u_{zd}^{t} is zero for wave damping.

10.2.5. Manipulation of the Original Right Hand Side

Using (10.16) requires to manipulate the original right hand side s_z including the terms depending on the pressure p. Therefore it is necessary to build a new pressure equation. This can be done straightforward but is not our finally chosen way. For the sake of completeness this way will be given here, first. Afterward a simpler way leading to the final method will be described.

First way

With respect to (10.6), the vector form of the simplified, semi-discretized momentum equation considering the wave damping method can be written as:

$$a_\mathrm{d}\mathbf{u}_\mathrm{d}^{q+1} + \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^{q+1} = \mathbf{R}\cdot\left(-\nabla p^q + \mathbf{s}_{\mathrm{w/o\ p}}\right) + \mathbf{s}^* \quad . \tag{10.18}$$

Here, $\mathbf{s}_{\mathrm{w/o\ p}}$ is the original right hand side without the pressure gradient ($\mathbf{s} = -\nabla p + \mathbf{s}_{\mathrm{w/o\ p}}$). The tensor $\mathbf{R}$ allows to manipulate only the z-component of the equation and is defined as $\mathbf{R} = \begin{bmatrix} r_\mathrm{x} & 0 & 0 \\ 0 & r_\mathrm{y} & 0 \\ 0 & 0 & r_\mathrm{z} \end{bmatrix}$ with the relaxation factors $r_\mathrm{x} = r_\mathrm{y} = 1.0$ and $r_\mathrm{z} = r$ for the three Cartesian directions. The source term $\mathbf{s}^*$ is defined as

$$\mathbf{s}^* = \left(\delta_{ij} - \mathbf{R}\right)\cdot\left(a_\mathrm{d}\mathbf{u}_\mathrm{d}^\mathrm{t} + \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^q\right) \tag{10.19}$$

with the Kronecker-delta $\delta_{ij} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Rearranging (10.18) for $\mathbf{u}_\mathrm{d}^{q+1}$ leads to the new velocity equation allowing to calculate the corrected velocity $\mathbf{u}_\mathrm{d}^{q+1}$

$$\mathbf{u}_\mathrm{d}^{q+1} = \frac{1}{a_\mathrm{d}}\left(\mathbf{R}\cdot\left(-\nabla p^{q+1} + \mathbf{s}_{\mathrm{w/o\ p}}\right) + \mathbf{s}^* - \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^q\right) \quad . \tag{10.20}$$

Substituting the velocity of the continuity equation (2.2) with (10.20) and rearranging gives the adopted pressure equation

$$\nabla\cdot\left(\frac{1}{a_\mathrm{d}}\mathbf{R}\cdot\nabla p^{q+1}\right) = \nabla\cdot\frac{1}{a_\mathrm{d}}\left(\mathbf{R}\cdot\mathbf{s}_{\mathrm{w/o\ p}} + \mathbf{s}^* - \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^q\right) \quad . \tag{10.21}$$

Second way

For the second way the momentum equation is written in a way containing the unmodified right hand side $\mathbf{s}$ and an additional source term

$$a_\mathrm{d}\mathbf{u}_\mathrm{d}^{q+1} + \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^{q+1} = \underbrace{-\nabla p^q + \mathbf{s}_{\mathrm{w/o\ p}}}_{\mathbf{s}} + \mathbf{s}_\mathrm{wavedamping} \tag{10.22}$$

with

$$\mathbf{s}_\mathrm{wavedamping} = \left(\delta_{ij} - \mathbf{R}\right)\cdot\left(\nabla\tilde{p}^q - \mathbf{s}_{\mathrm{w/o\ p}} + a_\mathrm{d}\mathbf{u}_\mathrm{d}^\mathrm{t} + \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^q\right) \quad . \tag{10.23}$$

It is important to notice, that the pressure gradient is used in two terms. To build the pressure equation it is only solved for the pressure of the original right hand side $\mathbf{s}$. The pressure gradient inside the source term for the wave damping $\mathbf{s}_\mathrm{wavedamping}$ is fixed and therefore marked with a tilde. The equation for the corrected velocity becomes

$$\mathbf{u}_\mathrm{d}^{q+1} = \frac{1}{a_\mathrm{d}}\left(-\nabla p^{q+1} + \mathbf{s}_{\mathrm{w/o\ p}} + \mathbf{s}_\mathrm{wavedamping} - \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^q\right) \tag{10.24}$$

and the pressure equation becomes

$$\nabla \cdot \left(\frac{1}{a_\mathrm{d}} \nabla p^{q+1} \right) = \nabla \cdot \frac{1}{a_\mathrm{d}} \left(\mathbf{s}_{\mathrm{w/o\ p}} + \mathbf{s}_{\mathrm{wavedamping}} - \sum_n a_\mathrm{n} \mathbf{u}_\mathrm{n}^q \right) \quad . \tag{10.25}$$

Because $\mathbf{s}_{\mathrm{wavedamping}}$ contains the pressure of the last iteration p^q the solution of the pressure could be interpreted as deferred inside the wave damping zone.

This is the final wave damping method made up of only one additional source term. The original right hand side $\mathbf{s}$ can be pre-calculated and then reused to build the wave-damping source term $\mathbf{s}_{\mathrm{wavedamping}}$. The weight function w included in the relaxation factor r has not the final form and will be substituted with the optimized function w_{opt} as described in Chapter 10.4.3.

10.3. Wave Generation

The main part of the wave generation is based on the OpenFOAM library *waves2Foam* (Jacobsen, Fuhrman, and Fredsøe 2012) which has been incorporated into the presented. An alternative library for sea wave simulations was presented by Higuera, Lara, and Losada 2013 and Iturrioz et al. 2015. A detailed comparison of alternative OpenFOAM solvers for numerical wave tanks has been given in Schmitt et al. 2020.

The waves2Foam library comes with a huge variety of functions for different monochromatic waves or complete wave spectra. At the inlet the values for the velocities and volume fractions are prescribed according to the chosen wave theory. An equation for the pressure value itself or the pressure gradient is missing, which is a significant problem. The chosen solution is based on using a Neumann boundary condition with a zero gradient for the pressure. Certainly, the assumption of a zero gradient is wrong in the presence of a wave and for many 3D simulations the wave will collapse behind the inlet. Therefore the waves2Foam library comes with the option to use a relaxation zone behind the inlet, which smoothly blends from the prescribed potential solution to the solution of the RANSE solver. The blending is implemented using an explicit relaxation. This means, after solving the momentum equation of the RANS equation system, the solution will be explicitly relaxed to the potential solution in the region of the relaxation zone. The same approach is applied for the Volume-of-Fluid equation. Indeed, such explicit approach will prevent the convergence of the two equations, as these equation do not *know* the potential solution. Hence, an implicit relaxation for the momentum and Volume-of-Fluid equation was implemented in the given work. The relaxation factor has the reversed characteristic of the relaxation factor of the damping-zone. Meaning, full relaxation is applied at the inlet going to no relaxation at the end of the generation-zone. The target values $\mathbf{u}^\mathrm{t}$ and α^t are calculate with the chosen wave theory for example 5th-order Stokes. All three velocity components are relaxed to the target value by using (10.9). The relaxation for the volume fraction is applied in the same way

$$\frac{1}{r} a_{\mathrm{VOF\ d}} \alpha_\mathrm{d}^{q+1} + \sum_n a_{\mathrm{VOF\ n}} \alpha_\mathrm{n}^{q+1} = s_{\mathrm{VOF}} + \frac{1-r}{r} a_{\mathrm{VOF\ d}} \alpha_\mathrm{d}^\mathrm{t} \tag{10.26}$$

with the diagonal elements $a_{\mathrm{VOF\ d}}$ and neighbor elements $a_{\mathrm{VOF\ n}}$ of the coefficient matrix of the Volume-of-Fluid transport equation. The term s_{VOF} contains all possible source terms or contributions to the right hand side.

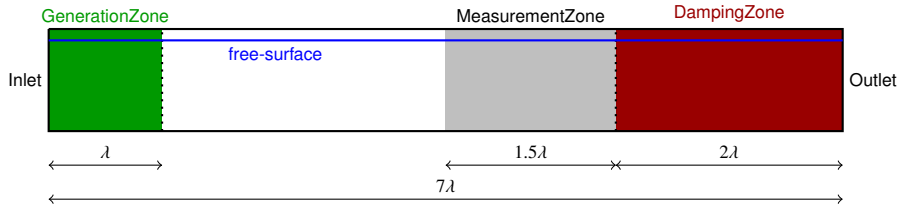


Figure 10.1.: 2d Simulation Setup - clinched in z-direction

The length of the relaxation zone was not tested systematically, but a length of 1.0λ produces a reliable wave generation in all simulations.

The free-surface is initialized wave-less in the hole domain. For the wave generation the T-soft-parameter of the waves2Foam library is used. This generates a wave growing from a smooth surface to the full height over the given time T_{soft} in the generation-zone. The growing time T_{soft} is set to the wave-period T. Due to the relaxation approach, the relaxation factor becomes small at the end of the generation zone. Therefore, while growing the wave gets smeared at the end of the zone, because the values are set to the new height only partially which leads to very small oscillations in the wave. Still this approach is assumed to be acceptable. These oscillations are quantified in Section 10.4.

10.4. Verification and Weight Function Optimization

To estimate the damping quality of the new wave damping method a test-case very close to the one presented in R. Perić and Abdel-Maksoud 2016 is chosen. The test-case is also used to optimize the function for the relaxation factor for monochromatic waves. For comparison, the typical sponge layer method of Section 10.1 was implemented.

10.4.1. Numerical Simulation Setup

The test-case is based on a 2D wave tank. Figure 10.1 shows the domain and its measurements. The wave is generated in the generation zone with a length of 1λ. Afterward the wave is propagated through a region of length 4.0λ. Finally the wave damping is applied in the damping zone of length 2.0λ. The Reflections are measured inside the measurement zone in front of the damping zone as explained later. The domain has a height of 4.5λ filled with water up to 4.0λ. The grid has a resolution of 20 cells per wave-height and 100 cells per wave-length as recommended in R. Perić and Abdel-Maksoud 2016 which refers to R. Perić 2013. The grid is coarsened with growing distance to the wave. The waves are generated by prescribing volume fractions and velocities of 5th-order Stokes waves at the inlet boundary and in the generation zone as explained in Chapter 10.3. A Neumann boundary condition (BC) with zero gradient is applied for the pressure at the inlet. A free slip wall is applied at the bottom. Indeed, instead of using the hydrostatic pressure gradient a Dirichlet BC with given hydrostatic pressure is applied for the pressure at the bottom. This is done to prevent problems with the preservation of the position of the free-surface. Same BC is applied at the top with the exception, that the velocity BC is set to a zero gradient BC to get a behavior as an open water tank. At the outlet a zero gradient BC is applied for all variables. This allows a lowered free-surface at the outlet or a wave traveling through the outlet.

The solver is using a time step of about $\frac{1}{500}$ of the wave period and 10 SIMPLE-Loops. The pressure is relaxed by 0.3, the velocity by 0.7 and the volume fraction by 0.9. The simulation is assumed to be laminar and no turbulence-model is applied.

Simulations were done for different wave scales; a medium sized wave with $\lambda = 4.0$m and $H = 0.16$m, a small wave with $\lambda = 0.04$m and $H = 0.0016$m and a big wave with $\lambda = 400$m and $H = 16.0$m. To compare the dependency of the wave steepness a steep wave with $\lambda = 0.4$ and $H = 0.16$ was simulated. The grid was scaled accordingly.

The influence of the grid is investigated by using a coarse grid with 10 cells per wave height and 50 cells per wave length for the medium sized wave. The influence of the damping zone length is investigated by using a damping zone with a length of 1λ for the medium sized wave. The grid is the same as for the initial, medium sized wave simulation. The measurement zone is also kept at the initial position.

10.4.2. Quantifying the Damping Quality

To quantify the damping quality, the maximal and minimal wave heights H_{max} and H_{min} are measured in the measurement zone in the time interval $[22.0T, 24.0T]$ at 40 evenly distributed times. These wave heights are used to build a reflection coefficient as suggested in Ursell, Dean, and Y. S. Yu 1960

$$C_{\text{R}} = \frac{H_{\text{max}} - H_{\text{min}}}{H_{\text{max}} + H_{\text{min}}} \quad . \tag{10.27}$$

For perfect wave reflection of 100% the coefficient C_{R} becomes 1.0 and zero for no reflection. Considering that the wave train moves with the half speed of a single wave, the chosen time interval allows, that the wave train propagates from the inlet to the outlet and, after the reflection, back to the measurement zone. The wave height is estimated by measuring the minimal and maximal z-Position of the free surface inside the measurement zone. Please notice, that the reflection coefficient includes the wave reflections, but also wave oscillations originating from the wave generation. Additionally a wave, flattening inside the measurement zone due to an insufficient discretization of time or of the convective term will influence the reflection coefficient. The superposition of all these phenomenons may influence the coefficient negatively, but also positively.

Reflections may occur at the outlet but also inside or at the beginning of the damping zone, due to too much dampening.

To estimate the background oscillations (BO) originating from the wave generation and the influence of flattening, all simulations were done without wave damping but with a longer domain of length 25.0λ.

10.4.3. Optimizing the Weight Function

The weight function given in (10.2) does not necessarily guarantee the least reflections. Looking at equation (10.1) of the sponge layer method one can interpret the parameters f_1 and f_2 as a scaling of the weight function w for the different waves. To allow an optimization for the new damping method the weight function is also scaled with the newly introduced scale factor χ

$$w_{\text{opt}} = \chi w \quad . \tag{10.28}$$

The scale factor χ is varied from 0.008 to 1.0 for each wave to achieve the least reflections and to show the dependency of our wave damping method from the wave characteristics.

10.4.4. Results

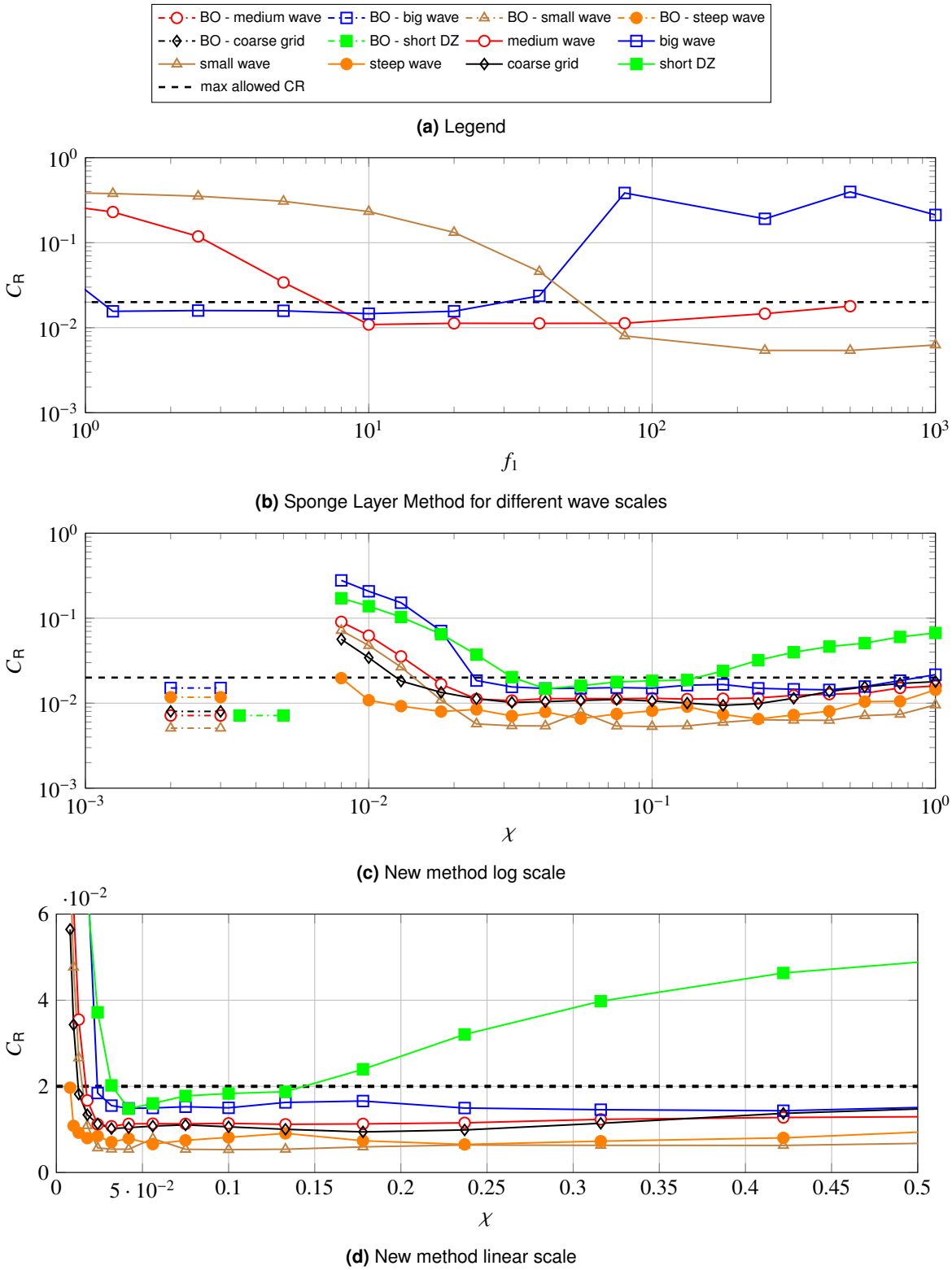
Figure 10.2 shows the results of all simulations. Sub-figure b shows the results of the common linear sponge layer method (with $f_2 = 0.0 = const$), also implemented in our solver. The results are remarkable close to the results presented in R. Perić and Abdel-Maksoud 2016 using the same method but implemented in Star-CCM+. The black, dashed line shows the limit of $C_{\text{R lim}} = 0.02$ for an acceptable reflection. Here, the sponge layer method was only used for different wave scales and one can clearly see that f_1 has to be adjusted to achieve an acceptable reflection. For the quadratic sponge layer method it is referred to the results of R. Perić and Abdel-Maksoud 2016, showing that f_2 has to be adjusted, too.

Sub-figure c shows the results of the new method with both axis scaled logarithmically. The initial background oscillations (BO) are marked with a dashed line in the left area of the diagram. All background oscillations are below the maximal allowed value. This shows that the oscillations produced by the wave generation method mentioned in Chapter 10.3 are acceptable. All curves show the same characteristics and an optimal wave damping is achieved for approximately $0.03 \leq \chi \leq 0.15$. The absolute minimum of each curve corresponds with the background oscillations. Only the simulation with the short damping zone clearly shows a worse damping quality. Additionally the absolute minimum for the steep wave deceeds the corresponding background oscillations. A clear reason can not be given but it is possible that the background oscillations and the reflections are canceling each other out.

The characteristics of the curves can be interpreted in that way, that the increasing reflections in the area $\chi < 0.03$ are reflections which arises predominantly at the outlet due to too little damping and that the increasing reflections in the area $\chi > 0.15$ are reflections which arises predominantly at the beginning or inside the wave damping zone due to too fast wave damping.

Sub-figure d shows the same results as c but with a linear scale. This should emphasize graphically that all curves have the same characteristics. Therefore the new wave damping method allows to use the same weight function for all different monochromatic waves and no user adjustment is required. To make short damping zones practical a scale factor χ of about 0.09 will be recommended here. Nevertheless a damping zone with a length of $2.0\lambda_{\text{max}}$ should be preferred, where λ_{max} should be the largest wave length of a given wave spectrum. One can assume that the evaluation of the damping quality would turn out much better without the presence of the background oscillations.

Figures 10.3 to 10.5 show the wave shapes for different damping parameters. The end of the generation-zone (left) and the beginning of the damping-zone (right) are marked with vertical red lines. Figure 10.3 shows the wave after $40s$ without any wave-damping applied, leading to reflections of 42%. One can clearly see the reflections superposing with the original wave. Figure 10.4 shows the result arising due to too much damping, which still leads to only 1.6% reflection. Here, the free-surface becomes completely flat in front of the outlet. Figure 10.5 shows the wave shape arising due to optimal wave-damping leading to only 1.1% reflection. Here, still a very small wave reaches the outlet. Compared to 10.4, this will even more reduce reflections at the beginning of the damping-zone. Furthermore, possible reflections from the outlet will be damped while traveling back through the damping-zone.



(a) Legend

(b) Sponge Layer Method for different wave scales

(c) New method log scale

(d) New method linear scale

Figure 10.2.: Reflection coefficient for different wave damping methods and different waves
small wave: $\lambda = 0.04$m, $H = 0.0016$m; medium wave: $\lambda = 4.0$m, $H = 0.16$m; big wave: $\lambda = 400$m, $H = 16.0$m; steep wave: $\lambda = 0.4$m, $H = 0.16$m

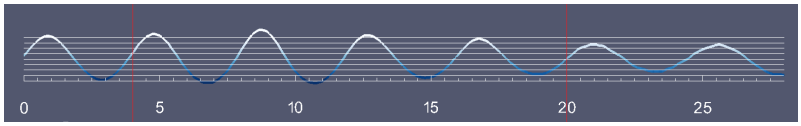


Figure 10.3.: 2D-Wave at $t = 40s$, no damping

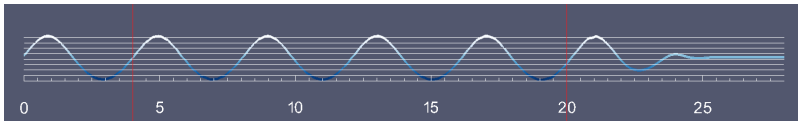


Figure 10.4.: 2D-Wave at $t = 40s$, damping with $\chi = 1.0$ (too much damping)

10.5. Application to a Yacht in Head Waves

This section shows an application of the new wave damping method to a yacht in head waves. This is a good example for the superposition of the sea waves with the small waves of the yacht and furthermore with the velocity of the yacht.

The yacht is a Class 40 (C40) designed by VPLP, France and has a length over surface of $L_{\text{oS}} = 12$m. The waves have a length of $\lambda = 18$m, a height of $H = 0.4$m and a period of $T = 3.489$s. Second order Stokes theory is used for wave generation. The yacht is accelerated to $u = 4.4$m/s in the first second. This leads to an encounter frequency of $0.53\frac{1}{\text{s}}$. The yacht is free in pitch and heave.

The solution-domain uses a wave-generation zone of 1λ and a wave-damping zone of 2λ. The free flow begins $1L_{\text{oS}}$ in front of the yacht and ends $2L_{\text{oS}}$ behind the yacht. The grid is generated using the OpenFOAM mesher snappyHexMesh, but with the use of some small in-house code and a lot of scripting to get a mesh of good quality with anisotropic refinement. The grid uses 100 cells per wave length and 20 cells per wave height. Depending on the part of the yacht, seven to nine prism-layers are applied leading to $y+$ values of 40 to 90 underwater. The kelvin-refinement ends in front of the wave-damping zone. The cells inside the kelvin-refinement have a size less or equal to 0.125m. The final mesh has 8.5×10^6 cells.

The modified $k-\omega$-SST turbulence model is used. The modification consist of a correct consideration of the density-derivations and a different production term as described in Jacobsen, Fuhrman, and Fredsøe 2012, which is the interim method as described in Chapter 6. The timestep was set to 0.005s and the solver uses 5 SIMPLE-Loops. The simulation was calculated on a cluster with three knots, each with two 6-core CPUs. The simulation took about 65 hours.

Figures 10.6 and 10.7 show the pressure and the viscous forces. One can clearly see a periodic behavior and no significant disturbance due to wave reflections. The viscous force in x-direction has the most variation. This can be explained with ventilation under the hull composed of correct ventilation due to encountering a trough and incorrect numerical ventilation due to smearing of the free-surface.

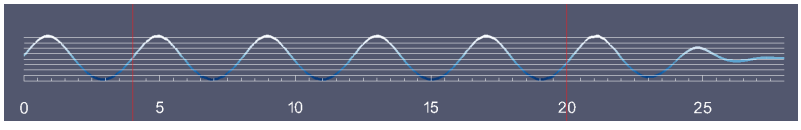


Figure 10.5.: 2D-Wave at $t = 40s$, damping with $\chi = 0.1$ (optimal damping)

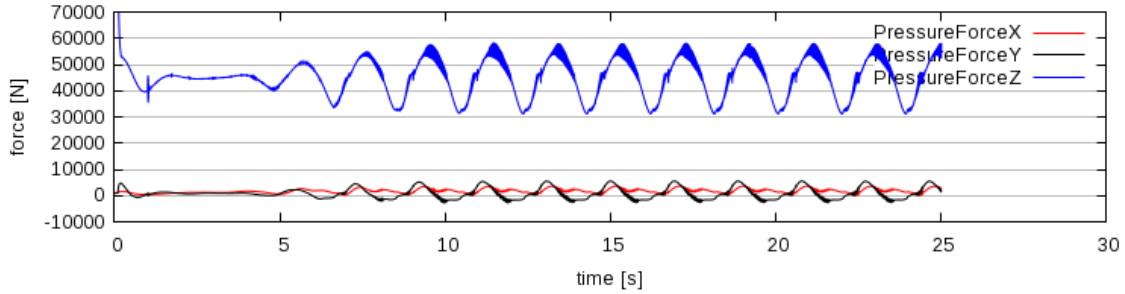


Figure 10.6.: C40-design in head waves - pressure forces

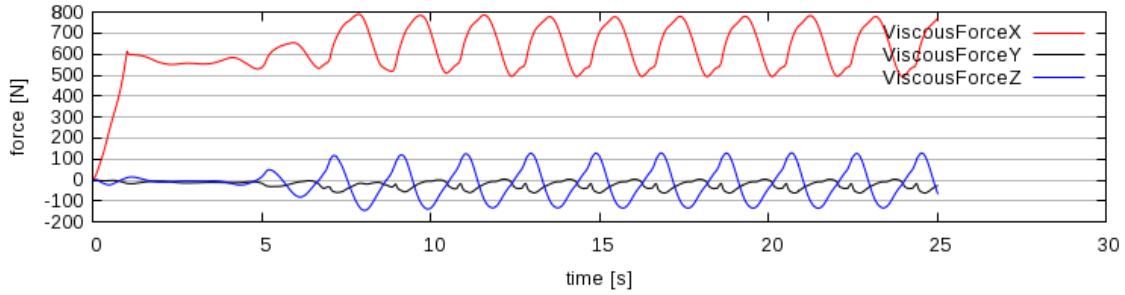


Figure 10.7.: C40-design in head waves - viscous forces

First simulations using a coarser grid (4.4×10^6 cells, three prism layers, coarser kelvin refinement) show much more variations in the viscous forces in the x-direction.

Figures 10.8 and 10.9 show a close view onto the hull and its wave system for two different times. Figure 10.8 shows the hull diving into the sea wave, whereas Fig. 10.9 shows the hull with the bow knuckle significantly above the free-surface. In both cases, one can clearly see a smooth and good resolved kelvin wave pattern and furthermore breaking waves at the hull.

Figures 10.10 and 10.11 show wave patterns at different times. The end of the generation-zone and the beginning of the damping-zone are marked with two red x-planes in Fig. 10.11. As one can see, the waves get damped in front of the outlet. The large scaled sea waves are not dampened completely at the outlet as expected for the chosen χ of 0.09. Apart from that, the small scaled wave-system of the yacht seems to be dampened completely at the outlet. The reason for this is first the length of the damping-zone, which is significantly longer than twice the wave length of the yacht induced waves. Therefore the damping for this small-scaled waves should be of a quality better than required. The second reason is the kelvin refinement ending in front of the damping-zone. At the y-max and y-min domain borders (the sides of the domain) the free traveling waves show no significant differences to each others. This underscores first the good wave propagation due to the 2nd order time discretization and secondly the suppression of wave reflections.

10.6. Conclusion

A new wave damping method was derived and investigated for monochromatic waves. The simulations show an overall good damping quality with reflections less than 2%. For different wave characteris-

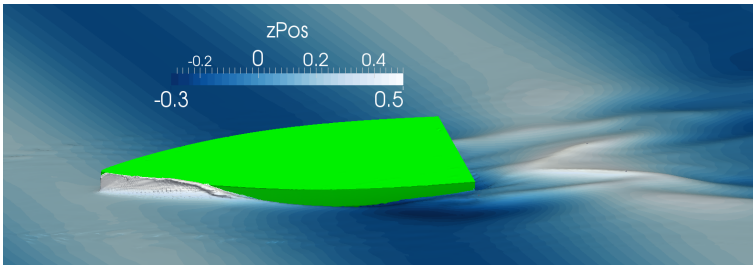


Figure 10.8.: C40-design close-up at $t = 23.1$

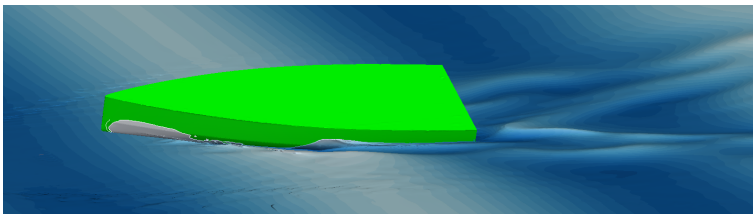


Figure 10.9.: C40-design close-up at $t = 23.9$

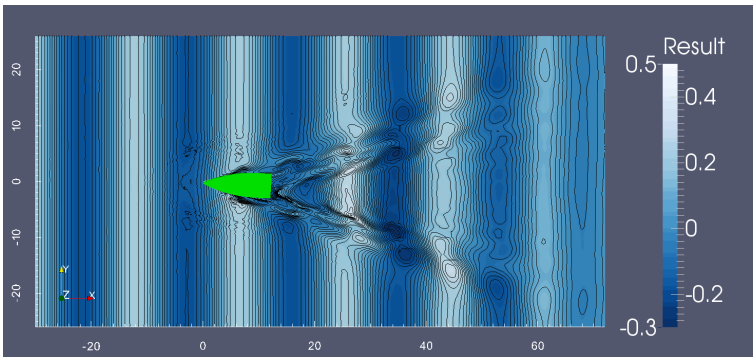


Figure 10.10.: C40-design wave pattern at $t = 23.9$

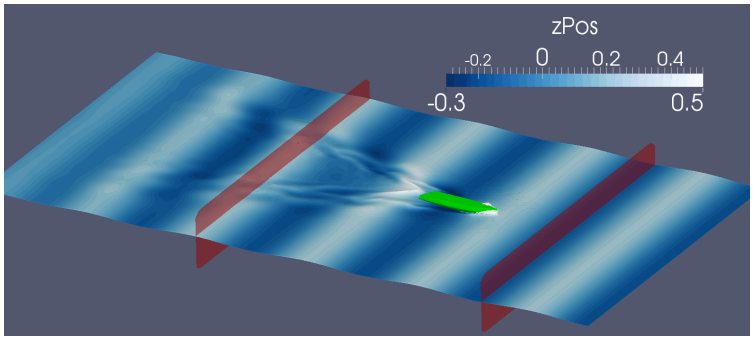


Figure 10.11.: C40-design 3D view at $t = 23.1$

tics the same parameters leads to the best damping quality. Therefore the method can be seen as adjustment-free in the scope of the investigated waves. The damping quality for irregular or breaking waves has not been investigated and is an open topic for future investigations. Nevertheless, especially for irregular waves, which can be seen as a superposition of regular waves, the authors expect a good damping quality, due to the adjustment-free behavior for monochromatic regular waves. The application of the method to a yacht in head waves emphasizes the user-friendly applicability, as it delivers a periodic solution with less variation just by activating the damping method with the optimized parameters.

11. Case Studies

This chapter shows three applications of the newly developed solver. First the solver is applied onto the *Rayleigh Taylor Instability*. This case allows to compare the simulation results with results from analytical equations.

Afterwards, the solver is applied on to the *Sysser 60* Test case. Within this work, this case was also simulated using the commercial solver Star-CCM+ using the same grid and a very similiar solver setup The results are compared qualitatively.

Finally, the solver is used to simulate the major part of the *Wide and Light* study. The simulation results are compared quantitatively to public towing tank results and numerical results from various users applying several CFD-codes.

Further applications of the new solver were published in Graf, Renzsch, and Meyer 2016, Graf, Meyer, et al. 2017 and Graf and Freiheit 2020.

11.1. Rayleigh Taylor Instability

Superposing a heavy fluid over a light fluid results in an unstable equilibrium. Any perturbation growth over time. This phenomena is called Rayleigh-Taylor instability. Analytical results are given in Chandrasekhar 2014 which allows to use this case for verification. As mentioned in Daly 1967 the rate of growth of the perturbation is predicted in Lord Rayleigh (John William Strutt) 1894 and Taylor 1950 by a linear analysis and generalized for the effects of viscosity in Bellman and Pennington 1954, Hide 1955 and Chandrasekhar 2014 and for the effects of surface tension in Taylor 1950, Bellman and Pennington 1954 and Chandrasekhar 2014. The rate of growth depends on the densities of the two fluids, the viscosities, the wave length and the gravity. Experimental results are given in Lewis and Taylor 1950 and first Numerical Results are given in Daly 1967. Based on Daly 1967 the case has also been used for verification of an interface capturing method in Patrick Queutey and Michel Visonneau 2007.
This chapter first describes the test case. Finally, the results achieved with the new solver are compared to the analytical and numerical solutions mentioned above.

For the solution of viscous fluids Chandrasekhar restricts the viscosities of the fluid to the case when both are the same. Furthermore he fixed the viscosities to $\nu = \nu_1 = \nu_2 = 1.0\frac{cm^2}{s}$ and the gravity to $g = 1.0\frac{cm}{s^2}$ and varies only the wavenumber k. For the calculation, one first defines pairs of Q and y for $1 < y$. For viscous fluids without surface tension Chandrasekhar gives the formula

$$Q = \frac{y-1}{\alpha_2 - \alpha_1}\left(y^3 + (1 + 4\alpha_1\alpha_2)\,y^2 + (3 - 8\alpha_1\alpha_2)\,y - (1 - 4\alpha_1\alpha_2)\right) \tag{11.1}$$

with

$$\alpha_1 = \frac{\rho_1}{\rho_1 + \rho_2} \text{ and } \alpha_2 = \frac{\rho_2}{\rho_1 + \rho_2} \quad . \tag{11.2}$$

Finally pairs of k and n are calculated. The wavenumber k in the unit $\left(\frac{g}{\nu^2}\right)^{\frac{1}{3}}$ is calculated with

$$k = Q^{-\frac{1}{3}} \tag{11.3}$$

and the rate of growth n^* in the unit $\left(\frac{g^2}{\nu}\right)^{\frac{1}{3}}$ is calculated with

$$n^* = \left(y^2 - 1\right) Q^{-\frac{2}{3}} \quad . \tag{11.4}$$

For the numerical solution different wave numbers requires different grids. Therefore it is reasonable to vary the viscosity instead. The results are then plotted over the Reynolds number R_e for comparison, whereas the Reynolds number is calculated by

$$R_\mathrm{e} = \frac{\lambda^{\frac{3}{2}} g^{\frac{1}{2}}}{\nu} \tag{11.5}$$

with the wavelength λ given directly or calculated by $\lambda = \frac{2\pi}{k}$.

11.1.1. Case Setup

The numerical simulation is done on a rectangular, two dimensional grid. The initial free surface is a flat plane located at the half grid height at the location $y = 0$. The grid has a width of L which is given by $L = \frac{\lambda}{2}$. The initial depth D of each fluid is given by the ratio $\frac{D}{L} = 1.5$. The chosen wavelength of $\lambda = 0.04m$ leads to a width of $L = 0.02m$ and a height of $2D = 0.06m$. The grid, consisting of squared cells, is 40 cells wide and 120 cells high. The cells have a edge length of $\delta y = 0.0005m$. This is the same resolution as used in Patrick Queutey and Michel Visonneau 2007 for the whole calculation and Daly 1967 for the calculation of the used marker particles. The gravity is set to $g_x = 0.0\frac{m}{s^2}$ and $g_y = -1.0\frac{m}{s^2}$. Daly makes following recommendation for the initial perturbation which is conservative in mass

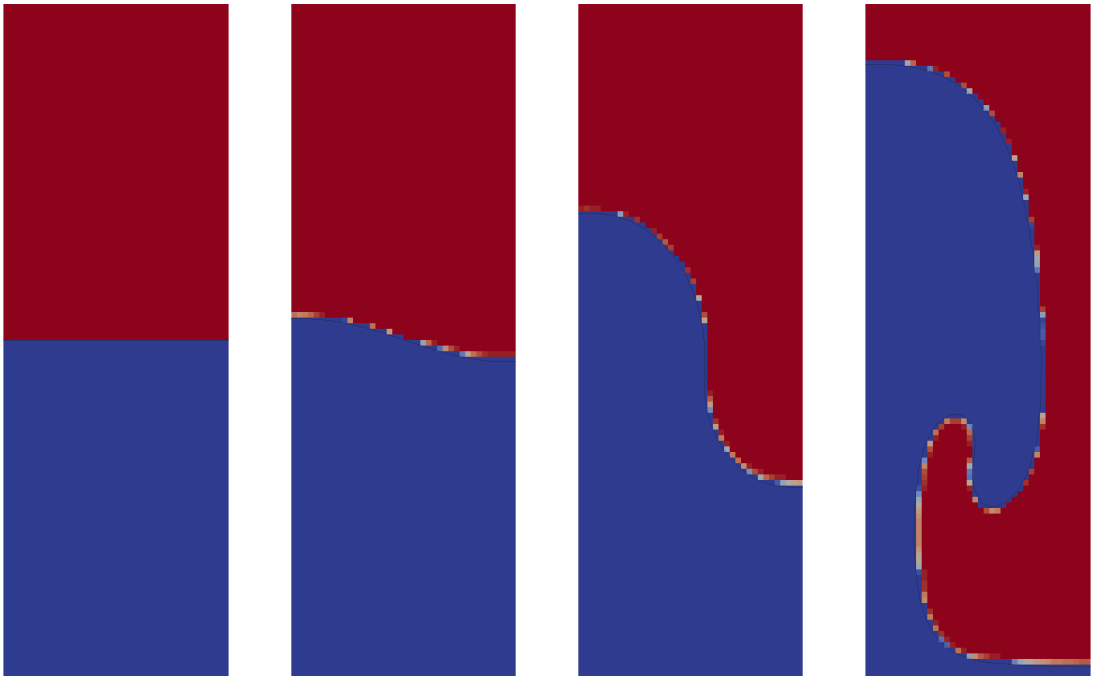
$$u_x = \frac{\pi A \delta y}{2L} \sin\left(\frac{\pi x}{L}\right)(2H(y) - 1) \exp\left(\frac{-x|y|}{L}\right) \tag{11.6}$$

$$u_y = \frac{\pi A \delta y}{2L} \cos\left(\frac{\pi x}{L}\right) \exp\left(\frac{-x|y|}{L}\right) \tag{11.7}$$

with the perturbation amplitude A and the Heavyside function H given by

$$H(y) = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{if } y < 0. \end{cases} \tag{11.8}$$

For viscous flow Daly recommends an amplitude of $A = 1.0$. All four boundaries are set to a slip condition. The viscosity is varied from $1.6E-5\frac{m^2}{s}$ to $8.0E-3\frac{m^2}{s}$. This results in Reynolds numbers from 10 to 500.



(a) t=0.0s **(b)** t=0.5s **(c)** t=1.0s **(d)** t=1.5s

Figure 11.1.: Rayleigh-Taylor instability: Wave at different timesteps for $R_e = 72$

11.1.2. Results

Wave amplitudes for different timesteps are shown exemplarily in Fig. 11.1 for a Reynolds number of $Re = 72$. After solving the different cases, for each case the logarithm of the maximal displacement of the free surface is plotted over the time, see Fig. 11.2. The maximal displacement of the free surface is obtained by averaging the magnitude of the maximal displacement up- and downwards. The slope n of the linear part of the curve yields the rate of growth. Here the slope is calculated with a linear fit obtained by a Least-Squares approximation considering the time interval [0.2,0.5]. Same interval has been used by Daly and Queutey.

The dimensionless slope $n^* = n\left(\frac{\nu}{g^2}\right)^{\frac{1}{3}}$ is plotted over the Reynolds number. Figure 11.3 shows the results of the described solver, Chandrasekhar, Daly and Queutey. The results are in overall very good agreement. It has to be mentioned that the linear fit behaves sensible to the considered time interval and the differences between all curves lie in the uncertainty of this sensibility. Therefore all numerical results could be interpreted as equally accurate.

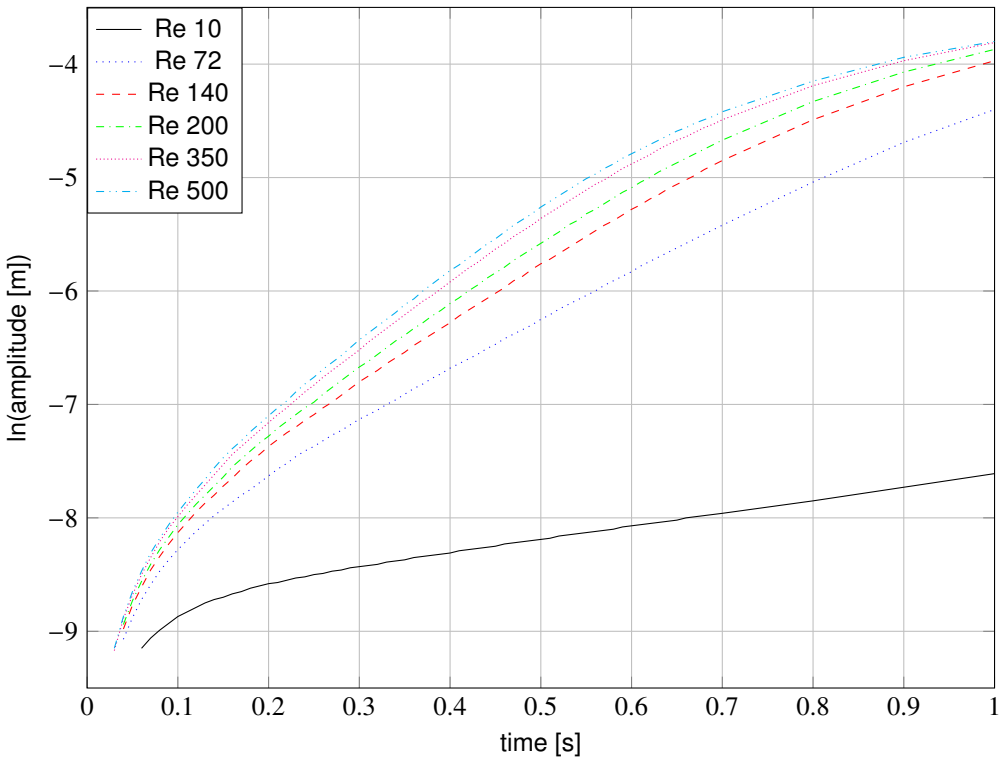


Figure 11.2.: Rayleigh-Taylor instability: Logarithm of the wave amplitude over time

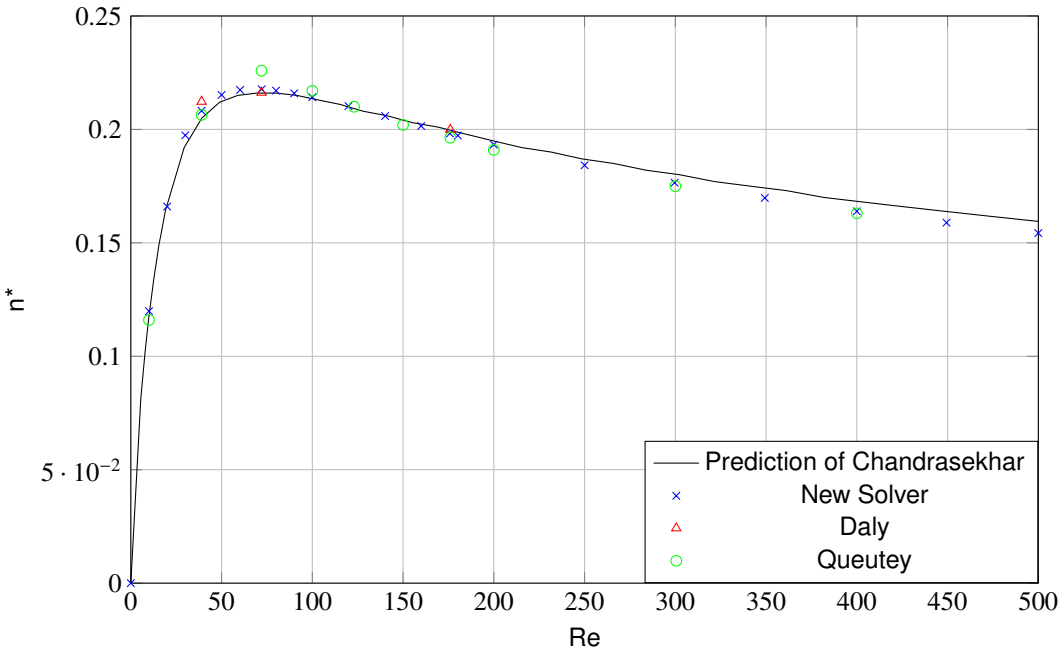


Figure 11.3.: Rayleigh-Taylor instability: Dimensionless rate of growth n^* over the Reynolds number R_e

11.2. Sysser60

In order to compare our new solver *NSTFoam* with *interFoam* (of OpenFOAM 2.4.x) and the commercial solver *Star-CCM+* v9.06 the Sysser 60 test case of the *Delft Systematic Yacht Hull Series* of Delft University of Technology 2013 has been chosen. The simulations are carried out without motion to better distinguish between phenomena induced by flow solver and motion solver. Particular focus during this test was on the phenomenon of numerical ventilation. To evaluate the solvers capabilities in this regard a flotation with the bow knuckle above water was chosen. The simulation is done in model scale. The hull has a waterline length of $L = 2.16m$. The flow speed is 1.806 m/s corresponding to a Froude no. of 0.39. The hull is heeled by $20°$ and yawed by $3°$.

Figure 11.4 to 11.6 show the grid. It was generated with snappyHexMesh (SHM) and has 2.7 million cells. The waterline was refined anisotropically in the z direction. The cells at the outlet were stretched to avoid reflections of the waves. Figure 11.4 only shows a part of these stretched cells. The domain has a length of $3L$ in front of the hull and $3L$ behind the hull before stretching the cells at the outlet. The height is $1.5L$ below waterline and $1.0L$ above waterline. In the area of the Kelvin wave pattern additional refinement is applied up to a length of $1.4L$ behind the hull. The domain is $4.3L$ wide. The grid was exported to Star-CCM+ so that all simulations are done on exactly the same grid.

11.2.1. Solver Setup

The interFoam solver uses the same algorithm as shown in Fig. 8.1. For the VOF-method an additional compressive term is used to keep a sharp interface. The equation is solved using the MULES algorithm as briefly described in 5.2. Despite a lot of time consuming tests it was not possible to get a stable setup for the given grid with the original interFoam solver. Therefore a function for the limitation of the velocity was added. Each time the velocity changes, the cell velocities are limited to a user specified value. In this case the velocity is limited to $15\frac{m}{s}$. In fact this modification harms the conversation of mass. But tests have shown that this has no influence onto the results. For clarity this solver is called *interFoamMod* in this Chapter. The Volume-of-Fluid implementation of interFoam allows to use the α-subcycling technique. This method allows the subdivision of the timestep in smaller subtimesteps for the solution of the VOF-transport equation which improves the stability and accuracy.

Star-CCM+ also uses an algorithm similar to Fig. 8.1. It is not known if any additional iterations for the correction of mesh non-orthogonality are applied. Also, it is not possible to prescribe any kind of α-subcycling. The HRIC scheme is used for the convective term of the VOF-equation. To supress numerical ventilation the blend to upwind differencing was switched off as recommended in Böhm 2014 for steady state simulations.

All three solvers are adjusted to the same settings as close as possible, see Tab. 11.1. The goal was to use a time step of $\Delta t = 0.02s$, 5 outer iterations, 1 Piso iteration, 1 additional iteration for the correction of non-orthogonality and no alpha-subcycling. The averaged CFL is 0.123. Although the modified solver interFoamMod was used, additional Piso iterations and α-subcycles were necessary to get sufficient solver stability here. For the sake of clarity, this is only done to get solver stability,

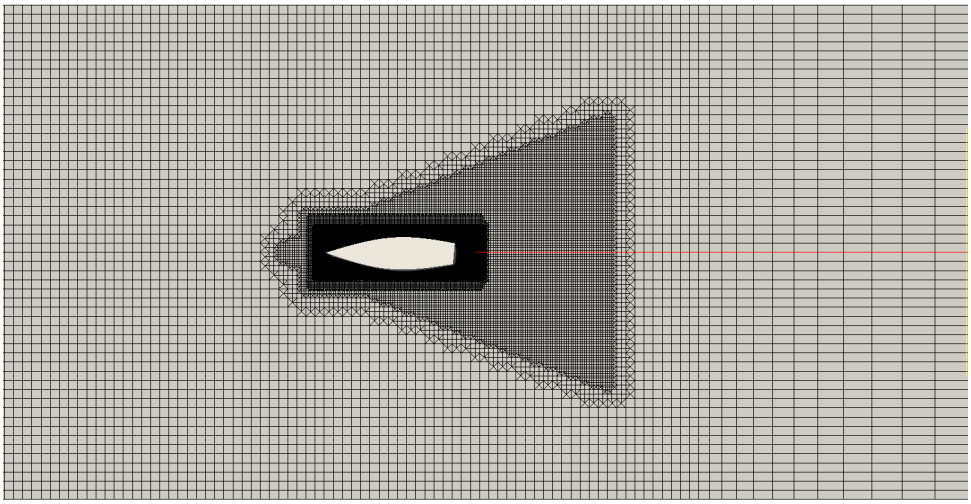


Figure 11.4.: Sysser60 grid top view

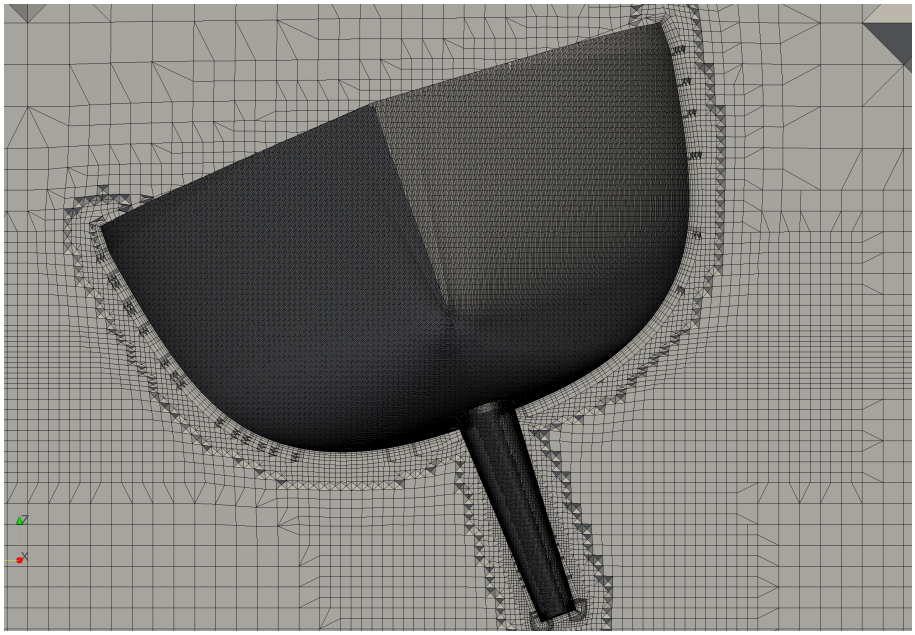


Figure 11.5.: Sysser60 grid front view

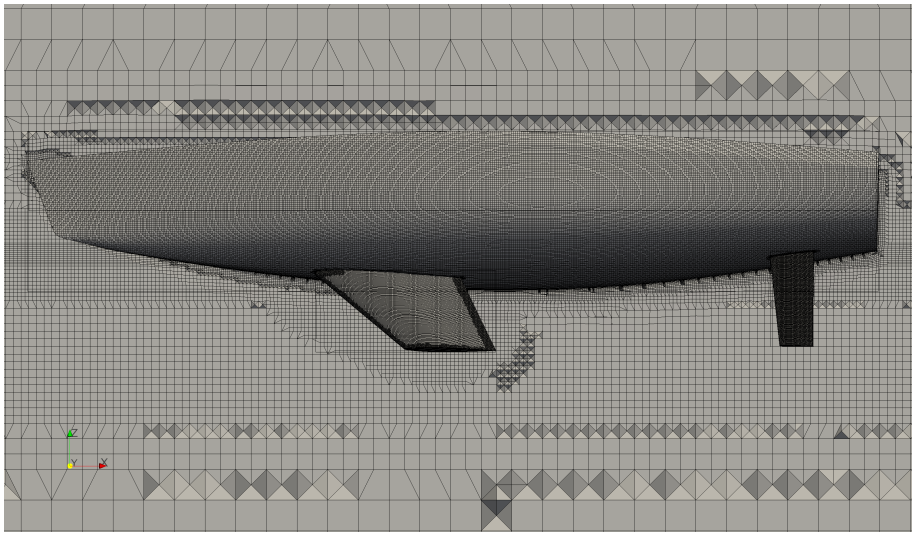


Figure 11.6.: Sysser60 grid side view

nevertheless additional iterations and subcycles generally have a positive influence onto the quality of the result.

In all simulations a k-ω-SST turbulence model was chosen. A total time of $30s$ is simulated.

Star-CCM+ is using some unknown method to suppress an unphysical splash wave at the beginning of the simulation. For NSTFoam the velocity is ramped over the first $0.2s$ to avoid the same unphysical behavior. The simulations were also done without ramping, showing that the ramping is not required for solver stability in this particular case.

11.2.2. Wave Pattern

The transversal wavecuts of Fig. 11.7 are obtained by cutting the free-surface with an increment of $\frac{x}{L} = 0.1L$. The longitudinal wavecuts of Fig. 11.8 are obtained by cutting with an increment of $\frac{y}{L} = 0.05L$. The wavecuts show good agreement in the region close to the hull. At the bow only the Star-CCM+ simulation produces a different result with three breaking waves which is discussed later in this Section. Both figures show that the waves of the interFoamMod simulation are increasingly damped the further from the hull. Other simulations (not shown here) with a much longer Kelvin refinement of about $6L$ behind the hull prove this behavior. Independent of the refinement, the interFoamMod waves are much flatter far from the hull. This emphasizes that the discretization of the convective term of (2.3) requires a high-resolution scheme for a correct convection of the waves instead of using an additional compression term. Considering simulations including sea waves this is an important point to avoid damping of these waves. Figure 11.9 also illustrates the damping characteristics of the interFoamMod simulation. It is an important point that the interFoamMod simulation is using five alpha-subcycles due to stability reasons, whereas both other solvers do not use any subcycles. Subcycling supports the accuracy of the simulation for both, wave sharpness and wave convection. Despite this, interFoamMod is less accurate.

The waves at the bow of the Star-CCM+ simulation are significantly different than the results of NSTFoam and interFoamMod. Figure 11.10 shows that Star-CCM+ generates a wave breaking three times. The other two solvers have one to one and a half breaking waves depending on the chosen timestep. Increasing the cell size to 11 million cells has no effect on this behavior and the wave count is still the same for NSTFoam and interFoamMod. Using the finer grid for Star-CCM+ boosts this behavior to four clearly breaking waves. As Star-CCM+ is a closed code, the implemented models might be more advanced than described in the theory guide. One possible explanation might be a more accurate treatment of the viscosity jump, as discussed in the end of Chapter 9.

11.2.3. Numerical Ventilation

The amount of *numerical ventilation* (NV) is shown in Fig. 11.11 plotting the volume fraction on the hull. For a distinct differentiation the volume fraction is scaled from 0.85 to unity. While the results of NSTFoam and Star-CCM+ are quite similar, the interFoamMod simulation produced much more air under the hull. Figure 11.11 is not sufficient for an reliable predication of the numerical ventilation, as the NV changes with time due to a periodic oscillation of the flow and the chosen timestep might have a significant influence on the result.

To estimate the NV the following equation is introduced.

$$N_\mathrm{v} = \frac{\sum_{f=1}^{n} \lambda_f |\mathbf{s}_f||(1-\alpha_w f)|}{\sum_{f=1}^{n} \lambda_f |\mathbf{s}_f|} \quad , \tag{11.9}$$

where $\mathbf{s}$ is the face area vector of the face f, n is the number of faces defining the hull patch, α_w is the volume fraction of the water and λ is a switch depending on the pressure

$$\lambda_f = \begin{cases} 1 & \text{if } p_f > p_\mathrm{l} \\ 0 & \text{if } p_f \leq p_\mathrm{l} \end{cases} \quad . \tag{11.10}$$

The limiting pressure p_l is set to $150\frac{N}{m^2}$. The pressure switch excludes faces above and at the free surface, respectively. Therefore physically correct spray does not have a negative effect on the assessment of the NV. This equation does not produce an absolutely correct evaluation but allows a rough approximation to evaluate the results.

Figure 11.12 shows the NV plotted over time. It shows that the result of interFoamMod has twice the NV compared to NSTFoam and Star-CCM+.

11.2.4. Velocity Overshoots

Figure 11.13 shows the maximum velocities for each timestep. The maximum velocity of the interFoamMod simulation is mostly at the prescribed limit of 15m/s. The results of NSTFoam and Star-CCM+ are physically more correct. NSTFoam only reaches the limiting velocity for the first timesteps. Star-CCM+ does not have exactly the same limitation and only exceeds this value for the first timesteps and at $t = 16\mathrm{s}$ (Nevertheless Star-CCM+ is also damping the growths of very high velocities). Overall the

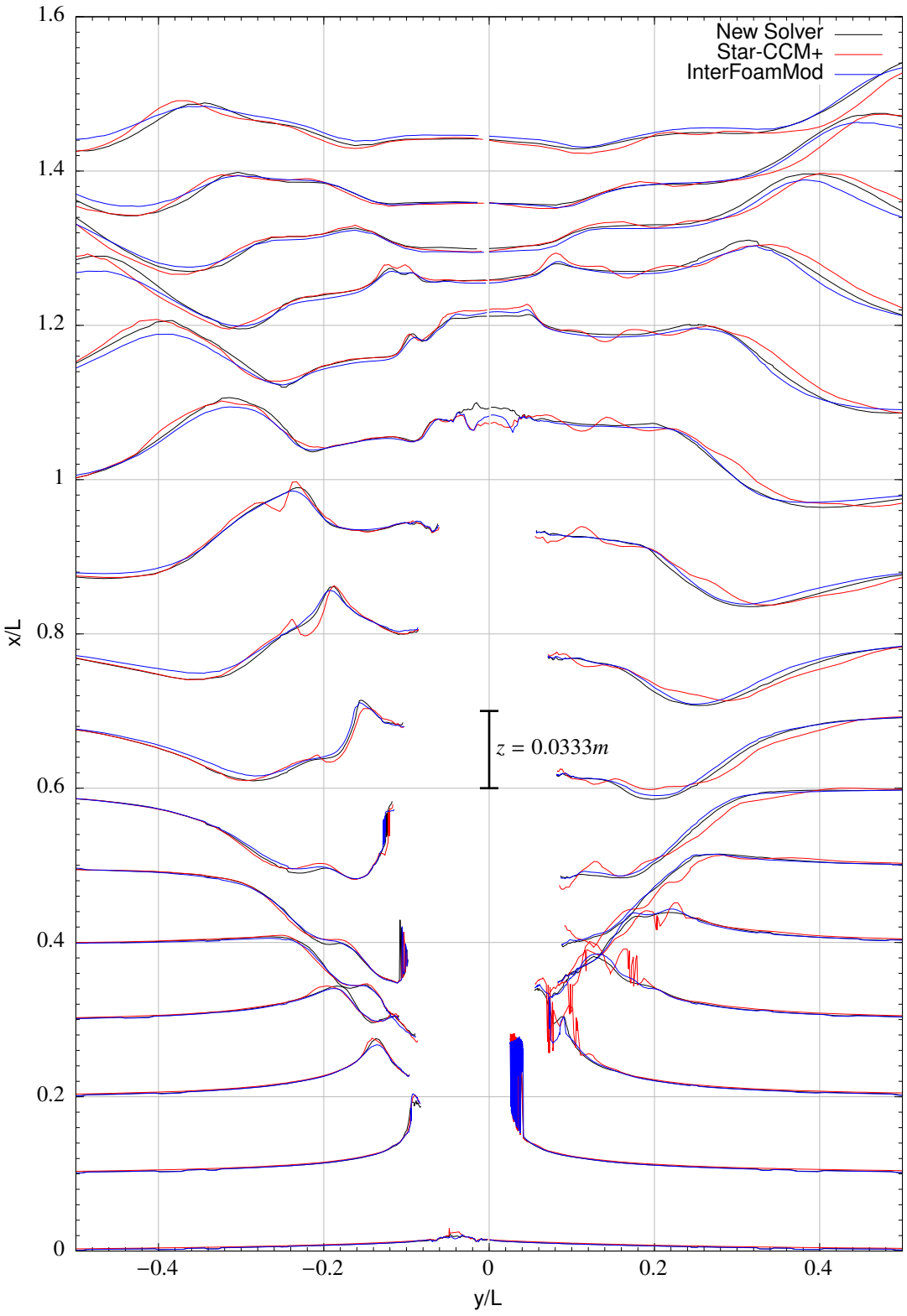


Figure 11.7.: Sysser60: transversal wavecuts

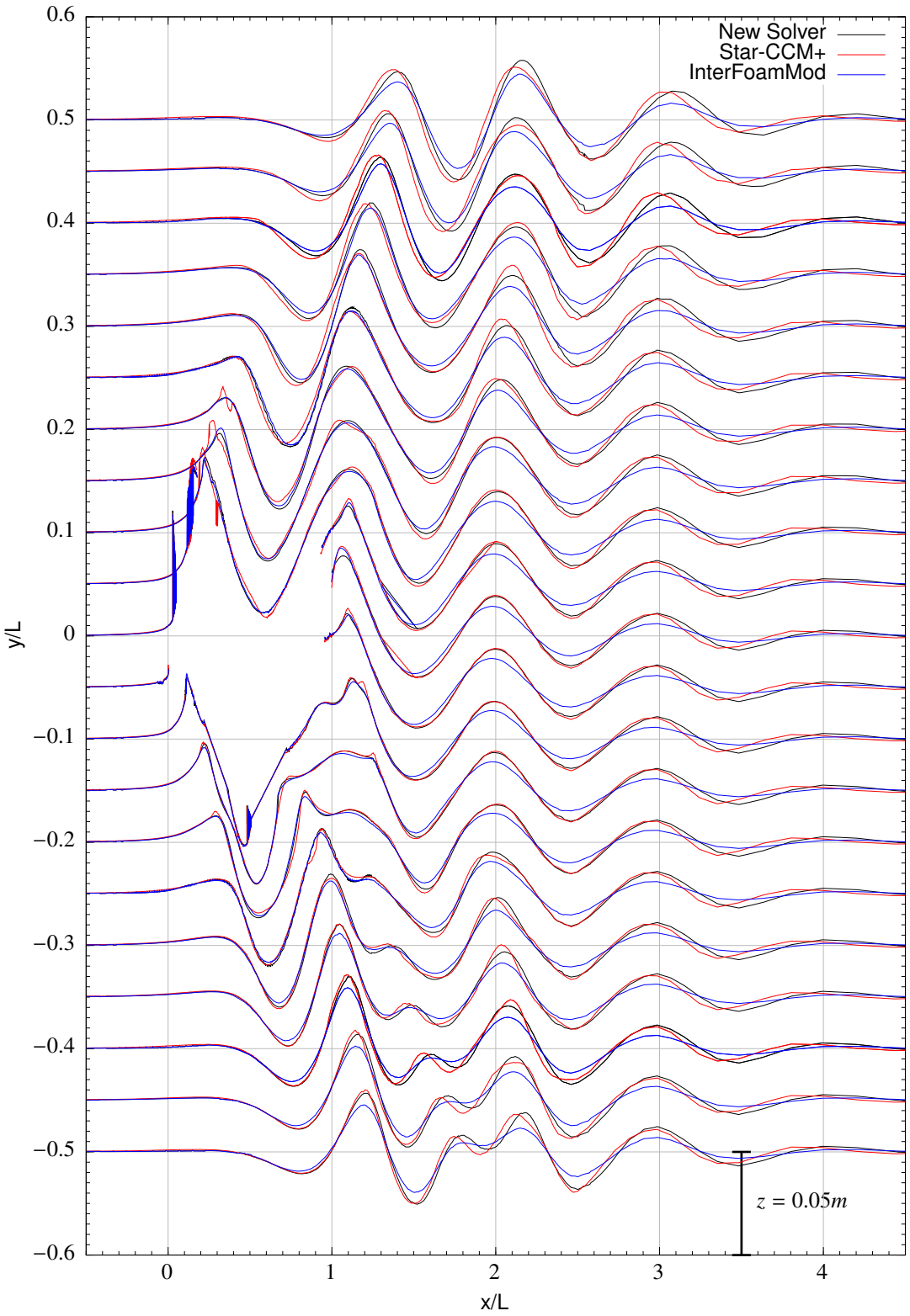
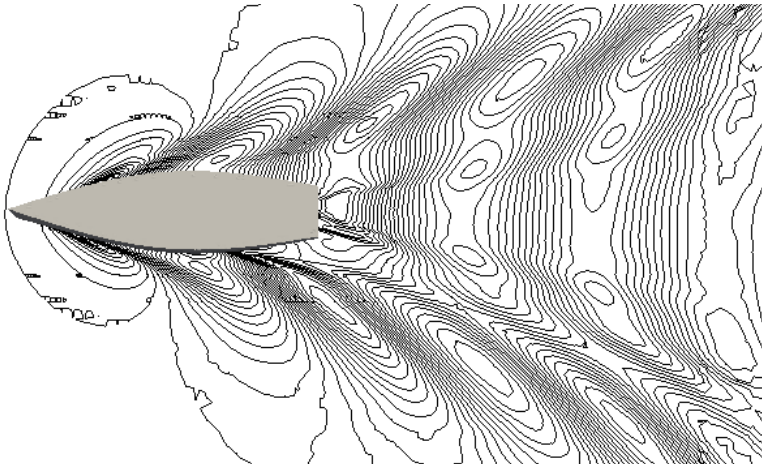
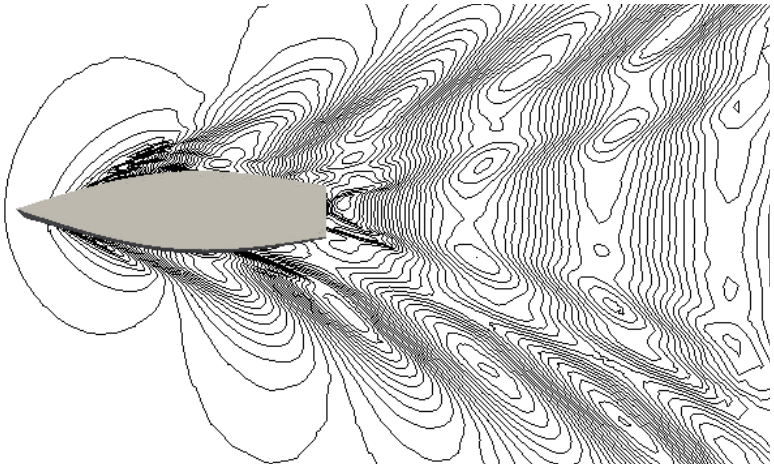


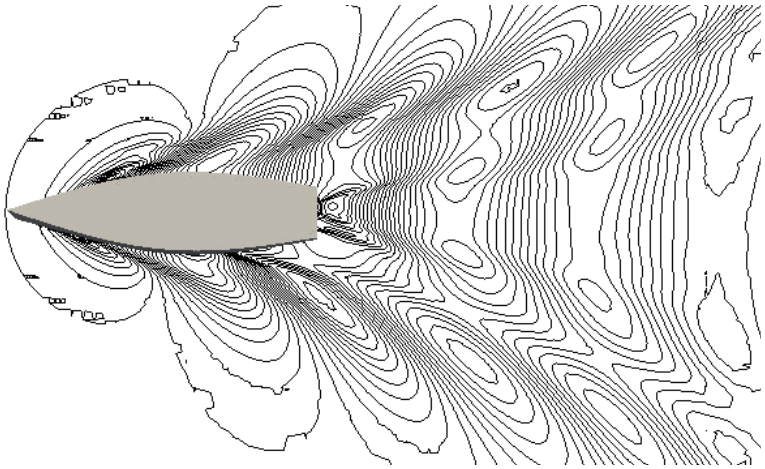
Figure 11.8.: Sysser60: longitudinal wavecuts



(a) NSTFoam

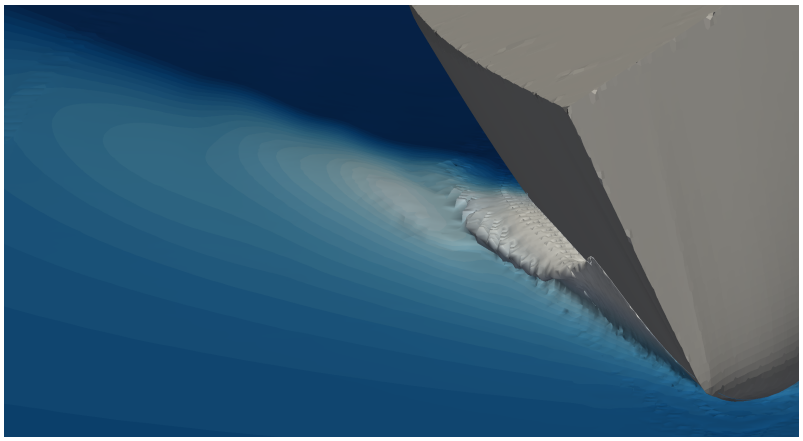


(b) Star-CCM+

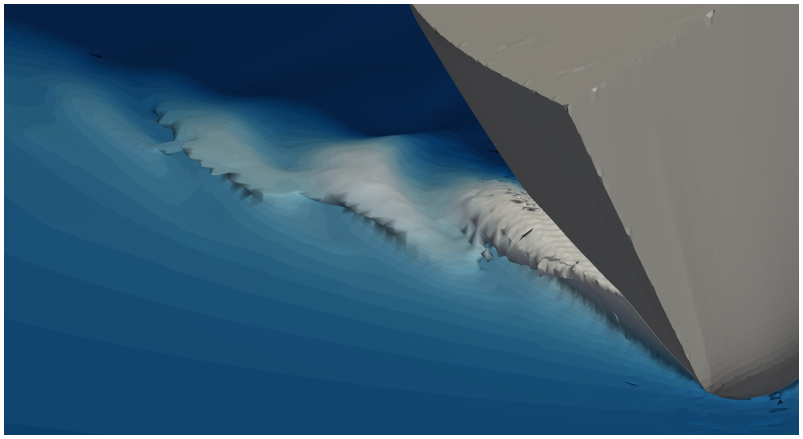


(c) interFoamMod

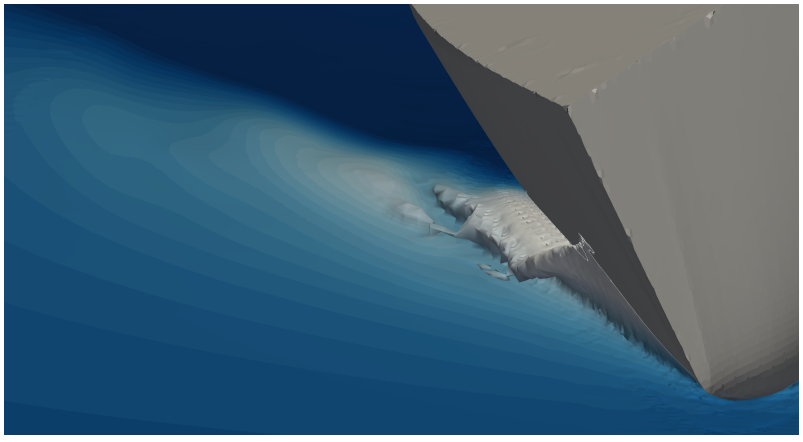
Figure 11.9.: Wavepattern of the Sysser60 test case



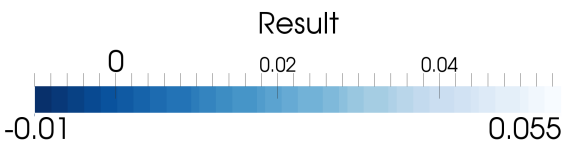
(a) NSTFoam



(b) Star-CCM+

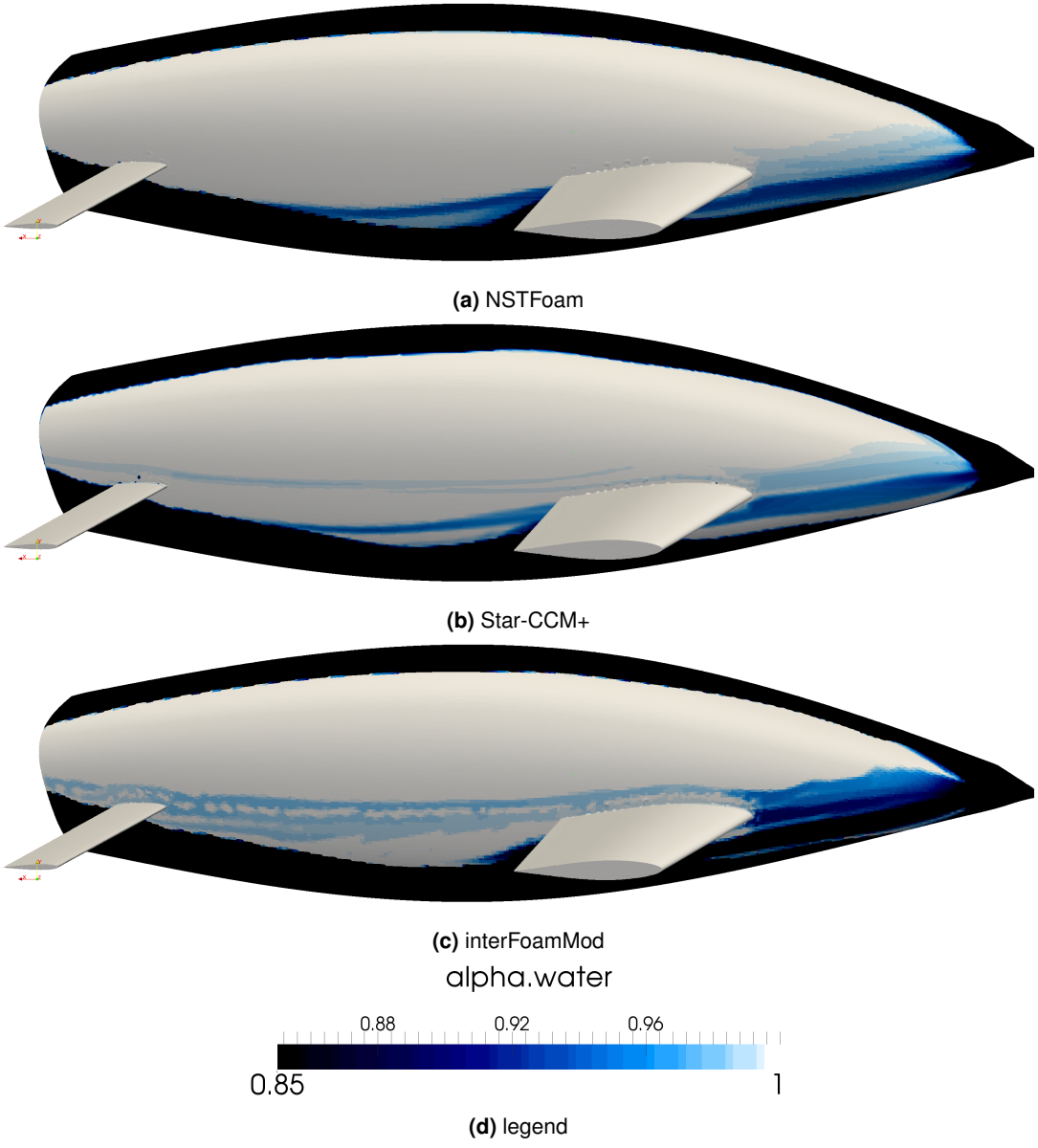


(c) interFoamMod



(d) legend

Figure 11.10.: Waves at the bow of the Sysser60 test case



(a) NSTFoam

(b) Star-CCM+

(c) interFoamMod

(d) legend

Figure 11.11.: Numerical ventilation of the Sysser60 test case

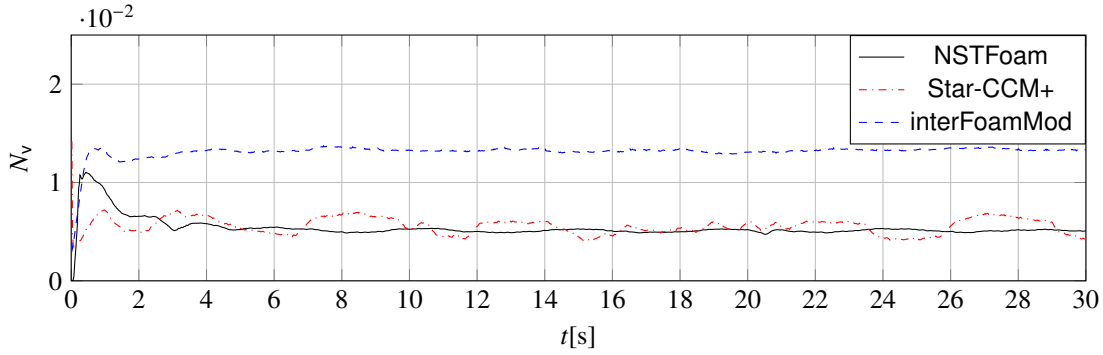


Figure 11.12.: Numerical ventilation

maximal velocity of Star-CCM+ is smaller than for NSTFoam.

Concluding it can be said, that the velocities of both solvers lie in a reasonable range, whereas the interFoamMod velocities have unacceptably high values and never drop down to plausible values. For the sake of clarity, the used velocity limitation is not included in the original interFoam solver and without this limitation the velocities would reach much higher values and the solution will diverge until the solver crashes in the first timesteps.

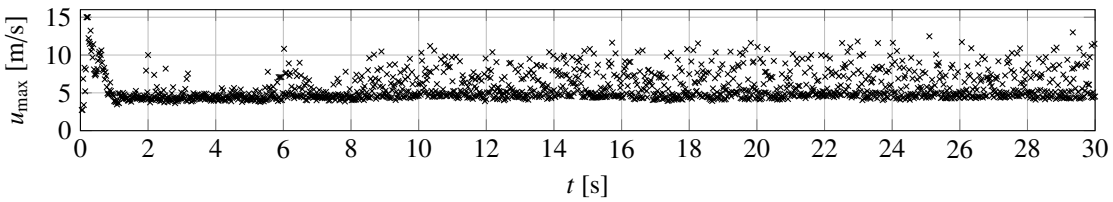
The simulation with NSTFoam has been done with and without pressure reconstruction. Without pressure reconstruction the maximal velocities have the same characteristic as the velocities of the interFoamMod simulation. Apart from that, the result looks almost identical to the result of the simulation with reconstruction. Thus, in this case the reconstruction is only necessary for the stability of the solver. It delivers a physically correct solution of the velocity overshoot problem compared to a hard limitation. Additionally it does not require a user-specified value. This simplifies automated set-ups and more complex cases.

11.2.5. Computation Time

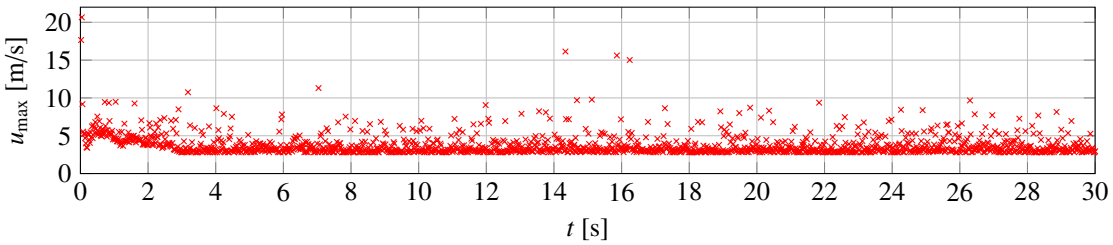
For the comparison of the computational time the case was varied in cell size and time step. The computations were done on a node with 16 cores at 2.6Ghz. Only 15 cores were used to have enough power left for background processes of the machine. Different nodes with identical construction have been used for all simulations. Overall, this is not a scientific investigation on the computational time and the time needed for the same simulation might vary about 20%. Still, the results given in Tab. 11.1 show a significant trend. The table demonstrates the solver capabilities for real engineering situations, where grid or timestep studies are required.

For all grids calculated with the smallest timestep of $\Delta t = 0.02$ NSTFoam is about one third slower than Star-CCM+. For the timestep variation NSTFoam needs about twice the time of Star-CCM+. The reason could be the additional non-orthogonal correction iteration and the settings for the absolute residual, which breaks the inner iterations of the linear solvers.

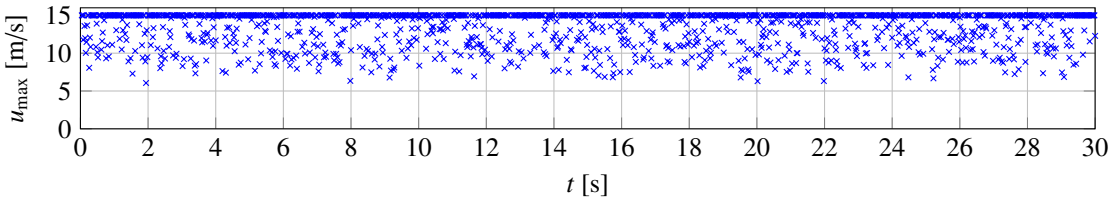
Comment of the author:
In later simulations the additional non-orthogonal correction iteration has not been applied anymore. The final, quasi steady results show to be of same accuracy while the simulation time could be reduced



(a) NSTFoam



(b) Star-CCM+



(c) interFoamMod

Figure 11.13.: Velocity overshoots

Table 11.1.: Simulation time on 15*2.6GHz for different case setups

	Δt	nCells	θ	κ	ξ	ϵ	Computation Time	Deviation to Star-CCM+
	[s]	[10^6 cells]	[ ]	[ ]	[ ]	[ ]	[h]	[%]
Star-CCM+	0.02	2.7	5	1	1	-	8.39	0.0
NSTFoam	0.02	2.7	5	1	1	1	10.71	+27
interFoamMod	0.02	2.7	5	3	5	1	27.95	+233
Star-CCM+	0.04	2.7	5	1	1	-	2.68	0.0
NSTFoam	0.04	2.7	5	1	1	1	5.92	+121
interFoamMod	0.04	2.7	5	3	10	1	22.53	+772
Star-CCM+	0.06	2.7	5	1	1	-	1.77	0.0
NSTFoam	0.06	2.7	5	1	1	1	2.92	+64
interFoamMod	0.06	2.7	10	3	15	1	42.51	+2301
Star-CCM+	0.08	2.7	5	1	1	-	1.24	0.0
NSTFoam	0.08	2.7	5	1	1	1	2.41	+94
interFoamMod	0.08	2.7	10	3	20	1	42.07	+3392
Star-CCM+	0.02	4.5	5	1	1	-	14.91	0.0
NSTFoam	0.02	4.5	5	1	1	1	18.91	+26
interFoamMod	0.02	4.5	5	3	5	1	48.23	+223
Star-CCM+	0.02	8.3	5	1	1	-	29.94	0.0
NSTFoam	0.02	8.3	5	1	1	1	40.21	+34
interFoamMod	0.01	8.3	5	3	5	1	213.12	+611

θ = number of outer iterations, κ = number of Piso corrector iterations,
ξ = number of α subcycles, ϵ = number of non-orthogonal correction iterations

in mean by 25%.

To facilitate the calculation with the interFoamMod solver its settings had to be adjusted. These adjustments are the major reason for the significant slower computations. For the grid variation the solver needs two to six times the Star-CCM+ calculation time. For the time step variation it needs up to 33 times of the calculation time. Increasing the timestep should decrease the calculation time. The cases with $\Delta t = 0.06$ and $\Delta t = 0.08$ show that this is not always the case for interFoamMod just because of stability reasons. Additionally it has to be emphasized that a typical grid or timestep study is based on scripting the simulations and then running all in one chunk. Here a lot of failing runs were necessary to get a setup with acceptable calculation time for each variation. Of course this is an unacceptable workflow. The 8.3 million cells test case forced a decrease in the timestep to $\Delta t = 0.01$. A grid with 10.8 million cells has been tested, also. NSTFoam and Star-CCM+ run as stable as before without modifying any settings. For interFoamMod the settings were adjusted until a simulation time of 2 weeks was reached, but the simulation still crashed.

11.2.6. Conclusion

The presented results clearly show the improved quality of the new solver NSTFoam compared to interFoam. Furthermore, it is shown that the accuracy is almost the same as of Star-CCM+. Problems, like arbitrary instabilities, velocity overshoots, numerical ventilation and damped waves were successfully remedied.

11.3. SYRF Wide and Light Study

This section uses the Wide and Light Study from the *Sailing Yacht Research Foundation* (Claughton 2015b) to make a quantitative validation for a test matrix close to industrial application. In this study five different stakeholders carried out "blind" CFD analysis on an identical test matrix using different computational codes and approaches. The same test matrix was run as a tank test for both canoe body only and appended (hell, keel, rudder) configurations. The CFD results were compared with the tank test control results to determine CFD model accuracy. The investigated hull is a modern, *wide* and *light* design. Both, CFD simulations and tank tests were done in model scale with free heave and pitch. To guarantee a good comparability no sailforce model is applied for the CFD simulations. Instead, the center of gravity is varied in the given test matrix to achieve the wished torque. The test matrix contains 120 cases, from which 35 cases where picked for the here presented investigation. The measurements of the yacht model are given in Tab. 11.2.

Table 11.3 shows the five CFD contributors, which are all well known in the yacht branch and the used CFD codes. For the following comparison, only the results of FINE/Marine, Star-CCM+ and OpenFOAM, which are pure RANSE simulations are compared to NSTFoam. The first two, delivered the most accurate results in the Wide and Light Study. The OpenFOAM results are used to point out the issues in the application of original OpenFOAM.

11.3.1. Simulation Setup

As mentioned in Chapter 1.5 the here developed solver NSTFoam is part of a very automated CFD tool, also developed by the author. The tool, requires the test matrix parameters, the geometries as a STL-file and some main measurements, like the length over surface of the individual yacht parts (hull, keel, rudder, keel-bomb). After defining this input, the complete grid setup, meshing, solving and post-processing is running fully automated. The solver NSTFoam uses a stopping criterion for the simulations. If the variation for the drag forces are small and the drag fulfills several stopping criteria the

Table 11.2.: Wide and Light Study: Measurements of the yacht model

Overall length	4.88m
Design waterline length	4.60m
Displacement (appended)	215kg
Displacement (canoe body)	197kg
Maximum beam	1.28m
Draft to datum	1.15m

Table 11.3.: Wide and Light Study: CFD Contributors

Software	Contributor	Affiliation	Type
FINE/Marine	Benoit Mallol, Jason Ker	Numeca, Ker Yacht Design	RANSE
Star-CCM+	Rodrigo Azcueta, Matteo Lledri	Cape Horn Engineering	RANSE
OpenFOAM 2.0.1	Sandy Wright	Wolfson Unit MTIA, University of Southampton	RANSE
Shipflow	Lars Larsson, Michal Orych	Chalmers University of Technology, FLOWTECH Int. AB	Combination of Panel Code and RANSE
FlowLogic	David Egan		Panel Code

simulation is stopped to save computational costs. In the current version of the automation tool, the focus of the setup lies on Froude numbers between 0.2 and 0.5. Lower or higher values are possible, but have not been treated with special care during development. The here presented results are achieved using this automated setup and the pictures of Fig. 11.14 to 11.17 were taken from its output.

Table 11.4 summarizes the important parts of the simulation setups. Except original OpenFOAM, all setups uses a similar grid resolution with cell numbers between 0.9 and 6.5 million cells. All setups are using some kind of k-ω-SST model with wall functions. Furthermore, they are all using the Volume-of-Fluid method to model the free-surface.

At this point, it has to be noted, that the yacht model is not optimal for the application of wall-functions due to its short length. An own y+ study has shown that the wall functions of OpenFOAM are working good in the range of 25 to 250. Therefore the minimal cell size is limited by a y+ value of 25, when using wall functions. In the case of the wide and light yacht, it might be interesting to refine the cells even more to perfectly resolve the pressure. The only valid solution doing this, would be to disable the wall functions and to use y+ values below 1. As this is typically not done in industrial applications, this was not done here, too. It was decided to target for a minimal y+ of 25 and no finer pressure resolution. Especially for the small parts like the rudder and the keel, this leads to somewhat less than perfect pressure resolution.

Furthermore, this phenomenon prevents making a reliable grid study. A finer resolution might improve the pressure forces but it will also impair the shear stress forces as the wall functions leaves their optimal working region.

11.3.2. Results

Figure 11.14 shows the used grid and prism layers on the basis of case 86. All important grid features, like the length of the kelvin refinement, prism layer sizes, domain size or the free-surface refinement

Table 11.4.: Wide and Light Study: Overview of the simulation setups

Software	number of cells (asymmetric)	y+	wall functions	turbulence model	free-surface method
FINE/Marine	2.8×10^6 (unappended) 5.4×10^6 (appended)	50	yes	k-ω-SST	VOF
Star-CCM+	1.7×10^6 (unappended) 2.4×10^6 (appended)	26-40	yes	k-ω-SST	VOF
OpenFOAM	15×10^6 (unappended) 19×10^6 (appended)	20-120	yes	k-ω-SST	VOF
NSTFoam	0.9×10^6 (unappended) 6.5×10^6 (appended)	25-120	yes	k-ω-SST	VOF

are adjusted by the automated setup with respect to case parameters like the Froude number. Figure 11.15 shows the y+, pressure and shear stress behavior on the yacht surface. The y+ values lie in the desired area, also the pressure and shear stress behavior is plausible. Figure 11.16 shows the numerical ventilation on the bottom of the yacht and the free-surface from the top view. The numerical ventilation is sufficient low, the free-surface is without noticeable problems. Figure 11.17 shows the free-surface from different viewing angles close to the yacht. All views are without noticeable problems. The surface itself is in alignment with the results presented in the wide and light study report (Claughton 2015a).

The complete results of the investigated test matrix are given in Tab. 11.5. The table shows the CFD results of Star-CCM+, FINE/Marine, NSTFoam and original OpenFoam. Additionally it shows the tank results and the deviation of the CFD Simulations to the tank results. Of course, the tank results are not absolutely correct, nevertheless they will be used as the reference results in the given comparison. Cases 1 to 45 uses the canoe body only, cases 46 to 95 uses the hull appended with the keel, keel bomb and a single rudder. In cases 91 to 95 the angle of the rudder is varied. For all results, the flow always points in the direction of the x-axis. Hence, Fx represents the drag and Fy the lift forces.

Cases 1,3,5 and 9 investigate the upright resistance of the canoe body for different Froude numbers. Figure 11.18 shows the corresponding trend graphically. With respect to the tank results, Star-CCM+ seems to deliver the best results. NSTFoam has bigger deviations for the two smallest Froude numbers, especially for Fn = 0.1 the deviation of -19.6% is high. Still this is much better than original OpenFOAM with a deviation of 115.8%, which seems to be an outlier. Furthermore, Tab. 11.5 shows, that Star-CCM+, FINE/Marine and NSTFoam have the same tendency in the ΔFx trend. Based on this, one can not exclude that the tank results deliver a relatively higher drag for the lower Froude numbers. Except for Fn=0.1, which is not in the design area of the automation tool, the results of NSTFoam are as good as the results of the two other commercial solvers.

Cases 21, 23 and 25 investigate a variation of the longitudinal center of gravity for the upright canoe

body at constant speed. The trend is shown graphically in Fig. 11.19. The results of Star-CCM+, FINE/Marine and NSTFoam seem to be equally good, whereby NSTFoam has the least deviations to the tank results. Again, Tab. 11.5 shows that the deviations to the tank results have the same trend for all CFD solvers.

Cases 26, 28 and 29 investigate the lift and drag behavior of the heeled canoe body for different Froude numbers, see Fig. 11.20. Overall, all four codes deliver good results for the drag. The drag results of original OpenFOAM are remarkable close to the tank results. However, the lift differences vary a lot for all four solvers. Original OpenFOAM has unacceptable high differences between 33% and 127%. Also, the Star-CCM+ results have an outlier of 75% difference for the lowest Froude number. FINE/Marine has the smallest deviations from -9.3% to 9.5%, indeed, NSTFoam has a similar range of variation with differences from -15.5% to 0.2%.

Cases 36, 38 and 40 investigate the lift and drag behavior of the heeled canoe body at a constant speed of Fn=0.35 for different yaw angles, see Fig. 11.21. Again, all four codes deliver good results for the drag. Except original OpenFOAM, which was not able to deliver a converging result for one case. As in the previous cases, the lift differences are varying a lot. Star-CCM+, FINE/Marine and original OpenFOAM are far away from the tank results. NSTFoam has the smallest differences with values between -8.7% and 14.8%.

Cases 41, 43 and 45 investigate the lift and drag behavior of the heeled canoe body at a constant speed of Fn=0.5 for different yaw angles, see Fig. 11.22. All four codes deliver equally good results for the drag. For the lift, only original OpenFOAM has higher deviations up to 17.8%, which are still quite acceptable.

The following cases 46 to 85 investigate the same parameters as in the previous ones but for the fully appended yacht.

Cases 46, 48, 50 and 54 investigate the upright resistance of the fully appended yacht for different Froude numbers, see Fig. 11.23. Only FINE/Marine and NSTFoam deliver overall good results for the drag. Star-CCM+ has too high values for the lower Froude numbers. This is very interesting, as for the same cases in the canoe body investigation, only Star-CCM+ delivered values, which were not too low. The drag results from original OpenFOAM are far away from the tank results with deviation from 22.9% to 61.3%.

Cases 61, 63 and 65 investigate a variation of the longitudinal center of gravity for the upright and fully appended yacht at constant speed, see Fig. 11.24. Except original OpenFOAM, all solver deliver very good results for the drag.

Cases 71, 73 and 75 investigate the lift and drag behavior of the heeled and fully appended yacht for different Froude numbers, see Fig. 11.25. As in the previous investigation, all solvers except original OpenFOAM, deliver very good results for the drag. Original OpenFOAM has a big gap to the tank results with deviations between 28.4% and 38.3%. No lift results are given for the tank tests. Star-CCM+

and original OpenFOAM deliver nearly the same values for the lift. FINE/Marine has big differences to them. NSTFoam is lying somewhere in the middle except for the highest Froude number, where NSTFoam seems to have an outlier, as the overall lift trend is not in alignment with the other three solvers (see Fig. 11.25).

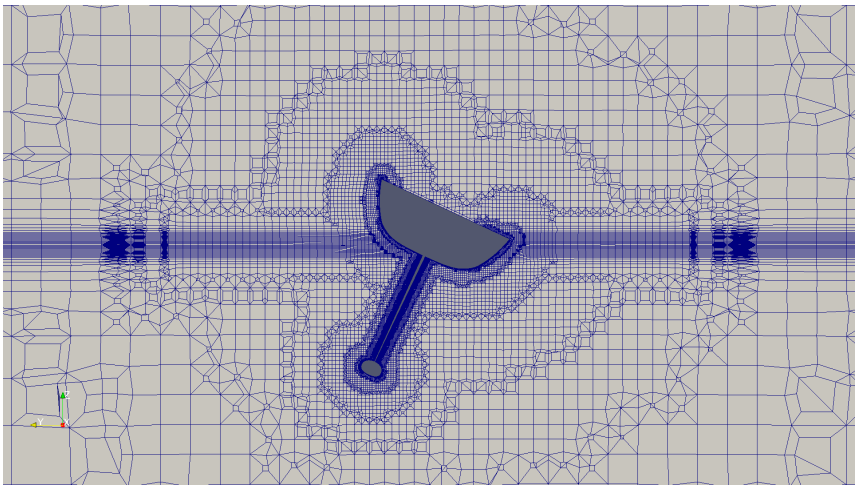
Cases 81, 83 and 85 investigate the lift and drag behavior of the heeled and fully appended yacht at a constant speed of Fn=0.35 for different yaw angles, see Fig. 11.26. Star-CCM+, FINE/Marine and NSTFoam deliver good results for the drag and lift. FINE/Marine has slightly higher deviations for the lift. Original OpenFOAM has high deviations for the drag of approximately 40% and acceptable values for the lift.

Cases 86, 88 and 90 investigate the lift and drag behavior of the heeled and fully appended yacht at a constant speed of Fn=0.5 for different yaw angles, see Fig. 11.27. For the drag, Star-CCM+, FINE/Marine and NSTFoam give good results. Original OpenFOAM delivers high deviations of approximately 20% for the drag for two cases. For one case original OpenFOAM is not able to give a solution. For the lift, all four solvers have higher deviations to the tank results. Indeed, the trend of these deviation is very similar for Star-CCM+, FINE/Marine and NSTFoam. According to the table, the tank results might be inaccurate for case 88.

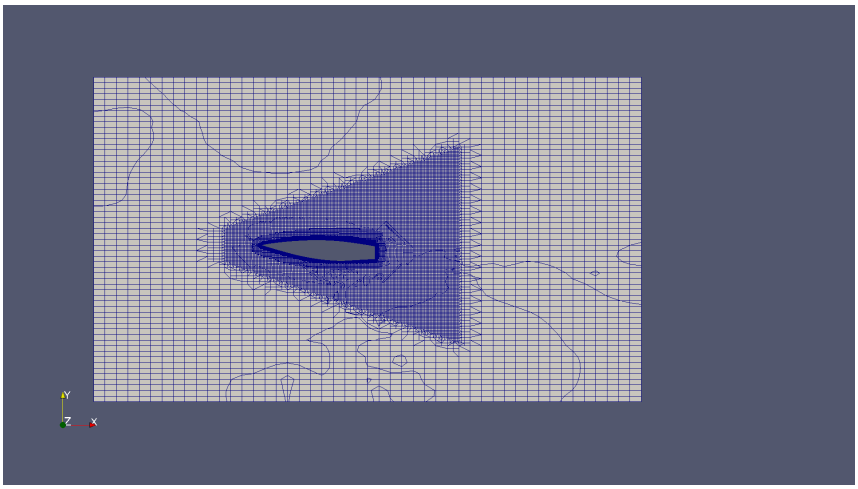
Cases 91, 94 and 95 investigate the lift and drag behavior of the heeled and fully appended yacht at constant speed for different rudder angles, see Fig. 11.28. Except original OpenFOAM, all solvers give very good results for the drag. For the lift all four solver have equal results with the same trend for the deviations. For case 91 with a rudder angle of -2.0° the differences are small and the deviations of the solvers lie between -2.6% and 3.4%. For cases 95 and 94 the differences are bigger and the results have deviations from -10.4% to -19.1%. This might be an indicator for wrong tank results. On the other hand, it should be noted again, that all CFD simulations are done with wall functions, which are problematic especially at the rudder due to the small model scale. Therefore the pressure gradient might be under-resolved at the rudder, which might lead to such underestimation of the lift.

Besides the discussion of the pure quantities it is important to see if the results fulfill the overall trend. As shown in figures 11.18 to 11.28 this is the case for Star-CCM+, FINE/Marine and NSTFoam. The results also clearly show that NSTFoam is performing on the same level as its commercial competitors. On the other hand, for original OpenFOAM, the results underline its unreliability. This is also confirmed by the contributor of the results, the Wolfson Unit, emphasizing the poor cost effectiveness.

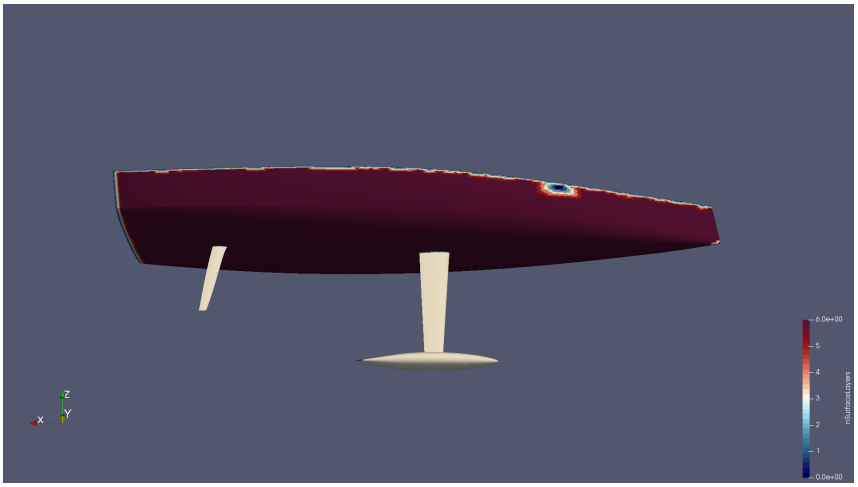
Star-CCM+, FINE/Marine and the newly developed NSTFoam are all performing very well for the drag prediction. For lift, the differences are bigger, but still very usable for industrial applications. Nevertheless all codes deliver at least one outlier, where the quantitative result was definitively wrong. Last but not least, this validations shows that NSTFoam reaches the quality and reliability of state-of-the-art commercial RANSE solvers.



(a) Grid: X-Plane

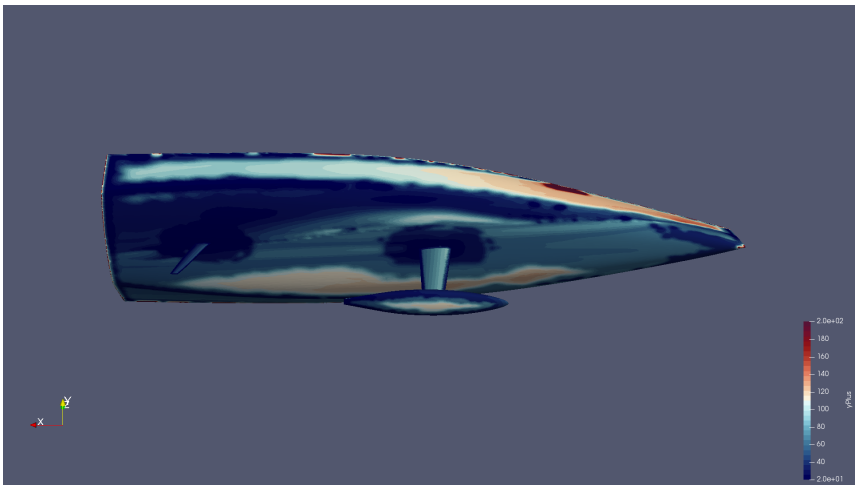


(b) Grid: Z-Plane

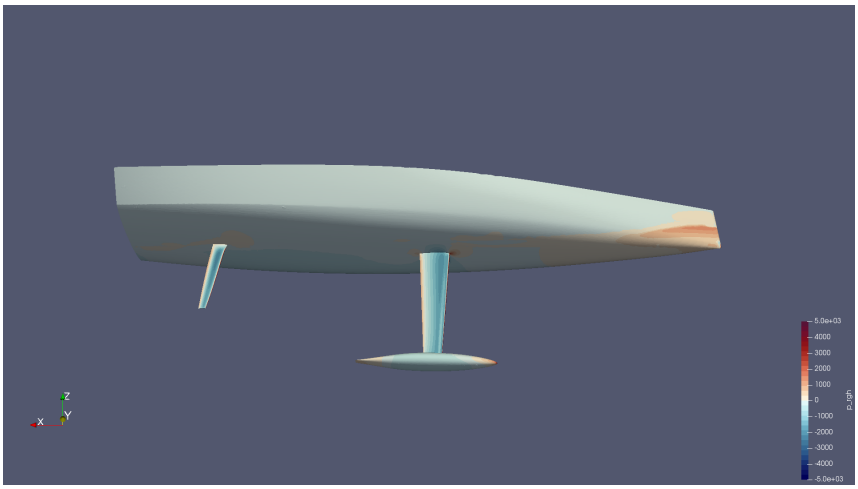


(c) Number of Prism Layers [0, 6]

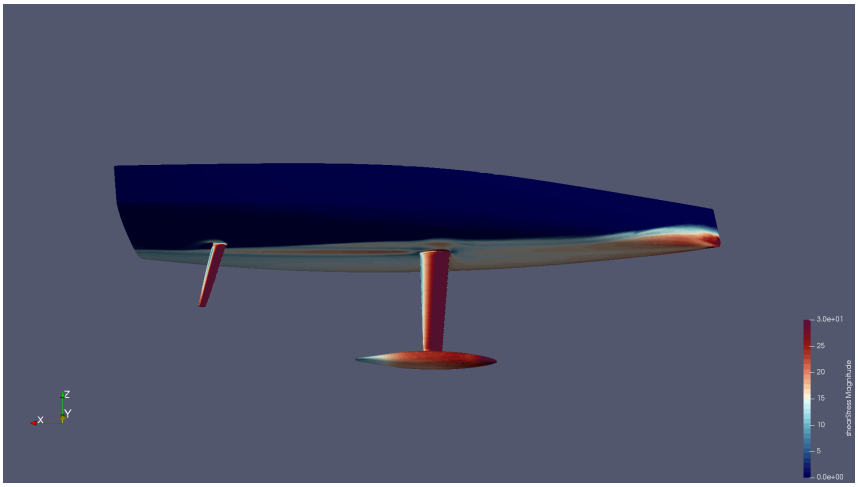
Figure 11.14.: Wide and Light Study: Pictures for Case 86 1/4



(a) yPlus [20, 200]

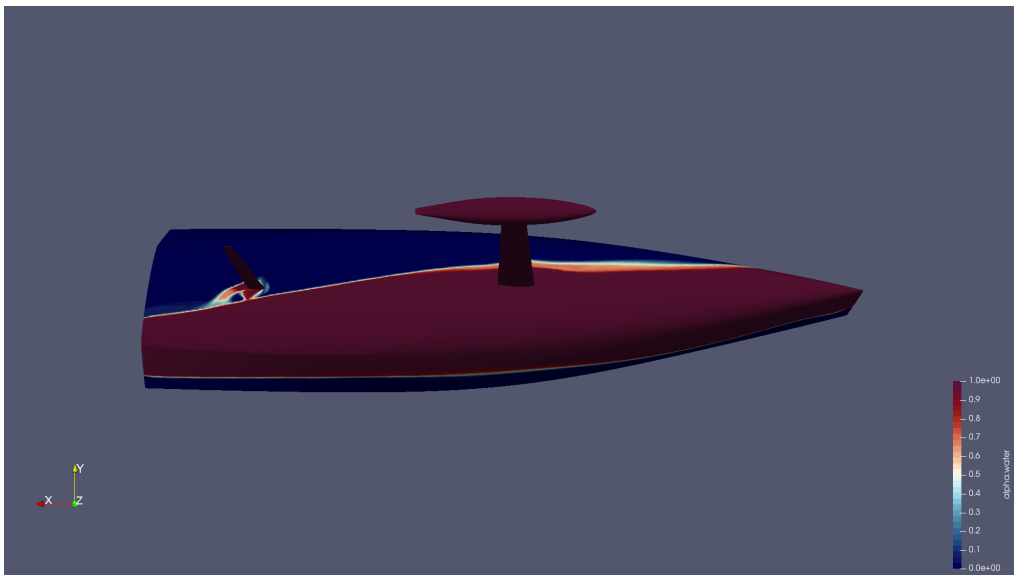


(b) Dynamic Pressure [-5000, 5000]

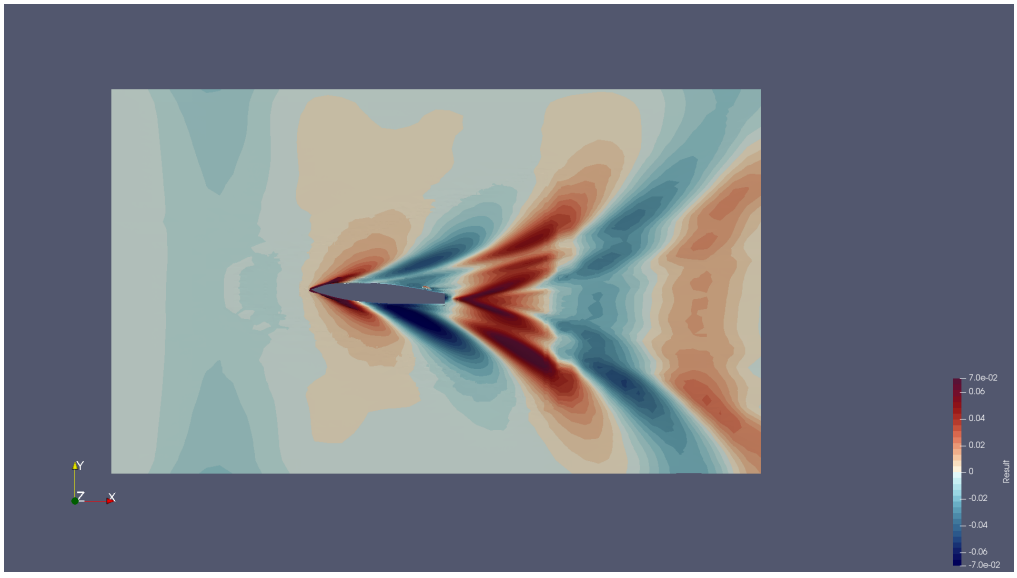


(c) Magnitude of Shear Stress [0, 30]

Figure 11.15.: Wide and Light Study: Pictures for Case 86 2/4

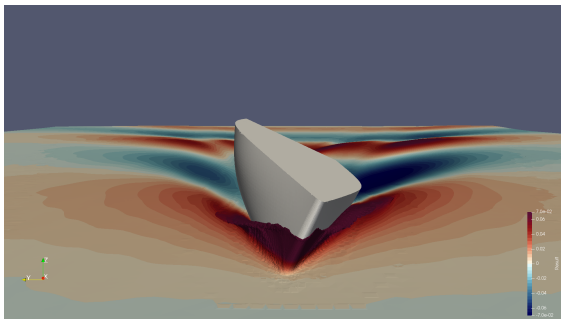


(a) Volume Fraction at yacht [0, 1]

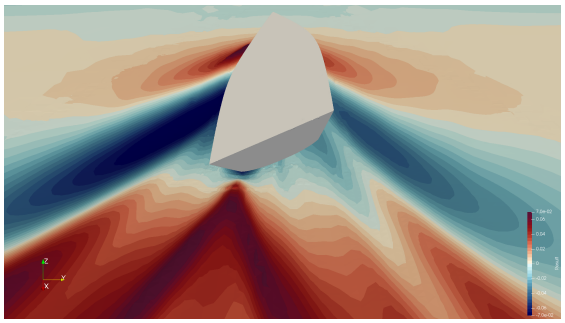


(b) Free Surface: Top View

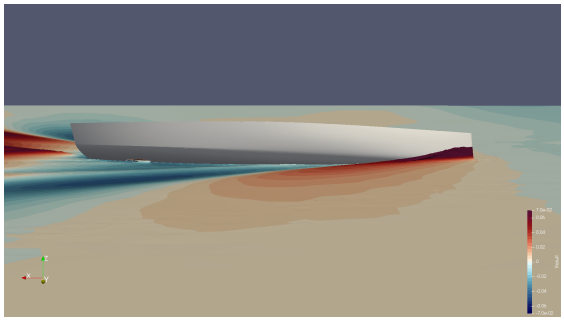
Figure 11.16.: Wide and Light Study: Pictures for Case 86 3/4



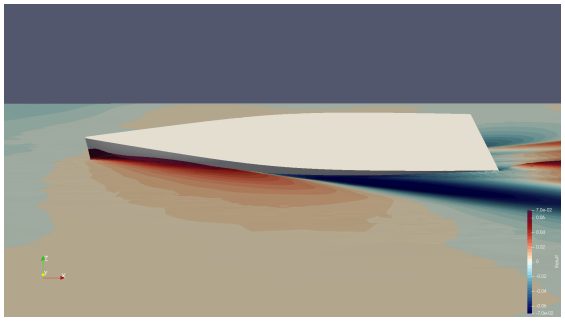
(a) Free Surface: Bow



(b) Free Surface: Stern



(c) Free Surface: Starboard



(d) Free Surface: Portside

Figure 11.17.: Wide and Light Study: Pictures for Case 86 4/4

Table 11.5.: Wide and Light Study: Overview of the results

						Tank		Star-CCM+				FINE/Marine				NSTFoam				OpenFOAM			
No	Fn	LCG	Heel	Yaw	Rudder	Fx	Fy	Fx	Fy	ΔFx	ΔFy	Fx	Fy	ΔFx	ΔFy	Fx	Fy	ΔFx	ΔFy	Fx	Fy	ΔFx	ΔFy
[ ]	[ ]	[m]	[°]	[°]	[°]	[N]	[N]	[N]	[N]	[%]	[%]	[N]	[N]	[%]	[%]	[N]	[N]	[%]	[%]	[N]	[N]	[%]	[%]
1	0.10	2.488	0.0	0.0		3.2	0.0	3.3	0.0	3.2	0.0	2.9	0.0	-10.4	0.0	2.6	0.0	-19.6	0.0	6.9	0.0	115.8	0.0
3	0.20	2.469	0.0	0.0	-	11.9	0.0	12.3	0.0	3.8	0.0	10.8	0.0	-9.3	0.0	10.7	0.0	-10.1	0.0	11.8	0.0	-0.1	0.0
5	0.30	2.447	0.0	0.0	-	28.6	0.0	27.8	0.0	-2.7	0.0	26.1	0.0	-8.5	0.0	27.0	0.0	-5.5	0.0	27.9	0.0	-2.4	0.0
9	0.50	2.330	0.0	0.0	-	116.6	0.0	111.2	0.0	-4.6	0.0	108.9	0.0	-6.6	0.0	113.8	0.0	-2.3	0.0	111.5	0.0	-4.3	0.0
21	0.50	2.784	0.0	0.0	-	118.7	0.0	109.0	0.0	-8.1	0.0	107.2	0.0	-9.7	0.0	110.0	-0.3	-7.3	0.0	103.3	0.0	-12.9	0.0
23	0.50	2.484	0.0	0.0	-	113.6	0.0	107.5	0.0	-5.4	0.0	103.3	0.0	-9.0	0.0	109.3	-0.2	-3.8	0.0	100.8	0.0	-11.3	0.0
25	0.50	2.184	0.0	0.0	-	124.1	0.0	119.0	0.0	-4.2	0.0	117.8	0.0	-5.1	0.0	121.6	-0.2	-2.0	0.0	120.8	0.0	-2.7	0.0
26	0.25	2.459	15.0	0.0	-	18.1	1.2	17.8	2.0	-1.6	75.8	16.2	1.1	-10.3	-9.3	16.4	1.0	-9.2	-15.5	18.2	2.6	0.6	127.3
28	0.35	2.431	15.0	0.0	-	38.6	3.5	36.9	4.0	-4.4	15.3	35.7	3.4	-7.3	-1.7	37.3	3.0	-3.1	-12.0	39.6	5.8	2.6	66.9
30	0.45	2.367	15.0	0.0	-	86.4	12.5	82.2	13.6	-4.9	8.5	81.1	13.7	-6.2	9.5	84.3	12.5	-2.4	0.2	86.6	16.7	0.2	33.5
36	0.35	2.431	15.0	2.0	-	38.3	-1.5	36.0	-0.4	-6.0	-72.3	35.2	-0.7	-8.2	-55.0	36.6	-1.4	-4.5	-8.7	39.3	0.4	2.5	-129.1
38	0.35	2.431	15.0	-1.0	-	38.9	4.5	37.0	6.6	-4.9	46.7	36.4	5.4	-6.5	18.4	37.9	5.3	-2.7	16.3			-100.0	-100.0
40	0.35	2.431	15.0	-3.0	-	40.1	8.8	39.0	11.3	-2.7	29.0	39.0	10.3	-2.9	17.6	39.1	10.1	-2.4	14.8	43.4	15.7	8.2	79.3
41	0.50	2.330	25.0	2.0	-	106.8	20.4	100.1	20.4	-6.3	0.3	100.2	20.2	-6.1	-1.1	98.5	20.4	-7.8	0.0	101.4	24.0	-5.0	17.8
43	0.50	2.330	25.0	-1.0	-	116.4	50.4	109.8	48.3	-5.7	-4.1	110.2	49.8	-5.4	-1.2	108.7	50.7	-6.6	0.6	112.7	56.7	-3.2	12.6
45	0.50	2.330	25.0	-3.0	-	130.3	76.1	122.2	72.4	-6.2	-4.8	123.2	74.5	-5.4	-2.1	119.4	73.3	-8.3	-3.7	123.6	82.1	-5.2	7.9
46	0.10	2.488	0.0	0.0	0.0	4.6	0.0	5.9	0.0	28.9	0.0	4.5	0.0	-2.2	0.0	4.2	0.9	-8.1	0.0	7.4	-0.3	61.3	0.0
48	0.20	2.469	0.0	0.0	0.0	17.3	0.0	19.3	0.0	11.6	0.0	16.4	0.0	-4.8	0.0	16.7	-1.6	-3.0	0.0	23.5	-0.9	35.9	0.0
50	0.30	2.447	0.0	0.0	0.0	40.2	0.0	41.7	0.0	3.7	0.0	38.1	0.0	-5.3	0.0	39.4	0.8	-2.1	0.0	53.7	-1.9	33.7	0.0
54	0.50	2.330	0.0	0.0	0.0	153.0	0.0	150.3	0.0	-1.8	0.0	144.1	0.0	-5.8	0.0	151.5	-3.4	-1.0	0.0	188.1	-5.6	22.9	0.0
61	0.50	2.784	0.0	0.0	0.0	151.2	0.0	147.6	0.0	-2.4	0.0	142.9	0.0	-5.5	0.0	146.3	-3.5	-3.2	0.0	174.5	-5.6	15.4	0.0
63	0.50	2.484	0.0	0.0	0.0	147.2	0.0	144.7	0.0	-1.7	0.0	140.3	0.0	-4.7	0.0	146.3	-2.8	-0.6	0.0	176.1	-5.7	19.7	0.0
65	0.50	2.184	0.0	0.0	0.0	163.5	0.0	159.7	0.0	-2.3	0.0	156.6	0.0	-4.2	0.0	160.2	-3.3	-2.1	0.0	196.6	-5.7	20.2	0.0
71	0.25	2.459	15.0	0.0	0.0	26.3	-	28.2	-5.6	7.3	-	25.5	-3.9	-2.9	-	25.6	-6.0	-2.5	-	36.3	-5.6	38.3	-
73	0.35	2.431	15.0	0.0	0.0	54.7	-	55.8	-6.5	2.0	-	53.1	-0.9	-2.9	-	54.9	-4.2	0.4	-	75.0	-6.6	37.2	-
75	0.45	2.367	15.0	0.0	0.0	115.4	-	115.0	-2.9	-0.4	-	111.7	8.8	-3.2	-	115.3	-5.4	0.0	-	148.1	-2.8	28.4	-
81	0.35	2.431	15.0	2.0	0.0	56.2	-134.9	56.9	-128.3	1.2	-4.9	53.6	-122.0	-4.7	-9.5	56.1	-128.7	-0.2	-4.6	77.7	-122.0	38.3	-9.6
83	0.35	2.431	15.0	-1.0	0.0	55.2	54.5	55.9	55.1	1.2	1.1	52.5	60.0	-4.8	10.1	55.4	56.7	0.3	4.1	76.3	53.0	38.2	-2.8
85	0.35	2.431	15.0	-3.0	0.0	58.8	193.3	59.3	177.9	1.0	-8.0	57.0	173.1	-2.9	-10.5	58.0	181.5	-1.3	-6.1	83.2	168.4	41.7	-12.9
86	0.50	2.330	25.0	2.0	0.0	151.7	-245.1	148.6	-248.9	-2.1	1.6	143.6	-235.0	-5.3	-4.1	143.8	-247.1	-5.2	0.8	185.3	-210.4	22.2	-14.1
88	0.50	2.330	25.0	-1.0	0.0	155.6	111.6	149.6	77.9	-3.9	-30.2	145.0	91.8	-6.8	-17.8	146.3	82.1	-6.0	-26.4	185.7	82.2	19.3	-26.3
90	0.50	2.330	25.0	-3.0	0.0	167.1	325.1	161.4	291.0	-3.4	-10.5	159.5	290.7	-4.5	-10.6	157.4	296.6	-5.8	-8.8			-100.0	-100.0
91	0.35	2.431	15.0	-1.0	-2.0	55.4	75.1	55.9	77.6	0.8	3.4	53.9	74.0	-2.8	-1.3	55.3	76.3	-0.2	1.6	76.5	73.1	38.1	-2.6
94	0.35	2.431	15.0	-3.0	0.0	58.8	208.1	59.3	177.9	1.0	-14.5	57.0	173.1	-2.9	-16.8	58.0	181.5	-1.3	-12.8	83.2	168.4	41.7	-19.1
95	0.35	2.431	15.0	-3.0	-4.0	59.9	246.1	61.0	220.7	1.8	-10.3	58.4	215.9	-2.4	-12.3	58.3	220.6	-2.6	-10.4	86.0	207.0	43.5	-15.9

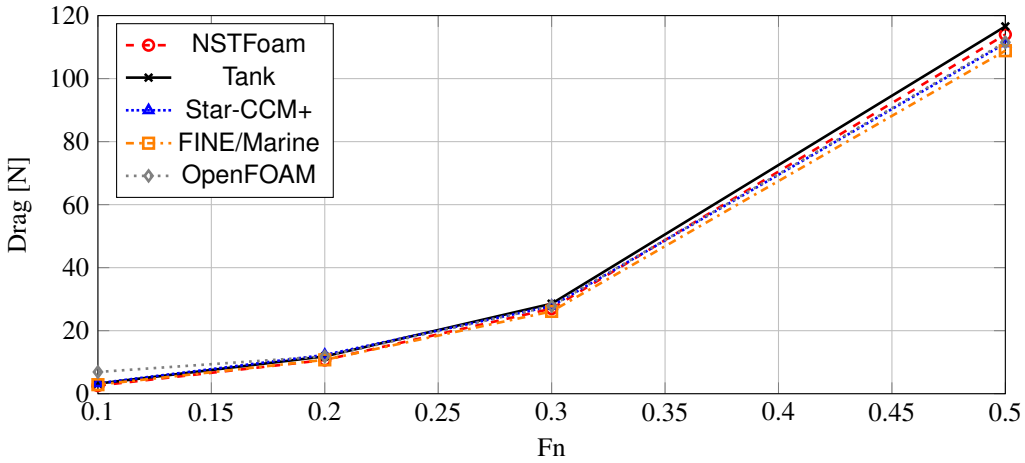


Figure 11.18.: Wide and Light Study: Canoe Body, Upright Resistance

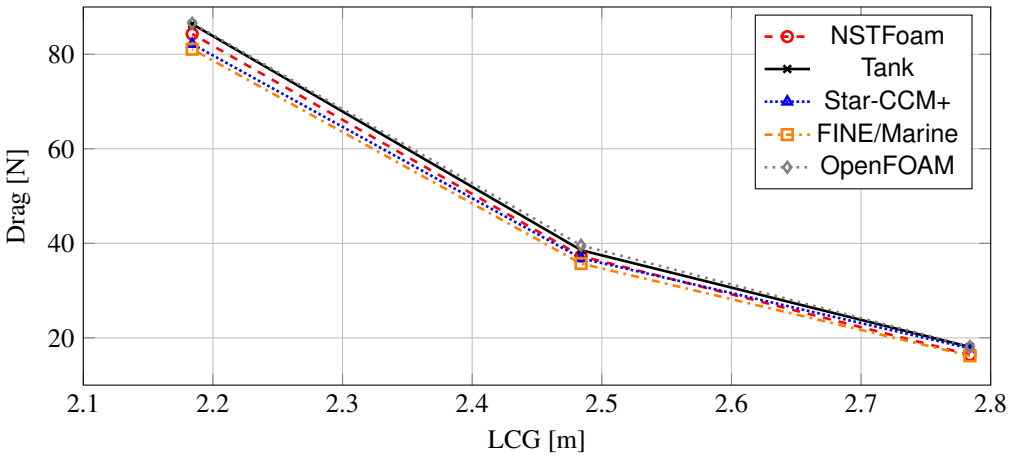
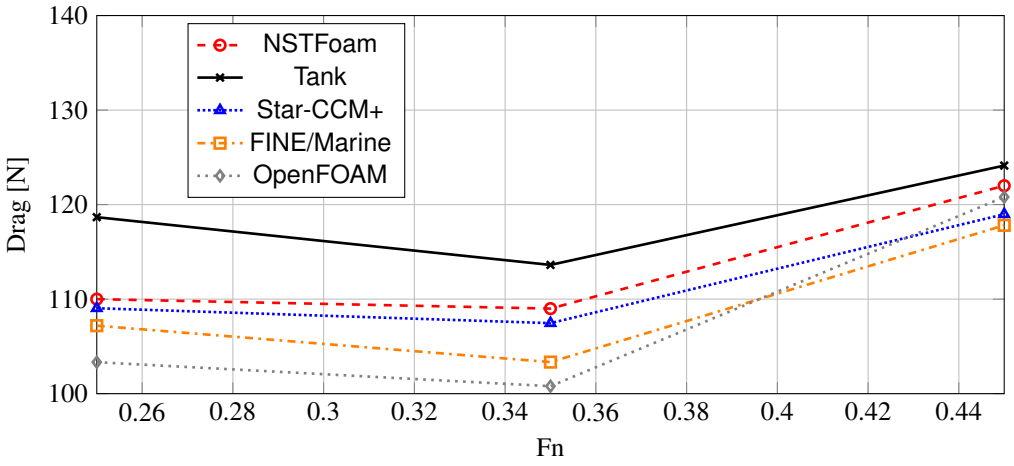
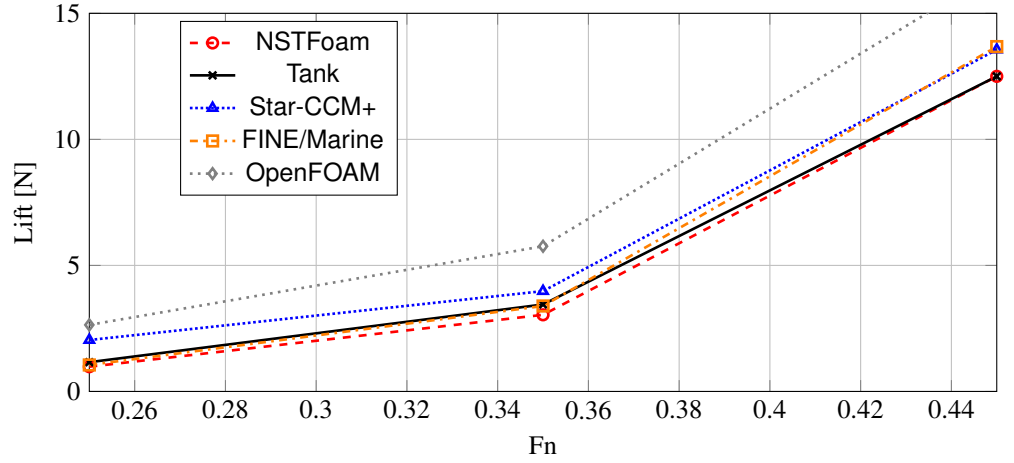


Figure 11.19.: Wide and Light Study: Canoe Body, Upright, LCG Variation, Fn = 0.5

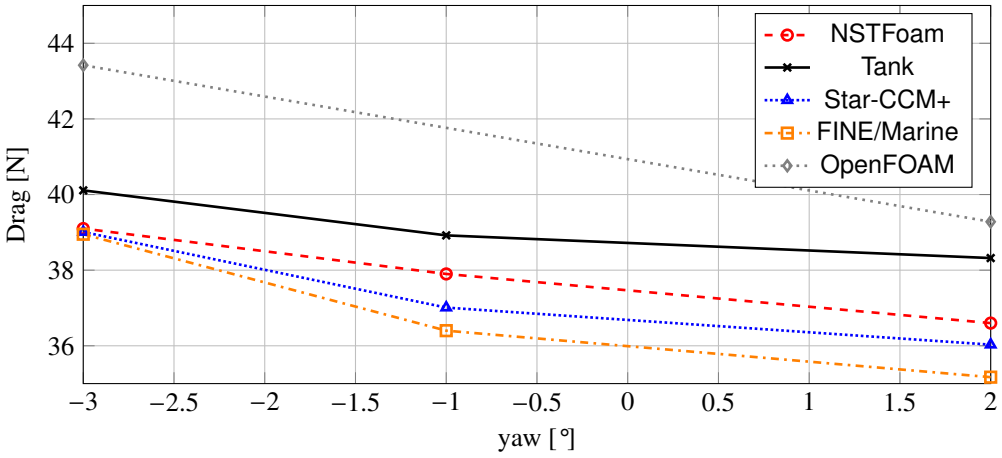


(a) Drag

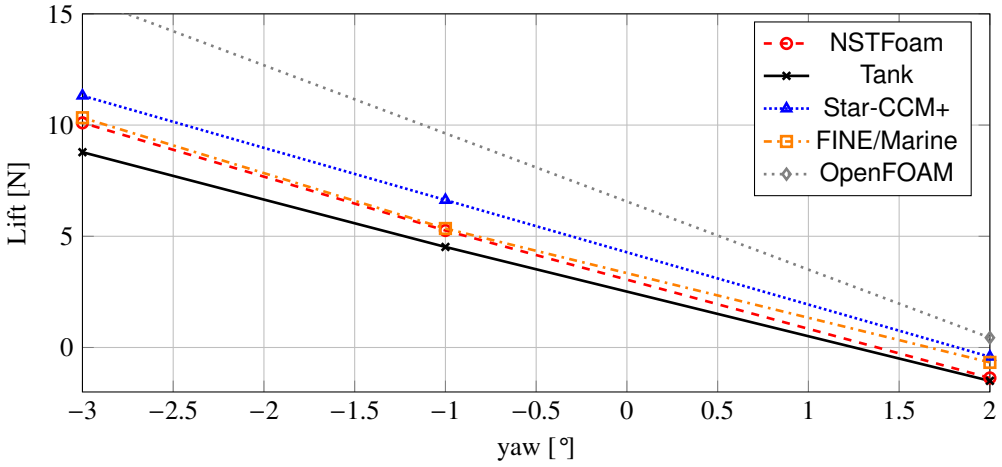


(b) Lift

Figure 11.20.: Wide and Light Study: Canoe Body, Heeled at zero yaw

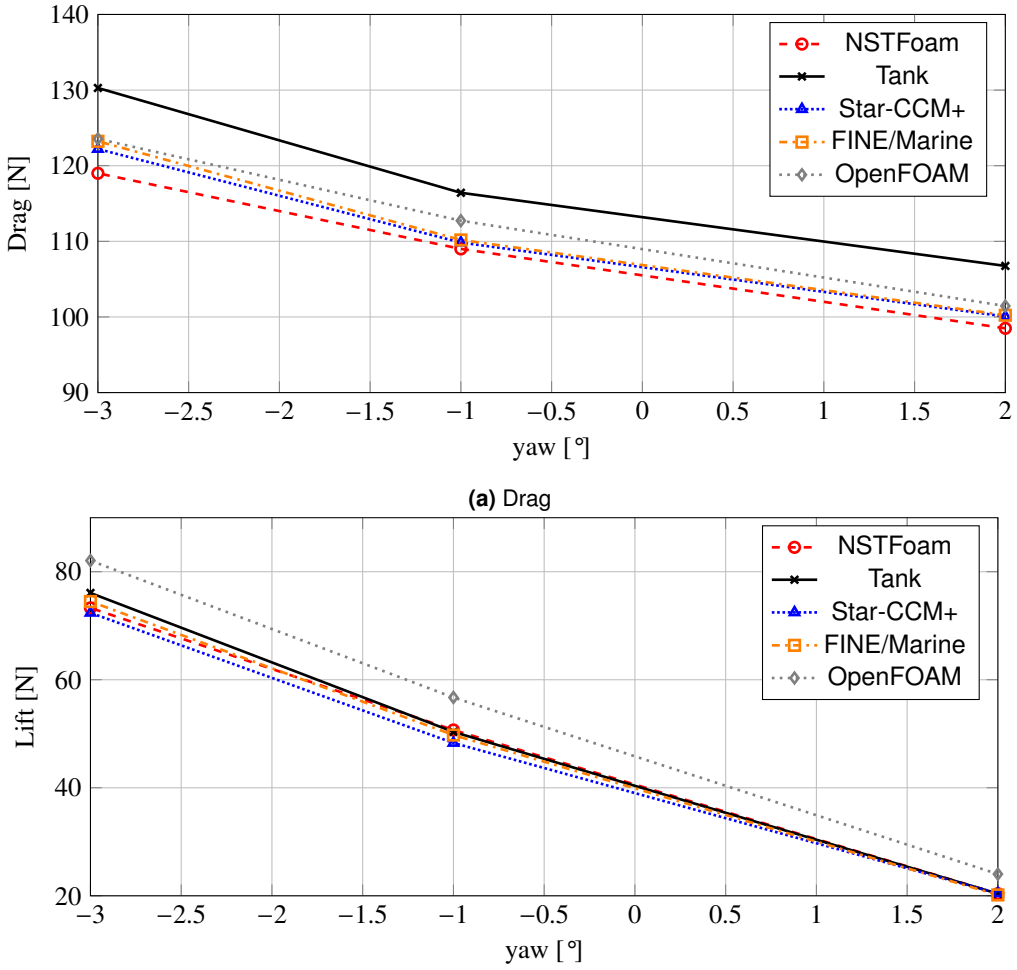


(a) Drag



(b) Lift

Figure 11.21.: Wide and Light Study: Canoe Body, Heeled with yaw, Fn = 0.35



(a) Drag

(b) Lift

Figure 11.22.: Wide and Light Study: Canoe Body, Heeled with yaw, Fn = 0.5

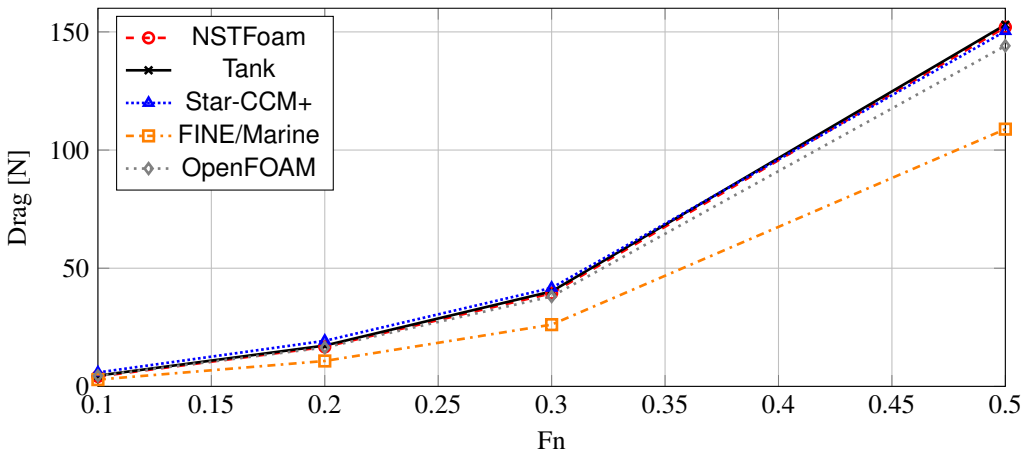


Figure 11.23.: Wide and Light Study: Hull/Keel/Rudder, Upright Resistance

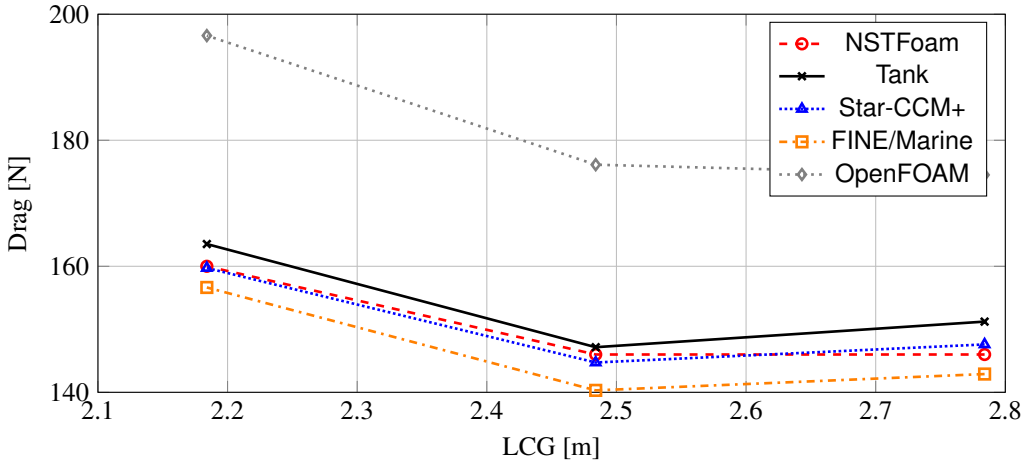
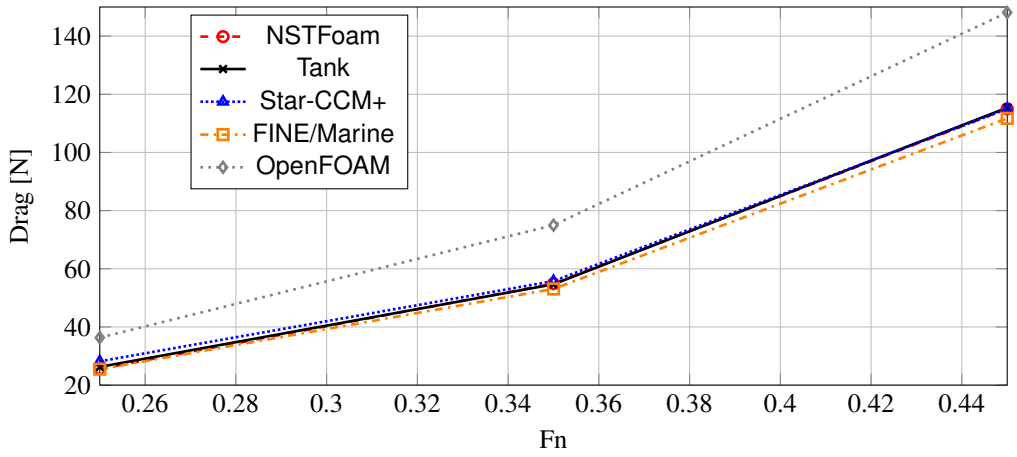
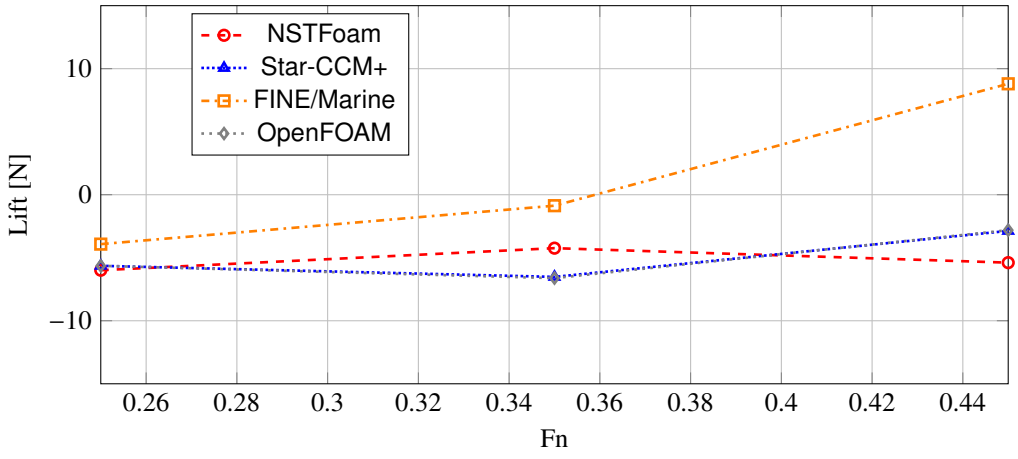


Figure 11.24.: Wide and Light Study: Hull/Keel/Rudder, Upright, LCG Variation, Fn = 0.5

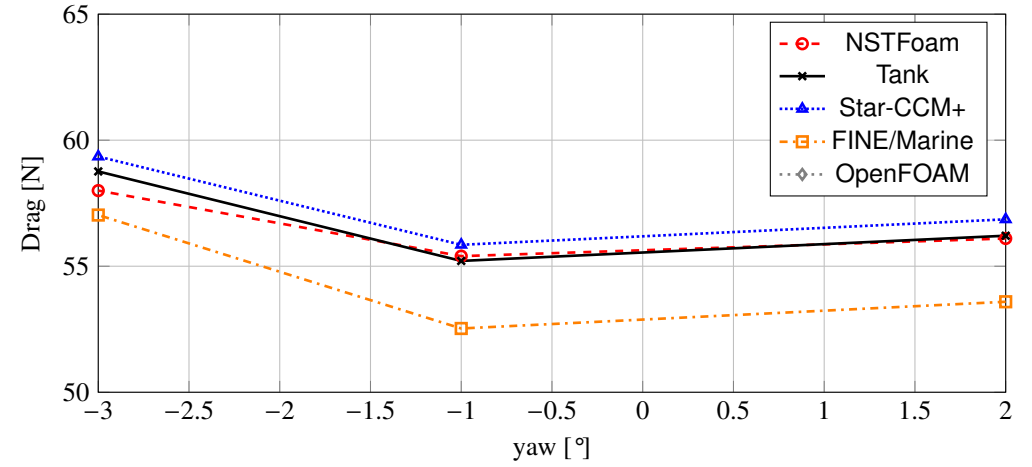


(a) Drag

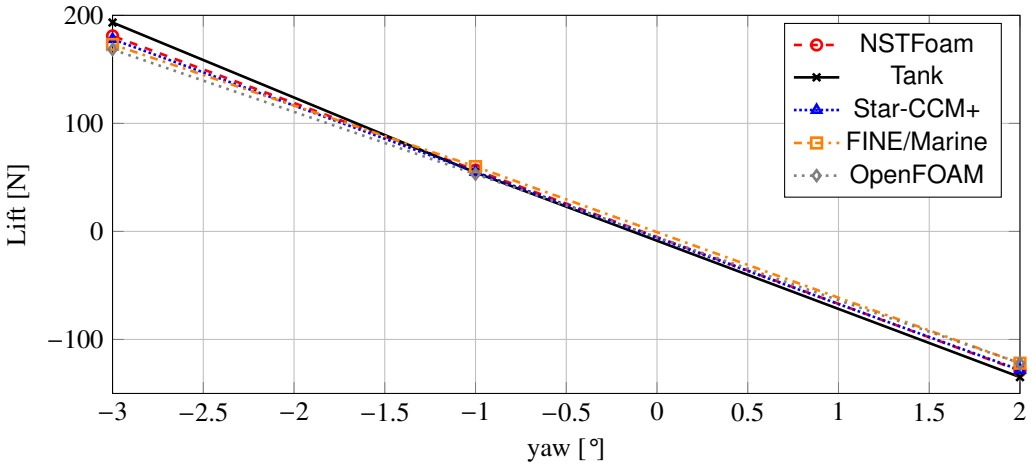


(b) Lift

Figure 11.25.: Wide and Light Study: Hull/Keel/Rudder, Heeled at zero yaw

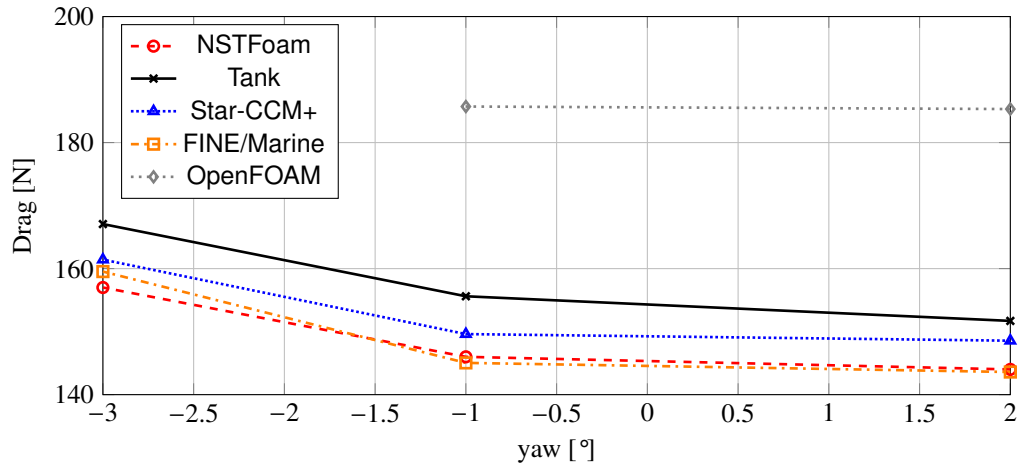


(a) Drag

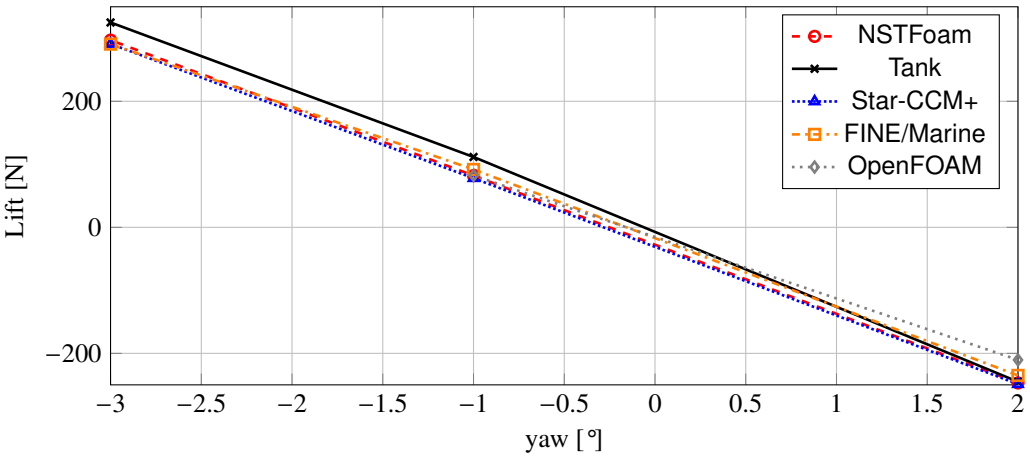


(b) Lift

Figure 11.26.: Wide and Light Study: Hull/Keel/Rudder, Heeled with yaw, Fn = 0.35

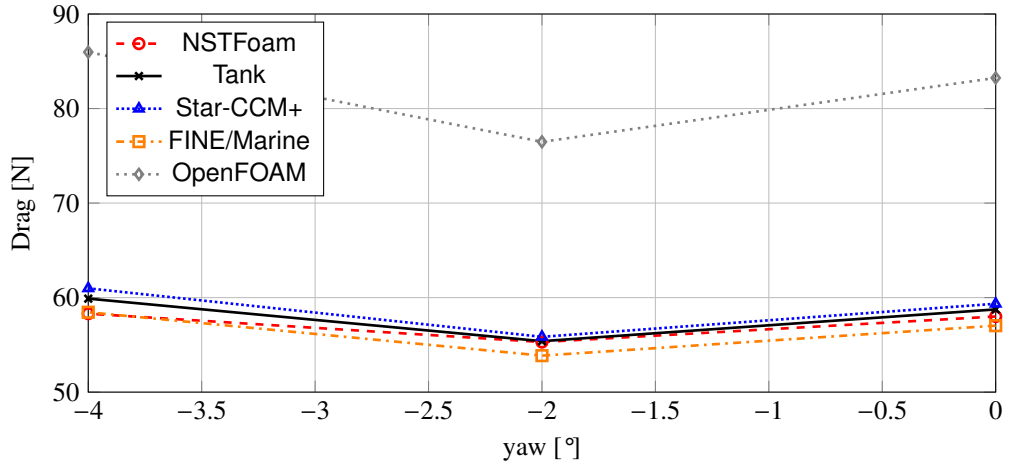


(a) Drag

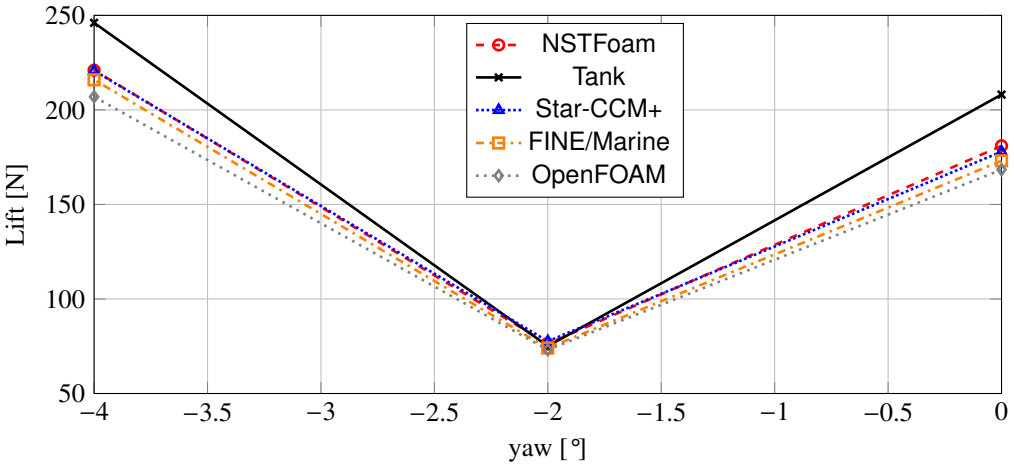


(b) Lift

Figure 11.27.: Wide and Light Study: Hull/Keel/Rudder, Heeled with yaw, Fn = 0.5



(a) Drag



(b) Lift

Figure 11.28.: Wide and Light Study: Hull/Keel/Rudder, Yaw and Rudder Sweep, Fn = 0.35

12. Additional Contributions for Simulating Scour

This chapter gives a short introduction into the extension of the flow solver for scour simulations. After a brief description of the motivation and the basic principle of the newly developed sediment simulation model, the reader is referred to the publication attached in appendix C for the detailed description.

12.1. Motivation

Scour is the erosion of sediment on the seabed near offshore structures like offshore wind power stations. To avoid an excessive weakening of the structural strength the driving depth of respective offshore foundings is increased and scour protection systems might be applied. Furthermore the classification society may dictate the application of scour monitoring systems. As all measures result in substantial extra costs it is of large interest to be able to predict the magnitude of the scour. Currently most common prediction methods are based on very simple empirical equations, which are restricted to be applied to simple obstacles like circular piles. They may overestimate the scour many times.

In the last decades one has begun to develop simulation methods for the scour prediction. In this context it has been shown that the long time scale of the scour development in combination with the small time scale of the vortices shedded by the offshore structures is challenging and leads to high computational costs. Furthermore, the structures can be of complex shape (for example mudplates). In addition the forces are not easy to model as they are often generated by a combination of a current and irregular waves.

The main goal is to extend the previously described free-surface simulation method with the possibility to simulate scouring around complex arbitrarily shaped offshore structures. Furthermore, the solution time should allow to use the method for industrial applications, meaning the added scouring model should not significantly increase the calculation effort. For the final method, the environmental forces acting onto the structures should result from current, waves or a combination of both. Nevertheless, as the first developing step only current is considered in this work.

12.2. Basic Principle of the Sediment Model

For the scouring simulations the previously described Volume-of-Fluid model is extended with an additional soil phase leading to three phases air, water and soil. An additional suspension phase is than indirectly given if a cell contains a specific amount of water and soil.

The sediment behavior is modeled by a Bingham model (Voelkner et al. 2015). This allows to model the sediment as a non-Newtonian fluid using an additional variable dynamic viscosity μ_{Bingham} in the

diffusive term of the momentum equation (2.1).

If the local shear stress in the sediment is below the sediment specific yield stress the sediment should behave as a solid. The solid behavior is modeled by using a relatively high local dynamic Bingham viscosity. If the local shear stress is higher than the yield stress the sediment begins to behave like a fluid. This behavior is modeled by reducing the local dynamic Bingham viscosity. To make this approach functional additional methods were developed within this work:

- development and implementation of a new approach to estimate relative pressure required for the Bingham model,
- development and implementation of a new model to simulate soil sliding,
- development and implementation of a way to transfer OpenFOAMs wall functions onto the domain internal sediment wall,
- and the development and implementation of a method to prevent wrong sediment creeping while using only one shared velocity field for all relevant phases (soil, suspension and water).

For the detailed description of the developed method and the achieved results, the reader is referred to appendix C.

13. Conclusion, Recommendations and Outlook

13.1. Conclusion

The main objectives of the presented work can be divided into two parts. The first was to develop a RANSE-based, state-of-the-art fluid simulation method to simulate moving ships, especially sailing yachts, in calm water or in sea waves. The second objective was to extending this method with the possibility to simulate scouring processes around arbitrary offshore-foundations.

The starting point for the development was the given free-surface solver *interFoam* contributed within the OpenFOAM framework. The interFoam solver contains approaches for many relevant parts of the first objective of this work. Indeed, some of its important features are of insufficient accuracy and/or stability for the desired field of application. By incorporating well known and newly developed methods, a completely new free-surface solver was added to OpenFOAM enabling state-of-the-art simulations and avoiding previously described issues. For the second part of this work, the new free-surface solver was extended with a simple but fast method to simulate scouring processes.

First simulations with the original interFoam solver show that the discontinuous density at the free-surface lead to numerical instabilities using standard unmodified linear interpolation. Using such an interpolation scheme leads to infinitely growing velocities in the air phase, directly above the free-surface. This issue was successfully solved by implementing a reconstruction method for this discontinuity.

Furthermore, a comparison of the interFoam results to those achieved with Star-CCM+ show that the used MULES algorithm for the calculation of the free-surface does not deliver state-of-the-art results. The free-surface is smeared over too many cells, the numerical ventilation is higher and the wave height is damped due to an insufficient convection. Three slightly different high-resolution schemes, HRIC, BICS and BRICS, were implemented and proved to successfully improve the results. All of them deliver good results and due to it's slightly better discontinuity treatment the BRICS scheme was finally chosen. As the numerical ventilation might still be high, especially for cases with the stem above the free-surface, an additional suppression of numerical ventilation was developed and implemented.

Also the original method for the calculation of the rigid body motion was affected by stability problems. The reason is the explicit time marging method which can not be changed to a real implicit method. The problem was solved by introducing an additional mass, which corrects the defect in the forces used inside the motion equation.

Furthermore, to make the simulation of the ship motions applicable onto sailing yachts, an estimation of the sail forces and moments, relevant for sink and trim, was implemented.

Necessary features for the simulation of sea waves were added by incorporating the *waves2Foam* library. A first problem in the simulation of sea waves were unphysical wave reflections at the outlet. Therefore, two wave damping methods were implemented, as damping of the waves in front of the outlet is a suitable solution for this problem. The first method is a well known sponge layer method. It delivers good results with reflections below 2%, but it also requires a parameter adjustment depending on the simulated waves. Therefore a new wave damping method was developed. It was shown that the new method, based on implicitly relaxing the solution to target values in front of the outlet, delivers equally good results as the sponge layer method. Furthermore, it does not underlie additional parameter adjustment for a huge variety of wave types.

As is widely known, the simulation of sea waves requires a second order discretization scheme for the time derivative. This was proven using the first order backward Euler scheme, which leads to a significant wave damping. Furthermore it was shown that the second order schemes Crank-Nicolson and BDF(2) do not suffer from wave damping. Instead, their application leads to numerical instabilities in typical simulations in the designated engineering field. The reason is the missing L-stability of these schemes, which is relevant in the application area of the solver. The problem was solved, by developing a slightly modified BDF(2) scheme, which locally blends to backward Euler for cells, where the stability is not given. It was shown, that this new method remedies the stability issues. Further, it was shown that the local blending is only applied in very few cells which is why the solution is still second order accurate, meaning that no undesired wave damping occurs.

Last but not least, the turbulence model also required a modification for the simulation of sea waves. As the turbulent viscosity becomes unphysical high in simulations with swell, the sea waves itself gets damped significantly. This was counteracted, by implementing a modified k-ω-SST turbulence model, which uses a stabilized calculation of the turbulent viscosity for potential flow regions. Another shortcoming of the k-ω-SST model, not only relevant for sea waves, is the inherently given turbulent decay behind the inlet. It was remedied by an approach based on ambient values. With this, the turbulent variables are only calculated by the unmodified turbulence model close to the object. In the area far away from the objects the ambient values are used.

For the validation, the newly developed method was applied onto several test cases. Based on the Rayleigh Taylor instability problem, the results could be compared to analytical results, whereby a good compliance was observed.

Based on the Sysser 60 hull from the Delft Systematic Yacht Hull Series, the results where investigated qualitatively. The results were compared to the simulation results of the original interFoam solver and to the results achieved with Star-CCM+. It could be shown, that only interFoam suffers from the previously described problems. Further, the quality of the newly developed solver were of same accuracy as those of Star-CCM+.

Finally, a quantitative validation was made based on the Wide and Light study of the Sailing Yacht Research Foundation. The results of the new solver were compared to tank results and simulation results of several contributors using different RANSE-codes. The results of the new solver showed to be in good agreement with the tank results, even though the lift estimation is not as good as the drag

estimation. Nevertheless, the same conclusion was made for the results of the other contributors and the comparison showed that the new solver is of same quantitative accuracy as the other commercial solvers.

For the simulation of scouring a Bingham approach was used to model the sediment and the suspension. It introduces an additional viscosity, which is set to relatively high values to model a solid sediment behavior. For the calculation of this viscosity the calculation of the relative pressure is required. For it's estimation a new approach based on the solution of a Poisson equation was introduced.

A new sliding model was developed for regions where the sediment surface angle exceeds the angle of internal friction. It is mainly based on identifying the relevant regions and reducing the Bingham viscosity to allow the sediment to being moved by the gravity or flow forces.

To achieve a good approximation of the flow in the region of the sediment wall, the standard smooth wall functions of OpenFOAM were transferred to the domain internal sediment wall using a newly developed approach. It was shown, that the results of the transferred wall function are in very good alignment with the results of OpenFOAMs original wall function at a real domain wall. Furthermore, it was shown that the results are significantly different, if the near wall region at the sediment surface is not modeled with such an approach.

It could be shown that essential inaccuracies occur while using one velocity field for the sediment and the water. Instead of using two individual velocity fields, it was decided to introduce a new method based on a buffer cell area between the two phases. This approach could successfully decouple the water velocity from the sediment velocity, so that a small water velocity in the cell above the sediment will not move the solid sediment.

The new simulation method was applied onto several test cases. First, the scour behind an apron was simulated and the results were compared to experimental results. The main phenomenons were resolved well, nonetheless the scour shape differs slightly from the experimental results. This might result from the rough approximation method for the relative pressure. Furthermore, the solver was used to simulate scouring around a vertical cylinder in current. The simulations were compared to experimental results and a proper modeling of nearly all relevant phenomenons could be demonstrated. Only the generation of the suspension was not satisfying and needs further investigations. Finally, the scour around a vertical cylinder with an additional mudplate was simulated to show the applicability onto arbitrary shaped three-dimensional structures. The scour showed to be plausible, but was not validated any further due to missing comparative data. The applicability onto arbitrary geometries is a main advantage of the developed method compared to others.

13.2. Recommendations and Outlook

Discontinuity Treatment

During this work a method for the reconstruction of the pressure jump at the free-surface was implemented to circumvent significant stability issues. This discontinuity reconstruction is used for the momentum and pressure equation. Obviously, the turbulence equations are affected by the same density induced jump. Therefore, it is recommended to incorporate a discontinuity reconstruction into the discretization of the turbulence equations.

In addition to the density, the viscosity also has a discontinuous behavior at the free-surface. The currently used linear interpolation of the viscosity is smearing this sharp jump. This inaccuracy is not stability relevant, but may reduce the quality of the results at the free-surface. Hence, it is recommended to consider the viscosity jump using a proper method. For example using a discretization based on the harmonic mean instead of the linear interpolation can be a first step.

Grid Quality

In the recent years the solver has been applied by several persons for a wide variety of ship simulations. Therefore its main limitations for industrial usage are well known. For such applications the solver is never used alone, but always in combination with a grid generator and a post processing tool for the visualization. Heretofore *snappyHexMesh* and *paraView* have been the chosen applications. With respect to the reliability all problems occurring so far can be affiliated to grid problems, whereby one bad cell can be enough to lead to a diverging solution. This problem can be pinned either to the grid generator or to the solver. In the end the combination of both must be suitable and at least one of them should have an adequate method to circumvent the problem.

Within the OpenFOAM framework this is mainly done on the side of the grid generator snappyHexMesh by prioritizing grid quality over prism layer coverage or an accurate representation of the object's surface. The resulting restrictions are huge and make an industrial application complicated. Therefore the author recommends to improve/introduce a bad cell treatment also on the side of the solver, while relaxing the restrictions for the grid generator. Depending on local mesh quality criteria or convergence criteria this might be done with several approaches. (1) Locally switching from 2nd order schemes to 1st order schemes. (2) Locally using a high under-relaxation factor (although especially for unsteady simulations this might destabilize the solution). (3) Deactivating the model-equations completely for the problematic cells and interpolating the result from the values of the neighbour cells, ideally using an implicit implementation of the interpolation.

Simulation of Very Large Vessels

One main restriction in snappyHexMesh is the cell aspect ratio. With its standard parameters only aspect ratios up to 80 are achievable reliably, which is far too less in the prism layer region. With some adjustment it is possible to achieve values up to 500. This is still not much compared to other commercial tools which easily achieve values from 10,000 to 1,000,000. For large vessels with a length over 200m this is a relevant problem, which can lead to twice as many cells. The problem has to be solved on two sides. First, snappyHexMesh needs to be improved. Second, the reconstruction

operator of OpenFOAM, used to reconstruct the pressure from face centered values to cell centered values, needs to be replaced, as it might result in instabilities for high aspect ratios.

Simulation of Very Small Objects

As mentioned in Chapter 11.3 it can be problematic to use wall functions for very small objects, because a proper fine grid for resolving the pressure might lead to y+ values below the scope of the commonly used wall functions. Therefore the solver should be validated without wall functions and using grids with y+ in the region of 1. Again the cell aspect ratio limitation can be obstructive.

Ships in Waves

To simulate vessels in relatively big waves the used mesh morphing approach requires that the grid has to be refined in a huge vertical area. Furthermore the refinement for the ship induced wave system has to go over the whole wave height. For the first point it might be a good idea to switch from the mesh morphing method to a overlapping grid method. For the second point an adaptive grid refinement method might be advantageous.

Quality of the Free-Surface

During this work high-resolution schemes were implemented for the discretization of the convective term of the Volume-of-Fluid equation. Although they were able to improve the sharpness of the free-surface significantly, they still lead to numerical ventilation and a water-air interface smeared over three cells. Both issues might be remedied replacing the Volume-of-Fluid transport equation with a surface reconstruction method like *Iso-Advector* (Roenby, Bredmose, and Hrvoje Jasak 2016) or the Piecewise-Linear Interface Calculation (PLIC) (see Brackbill, D. Kothe, and Zemach 1992 for example). The quality of the published results is very promising, nevertheless, for the author it is an open question, if the used explicit marging method (CFL limit of 1) might lead to a significant worsing of the calculation time or instabilities.

Automation

The usability of a tool depends on several parameters, whereas the quality of the output is only one of them. For engineers it is important to be able to focus on the engineering work and not to spent the time onto grid or solver setups. To fulfill such needs, the whole execution process of the free-surface solver has been automated giving the engineer an easy to use interface. The input of the tool consist of the geometry files and the user defined test matrix. The grid-setup and -generation, solver-setup and -running, graphical post-processing-setup and -rendering and the quantitative post-processing (averaging and collecting the forces) is completely automated based on a separate C++ and python based application. Besides setting adequate simulation parameters, like the size of the time step or cell sizes, the tool also remedies typical problems occurring while applying snappyHexMesh and allows to do semi-automatized grid studies and much more. The tool is in use since a few years and is subject to continuous further development to this day. It is highly recommended to use the new CFD solver in combination with such automation. This significantly reduces the time it takes to set up a simulation matrix and at the same time offers a high degree of comparability through the use of a uniform setup.

Scouring

As stated out in appendix C several adjustments are recommended to further improve the scouring simulation method. First, it should be advantageous to use two separate velocity fields, one for the sediment and one shared by the water and air phases. This allows to remove the buffer cell approach. Furthermore the modelling method for the sediment based on the Bingham approach should be replaced with a more detailed approach. Replacing it by the Exner equation should allow to consider more physical details of the sediment behaviour while keeping the same calculation speed. Additionally, this should allow to achieve a superior generation of the suspension. Also the smooth wall function used at the sediment surface might be replaced with a rough wall function, which delivers a better approximation of the sediment surface.

Scouring in Current and Sea Waves

The presented simulation results presented in appendix C consider only cases with two phases, the sediment and the water. Nevertheless, the method has been implemented for three phases, already. Therefore, the solver is able to simulate scouring around arbitrary offshore foundations in current and sea waves. First scour simulations with sea waves were done and preliminary results were presented at the *2nd German OpenFoam User meetiNg 2018* (GOFUN 2018). Indeed, these results are not validated. A further development of the solver for scouring in sea waves is highly recommended by the author. Again, as mentioned above, the author assumes that the relevant phenomenons are being resolved better using the Exner Equation instead of the Bingham model.

References

Azcueta, Rodrigo (2001). “Computation of Turbulent Free-Surface Flows Around Ships and Floating Bodies”. PhD thesis. Hamburg University of Technology.

Bellman, Richard and Ralph H. Pennington (1954). “Effects of Surface Tension and Viscosity on Taylor Instability”. In: *Quarterly of Applied Mathematics* 12.2, pp. 151–162. ISSN: 0033569X, 15524485.

Böhm, Christoph (2014). “A Velocity Prediction Procedure for Sailing Yachts with a hydrodynamic Model based on integrated fully coupled RANSE-Free-Surface Simulations”. PhD thesis. Delft University of Technology.

Böhm, Christoph and Kai Graf (2014). “Advancements in free surface RANSE simulations for sailing yacht applications”. In: *Ocean Engineering* 90. Innovation in High Performance Sailing Yachts - INNOVSAIL, pp. 11–20. ISSN: 0029-8018.

Brackbill, J.U., D.B. Kothe, and C. Zemach (1992). “A continuum method for modeling surface tension”. In: *Journal of Computational Physics* 100.2, pp. 335–354. ISSN: 0021-9991.

Chandrasekhar, S (2014). *Hydrodynamic and Hydromagnetic Stability*. New York: Dover Publications, Inc.

Choi, Junwoo and Sung Bum Yoon (2009). “Numerical simulations using momentum source wavemaker applied to RANS equation model”. In: *Coastal Engineering* 56.10, pp. 1043–1060. ISSN: 0378-3839.

Claughton, A.R. (2015a). *SYRF Wide-Light Project*. Tech. rep. Sailing Yacht Research Foundation.

– (2015b). *SYRF Wide-Light Project*. URL: http://sailyachtresearch.org/projects/wide-light-project (visited on 2015).

Courant, R., K. Friedrichs, and Hans Lewy (1928). “Über die partiellen Differenzengleichungen der mathematischen Physik”. In: *Mathematische Annalen* 100, pp. 32–74.

Crank, J. and P. Nicolson (1947). “A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 43.1, pp. 50–67.

Curtiss, C. F. and J. O. Hirschfelder (1952). “Integration of Stiff Equations”. In: *Proceedings of the National Academy of Sciences of the United States of America* 38.3, pp. 235–243. ISSN: 00278424.

Dahlquist, G. (Dec. 1956). “Convergence and stability in the numerical integration of ordinary differential equations”. In: *Mathematica Scandinavica* 4, pp. 33–53.

Dahlquist, G. (1961). “Stability and error bounds in the numerical integration of ordinary differential equations”. In: *Mathematics of Computation* 15, p. 311.

– (1963). “A special stability problem for linear multistep methods”. In: *BIT Numerical Mathematics* 3 (1), pp. 27–43. ISSN: 1572-9125.

Daly, Bart J. (1967). “Numerical Study of Two Fluid Rayleigh-Taylor Instability”. In: *The Physics of Fluids* 10.2, pp. 297–307.

Delft University of Technology (2013). *Delft Systematic Yacht Hull Series*. URL: http://www.dsyhs.tudelft.nl/dsyhs.php (visited on 2013).

Devolder, Brecht, Pieter Rauwoens, and Peter Troch (2017). “Application of a buoyancy-modified k-SST turbulence model to simulate wave run-up around a monopile subjected to regular waves using OpenFOAM ®”. eng. In: *Coastal Engineering* 125, pp. 81–94. ISSN: 0378-3839.

Ehle, B. L. (1973). “A-Stable Methods and Padé Approximations to the Exponential”. In: *Siam Journal on Mathematical Analysis* 4, pp. 671–680.

Euler, Leonhard (1768). *Institutiones calculi integralis*. [s.n.]

Gaskell, P. H. and A. K. C. Lau (1988). “Curvature-compensated convective transport: SMART, A new boundedness- preserving transport algorithm”. In: *International Journal for Numerical Methods in Fluids* 8.6, pp. 617–641. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.1650080602.

Gear, C. William (1971). *Numerical Initial Value Problems in Ordinary Differential Equations*. USA: Prentice Hall PTR. ISBN: 0136266061.

Graf, Kai and Oliver Freiheit (2020). “VPP-Driven Sail And Foil Trim Optimization For The Olympic Nacra 17 Foiling Catamaran”. In: *Innov’Sail 2020* (Gothenburg, Sweden). Vol. 5. Chalmers University of Technology.

Graf, Kai, Janek Meyer, et al. (2017). “Investigation of modern sailing yachts using a new free-surface RANSE code”. In: *Innov’Sail 2017* (Lorient, France). Vol. 4. Cité de la Voile Eric Tabarly and Naval Academy.

Graf, Kai, Hannes Renzsch, and Janek Meyer (2016). “Prediction and optimization of aerodynamic and hydrodynamic forces and boat speed of foiling catamarans with a wing sail and a jib”. In: *The Twenty-Second Chesapeake Sailing Yacht Symposium* (Annapolis, Maryland, USA). Ed. by Britton R. Ward. Society of Naval and Marine Engineers.

Hackbusch, Wolfgang (2005). *Theorie und Numerik elliptischer Differentialgleichungen*. 4th ed. Springer Spektrum.

Hide, Raymond (1955). “The character of the equilibrium of an incompressible heavy viscous fluid of variable density: an approximate theory”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 51.1, pp. 179–201.

Higuera, Pablo, Javier L. Lara, and Inigo J. Losada (2013). “Realistic wave generation and active wave absorption for Navier-Stokes models. Application to OpenFOAM®.” In: *Coastal Engineering* 71. ISSN: 03783839.

Hirt, C. W. and B. D. Nichols (1981). “Volume of fluid (VOF) method for the dynamics of free boundaries”. In: *Journal of Computational Physics*. ISSN: 10902716. arXiv: 0924-0136(97)00224-0.

Huang, Juntao, Pablo M. Carrica, and Frederick Stern (2007). “Coupled ghost fluid/two-phase level set method for curvilinear body-fitted grids”. In: *International Journal for Numerical Methods in Fluids* 55.9, pp. 867–897. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.1499.

Issa, R.I (1986). “Solution of the implicitly discretised fluid flow equations by operator-splitting”. In: *Journal of Computational Physics* 62.1, pp. 40–65. ISSN: 0021-9991.

Iturrioz, A. et al. (2015). “Validation of OpenFOAM® for Oscillating Water Column three-dimensional modeling”. In: *Ocean Engineering* 107. ISSN: 00298018.

Jacobsen, Niels G., David R. Fuhrman, and Jørgen Fredsøe (2012). “A wave generation toolbox for the open-source CFD library: OpenFoam®”. In: *International Journal for Numerical Methods in Fluids* 70.9, pp. 1073–1088. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.2726.

Jasak, H., H.G. Weller, and A.D. Gosman (1999). “High resolution NVD differencing scheme for arbitrarily unstructured meshes”. In: *International Journal for Numerical Methods in Fluids* 31.2, pp. 431–449. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291097-0363%2819990930%2931%3A2%3C431%3A%3AAID-FLD884%3E3.0.CO%3B2-T.

Jasak, Hrvoje (1996). “Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows”. PhD thesis. Imperial College.

Kawamura, T. and H. Miyata (1995). “Simulation of Nonlinear Ship Flows by Density-Function Method”. In: vol. 178. Society of Naval Architects of Japan, pp. 1–8.

Kutta, W. (1901). “Beitrag zur näherungsweisen Integration totaler Differentialgleichungen”. In: *Zeit. Math. Phys.* 46, pp. 435–53.

Larsen, Bjarke Eltard and David R. Fuhrman (2018). “On the over-production of turbulence beneath surface waves in Reynolds-averaged Navier–Stokes models”. In: *Journal of Fluid Mechanics* 853, pp. 419–460.

Launder, B. E. and B. I. Sharma (1974). “Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc”. In: *Letters in Heat and Mass Transfer* 1 (2). ISSN: 00944548.

Leonard, B. P. (1988). “Simple high-accuracy resolution program for convective modelling of discontinuities”. In: *International Journal for Numerical Methods in Fluids* 8.10, pp. 1291–1318. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.1650081013.

Leonard, B.P. (1991). “The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection”. In: *Computer Methods in Applied Mechanics and Engineering* 88.1, pp. 17–74. ISSN: 0045-7825.

Lewis, D. J. and Geoffrey Ingram Taylor (1950). “The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. II”. In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 202.1068, pp. 81–96.

Lord Rayleigh (John William Strutt) (1894). *Theory of Sound*. Vol. 2. New York: Dover Publicat, Inc.ions.

Maele, Karim Van and Bart Merci (2006). “Application of two buoyancy-modified k-epsilon turbulence models to different types of buoyant plumes”. In: *Fire Safety Journal* 41 (2). ISSN: 03797112.

Marić, Tomislav, Douglas B. Kothe, and Dieter Bothe (2020). “Unstructured un-split geometrical Volume-of-Fluid methods – A review”. In: *Journal of Computational Physics* 420, p. 109695. ISSN: 0021-9991.

Mayer, Stefan and Per A. Madsen (2000). “Simulation of Breaking Waves in the Surf Zone using a Navier-Stokes Solver”. In: *27th International Conference on Coastal Engineering*. Ed. by Billy L. Edge, pp. 928–941. eprint: https://ascelibrary.org/doi/pdf/10.1061/40549%28276%2972.

Menter, Florian R., J.C. Ferreira, and Thomas Esch (2003). “The SST Turbulence Model with Improved Wall Treatment for Heat Transfer Predictions in Gas Turbines”. In: *International Gas Turbine Congress 2003*.

Meyer, Janek, Kai Graf, and Thomas Slawig (May 2017). “A new adjustment-free damping method for free-surface waves in numerical simulations”. In: *MARINE VII : proceedings of the VII International Conference on Computational Methods in Marine Engineering* (Nantes, France). Ed. by Michel Visonneau, Patrick Queutey, and David Le Touzé. CIMNE, pp. 296–311. ISBN: 978-84-946909-8-3.

– (May 4, 2021). *Simulation of scour around arbitrary offshore foundations based on the Volume-of-Fluid method combined with a Bingham model*. Version 2. arXiv: arXiv:2012.03051v2.

Meyer, Janek, Hannes Renzsch, et al. (Mar. 2016). “Advanced CFD-Simulations of free-surface flows around modern sailing yachts using a newly developed OpenFOAM solver”. In: *The Twenty-Second Chesapeake Sailing Yacht Symposium* (Annapolis, Maryland, USA). Ed. by Britton R. Ward. Society of Naval and Marine Engineers, pp. 161–177.

Muzaferija, S. and M. Perić (1997). “Computation of Free-Surface Flows using the Finite Volume Method and Moving Grids”. In: *Numerical Heat Transfer, Part B: Fundamentals* 32.4, pp. 369–384. eprint: https://doi.org/10.1080/10407799708915014.

– (1999). “Computation of free surface flows using interface-tracking and interface-capturing methods”. In: Computational mechanics publications. WIT Press. Chap. 2, pp. 59–100.

Muzaferija, S., M. Perić, et al. (1999). “A Two-Fluid Navier-Stokes Solver to Simulate Water Entry”. In: *Twenty-Second Symposium on Naval Hydrodynamics. Washington, DC, 1999*.

Park, Jong Chun, Moo Hyun Kim, and Hideaki Miyata (1999). “Fully non-linear free-surface simulations by a 3D viscous numerical wave tank”. In: *International Journal for Numerical Methods in Fluids* 29 (6). ISSN: 02712091.

Patankar, S.V. (1981). *Numerical heat transfer and fluid flow*. Hemisphere Publishing Corporation.

Perić, Milovan (1985). “A Finite Volume Method for the Prediction of Three-Dimensional Fluid Flow in Complex Ducts”. PhD thesis. Imperial College.

Perić, Milovan and Joel H. Ferziger (2002). *Computational Methods for Fluid Dynamics*. 3rd ed. Springer Verlag.

Perić, R. (2013). “Internal generation of free surface waves and application to bodies in cross sea”. MA thesis. Hamburg-Germany: Hamburg University of Technology.

Perić, R. and M. Abdel-Maksoud (2016). “Reliable Damping of Free Surface Waves in Numerical Simulations”. In: *Ship Technology Research* 63.1, pp. 1–13. eprint: https://doi.org/10.1080/09377255.2015.1119921.

Perić, Robinson et al. (2021). *Optimizing wave-generation and wave-damping in 3D-flow simulations with implicit relaxation-zones*. arXiv: 1806.10995 [physics.flu-dyn].

Queutey, P., M. Visonneau, and P. Ferrant (2004). “Numerical investigation of wave interaction with a fixed vertical circular cylinder”. In: *International Journal of Offshore and Polar Engineering* 14 (3). ISSN: 10535381.

Queutey, Patrick and Michel Visonneau (2007). “An interface capturing method for free-surface hydrodynamic flows”. In: *Computers & Fluids* 36.9, pp. 1481–1510. ISSN: 0045-7930.

Rhie, C. M. and W. L. Chow (1982). “Numerical study of the turbulent flow past an airfoil with trailing edge separation”. In:

Roenby, Johan, Henrik Bredmose, and Hrvoje Jasak (2016). “A computational method for sharp interface advection”. In: *Royal Society Open Science*.

Rubin, S.G and P.K Khosla (1977). “Polynomial interpolation methods for viscous flow calculations”. In: *Journal of Computational Physics* 24.3, pp. 217–244. ISSN: 0021-9991.

Rumsey, Christopher L. and Philippe R. Spalart (2009). “Turbulence model behavior in low reynolds number regions of aerodynamic flowfields”. In: *AIAA Journal* 47 (4). ISSN: 00011452.

Runge, C. (1895). “Ueber die numerische Auflösung von Differentialgleichungen.” In: *Mathematische Annalen* 46, pp. 167–178.

Schmitt, Pál et al. (2020). “Beyond VoF: alternative OpenFOAM solvers for numerical wave tanks”. In: *Journal of Ocean Engineering and Marine Energy* 6 (3). ISSN: 21986452.

Shigunov, Vladimir, Heinrich Söding, and Yaozong Zhou (May 2001). “Numerical Simulation of Emergency Landing of Aircraft on a Plane Water Surface”. In: *HIPER’01* (Hamburg, Germany). Ed. by Volker Bertram. Vol. 2.

Singh, Toni R. (2012). “Die experimentelle Ermittlung der Tauchschwingungen von Strömungskörpern und deren Berechnung mit einem Verfahren der kombinierten Strömungs- und Bewegungssimulation”. MA thesis. University of Applied Sciences Kiel.

Söding, Heinrich (2001). “How to Integrate Free Motions of Solids in Fluids”. In: *4th Numerical Towing Tank Symposium* (Hamburg, Germany). Ed. by Volker Bertram. Vol. 4.

Spalart, Philippe R. and Christopher L. Rumsey (2007). “Effective inflow conditions for turbulence models in aerodynamic calculations”. In: *AIAA Journal* 45 (10). ISSN: 00011452.

Sussman, Mark, Peter Smereka, and Stanley Osher (1994). “A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow”. In: *Journal of Computational Physics* 114.1, pp. 146–159. ISSN: 0021-9991.

Sweby, P. K. (1984). “High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws”. In: *SIAM Journal on Numerical Analysis* 21.5, pp. 995–1011. ISSN: 00361429.

Tao, W.Q. (2000). *Recent Advances in Computational Heat Transfer*. Science Press, Beijing.

Taylor, Geoffrey Ingram (1950). “The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. I”. In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 201.1065, pp. 192–196. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1950.0052.

Ubbink, Onno (Jan. 1997). “Numerical prediction of two fluid systems with sharp interfaces”. PhD thesis. London: Imperial College.

Ursell, F., R. G. Dean, and Y. S. Yu (1960). “Forced small-amplitude water waves: a comparison of theory and experiment”. In: *Journal of Fluid Mechanics* 7.1, pp. 33–52.

van Leer, Bram (1979). “Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method”. In: *Journal of Computational Physics* 32.1, pp. 101–136. ISSN: 0021-9991.

Viola, Ignazio Maria, R.G.J. Flay, and R. Ponzini (2012). “CFD analysis of the hydrodynamic performance of two candidate America’s Cup AC33 hulls”. English. In: *Transactions of the Royal Institution of Naval Architects Part B: International Journal of Small Craft Technology* 154.1. Export Date: 6 March 2014 Source: Scopus, B1–B12. ISSN: 1740-0694.

Voelkner, S. et al. (2015). “An Overset-Grid Three-Phase Flow Model For Offshore Operations”. In: *VI International Conference on Computational Methods in Marine Engineering MARINE 2015*. Rome, Italy, pp. 943–954.

Vukčević, Vuko, Hrvoje Jasak, and Inno Gatin (2017). “Implementation of the Ghost Fluid Method for free surface flows in polyhedral Finite Volume framework”. In: *Computers and Fluids* 153, pp. 1–19. ISSN: 0045-7930.

Wackers, J. (2007). “Surface capturing and multigrid for steady free-surface water flows”. PhD thesis. Delft University of Technology.

Wackers, J. et al. (2011). “Free-Surface Viscous Flow Solution Methods for Ship Hydrodynamics”. In: vol. 18. Archives of Computational Methods in Engineering 1. Springer Netherlands, pp. 1–41.

Ward, Britton R., ed. (Mar. 2016). *The Twenty-Second Chesapeake Sailing Yacht Symposium* (Annapolis, Maryland, USA). Society of Naval and Marine Engineers.

Wilcox, David C. (1988). “Reassessment of the scale-determining equation for advanced turbulence models”. In: *AIAA Journal* 26 (11). ISSN: 00011452.

– (2006). “Turbulence Modeling for CFD (Third Edition)”. In: *DCW Industries* (1).

– (2008). “Formulation of the k-omega turbulence model revisited”. In: vol. 46.

Yu, B. et al. (2001). “Discussion on numerical stability and boundedness of convective discretized scheme”. In: *Numerical Heat Transfer, Part B: Fundamentals* 40.4, pp. 343–365. eprint: https://doi.org/10.1080/104077901317091721.

Appendices

A. Advanced CFD-Simulations of free-surface flows around modern sailing yachts using a newly developed OpenFOAM solver

THE 22nd CHESAPEAKE SAILING YACHT SYMPOSIUM
ANNAPOLIS, MARYLAND, MARCH 2016

Advanced CFD-Simulations of free-surface flows around modern sailing yachts using a newly developed OpenFOAM solver

Janek Meyer, Yacht Research Unit Kiel, Kiel, Germany
Hannes Renzsch, FluidEngineeringSolutions GmbH & Co. KG, Schleswig, Germany
Kai Graf, Yacht Research Unit Kiel, Kiel, Germany
Thomas Slawig, Kiel University, Kiel, Germany

ABSTRACT

While plain vanilla OpenFOAM (OF) has strong capabilities with regards to quite a few typical CFD-tasks, some problems actually require additional solvers and numerical methods for efficient computation of high-quality results. One of the fields requiring these additions is the computation of large-scale free-surface flows as found e.g. in naval architecture. This holds especially for the flow around typical modern yacht hulls, often planing, sometimes with surface-piercing appendages. Particular challenges include, but are not limited to, breaking waves, sharpness of interface, numerical ventilation (aka streaking) and a wide range of flow phenomenon scales. A new OF-based application including newly implemented discretisation schemes, gradient computation and rigid body motion computation is described. The new code is validated against published experimental data; the effect on accuracy, computational time and solver stability is shown by comparison to standard OF-solvers (interFoam / interDyMFoam) and Star-CCM+. The code's capabilities to simulate complex "real-world" flows are shown on a well-known racing yacht design.

NOTATION

Latin letters

BICS	Blended Interface Capturing Scheme
BRICS	Blended Reconstructed Interface Capturing Scheme
CBC	Convective Boundedness Criterion
CDS	Central Differencing Scheme
Co	Courant-Friedrichs-Lewy Number
CoE	Center of Effort
DDS	Downwind Differencing Scheme
GDS	Gamma Differencing Scheme
HRIC	High-Resolution Interface Capturing
IGDS	Inter-Gamma Differencing Scheme
LCG	Longitudinal Center of Gravity
MULES	multi-dimensional limiter for explicit solution
NV	Numerical Ventilation
NVA	Normalized Variable Approach
NVD	Normalized Variable Diagram
OF	OpenFOAM
PISO	Pressure Implicit with Splitting of Operator
SHM	snappyHexMesh
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
UDS	Upwind Differencing Scheme
VOF	Volume-of-Fluid
a	Matrix coefficient
$\mathbf{e}$	Explicit distance vector
$\mathbf{g}$	Gravity vector
g	Z-component of gravity vector
$\mathbf{n}$	Surface normal vector
N	Numerical Ventilation
p	Total pressure
$\mathbf{q}$	Source term
$\mathbf{s}$	Area vector
s	Surface
t	Time
$\mathbf{u}$	Velocity vector
$\hat{\mathbf{u}}$	Velocity vector of old timestep
V	Volume

Sub- and Superscripts

$_c$	Main diagonal or central cell
$_C$	Central cell
$_D$	Downwind cell
$_f$	Cell face
$_i$	Ith cell or row of the matrix
$_L$	Face owner cell
$_n$	Neighbor element in the matrix
$_R$	Face neighbor cell
$_U$	Upwind cell

$^-$	Face owner subdomain
$^+$	Face neighbor subdomain

Greek symbols

α	Volume fraction
Γ	Free surface
θ	Angle between face normal and free surface normal
λ_f	Pressure switch
μ	Dynamic viscosity
μ_e	Effective dynamic viscosity
ρ	Density (kg/m^3)
$\hat{\rho}$	Reverse interpolated face value of density
ϕ	Generic quantity
$\tilde{\phi}$	Normalized generic quantity
Φ	Face flux

Operators

$\mathscr{D}()$	Divergence operator
$\mathscr{I}()$	Interpolation operator
$\mathscr{R}()$	Reconstruct volume field operator
∇	Nabla Operator

INTRODUCTION

OpenFOAM (OF) used in various projects in academia and industry was found to be a powerful CFD-package. Unfortunately, using standard OF VOF-solvers ((LTS)inter(DyM)FOAM) on typical naval architectural problems, two major issues arose, namely lack of stability and exceedingly long turn-around times. Based on previous experience using commercial CFD programs like CFX or Star-CCM+ and developing and using proprietary codes like FreSCo$^+$, it was decided to implement new solvers and numerical methods based on general industry practice. Based on an extensive review of literature and existing programs various methods were implemented and tested and the most successful are described in the following. Within the scope of this paper we call the new solver *OurSolver*.

METHODS

Governing Equations and Solution Algorithm

For the calculation of the flow the incompressible unsteady Reynolds-averaged Navier-Stokes equations are solved. The momentum conservation equation is defined as (employ the eddy-viscosity hypothesis for closure)

$$\begin{aligned}&\frac{\delta\rho\mathbf{u}}{\delta t} + \nabla\cdot(\rho\mathbf{u}\mathbf{u}) - \nabla\cdot\mu_e\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\\ &= -\nabla p + \rho\mathbf{g}\end{aligned} \tag{1}$$

The mass conservation equation is

$$\nabla\cdot\mathbf{u} = 0 \tag{2}$$

Integrating equation (2) over the volume and applying the Gauss Theorem leads to

$$\int_s \mathbf{u}_s\cdot\mathbf{n}ds = 0 \tag{3}$$

or in a non-integral form using Divergence operator $\mathscr{D}()$

$$\mathscr{D}(\mathbf{u}) = 0 \tag{4}$$

For the calculation of the free surface the Volume-of-Fluid (VOF) method introduced in Hirt et al. (1981) is used. Here, an additional transport variable for the volume fraction is introduced. Its conservation equation is given as

$$\frac{\delta\alpha_i}{\delta t} + \nabla\cdot(\alpha_i\mathbf{u}) = 0 \tag{5}$$

with the volume fraction α_i for the ith fluid. The flow properties are then calculated by

$$\rho = \sum_i \rho_i\alpha_i \quad \mu = \sum_i \mu_i\alpha_i \quad 1 = \sum_i \alpha_i \tag{6}$$

The free surface is defined by the volume fraction $\alpha = 0.5$. Using implicit euler discretisation for time the semi-discretized, linearized momentum equation yields

$$\begin{aligned}&\frac{\rho}{\Delta t}\mathbf{u}_c + \frac{1}{V}a_c\mathbf{u}_c + \frac{1}{V}\sum_n a_n\mathbf{u}_n\\ &= \mathbf{q} + \rho\mathbf{g} - \mathscr{R}((\nabla p)_f\cdot\mathbf{s}_f) + \frac{\rho}{\Delta t}\hat{\mathbf{u}}_c\end{aligned} \tag{7}$$

For cell c a_c is the matrix coefficient of the main diagonal and a_n are the respective off diagonal elements resulting from the neighbor cells of the implicit part of the convective and diffusive terms. The subscript f indicates a value at the face. The time discretisation needs the velocity of the old time step $\hat{\mathbf{u}}_c$. The source term $\mathbf{q}$ contains all other explicit parts. For the *In the spirit of Rhie-Chow* interpolation generally used in OpenFOAM, the pressure gradient is calculated at the cell-faces. The reconstruct volume field operator $\mathscr{R}()$ reconstructs a volume field from the face flux field, where $\mathbf{s}_f$ is the face area vector.

$$\mathscr{R}(\phi_f) = \left(\sum_f \frac{\mathbf{s}_f}{|\mathbf{s}_f|}\mathbf{s}_f\right)^{-1}\cdot\left(\sum_f \frac{\mathbf{s}_f}{|\mathbf{s}_f|}\phi_f\right) \tag{8}$$

Rearranging (7) to $\mathbf{u}_c$ yields the velocity equation:

$$\begin{aligned}\mathbf{u}_c = &\left(\frac{\rho}{\Delta t} + \frac{1}{V}a_c\right)^{-1}\\ &\left(\mathbf{q} + \rho\mathbf{g} - \mathscr{R}((\nabla p)_f\cdot\mathbf{s}_f) + \frac{\rho}{\Delta t}\hat{\mathbf{u}}_c - \frac{1}{V}\sum_n a_n\mathbf{u}_n\right)\end{aligned} \tag{9}$$

Substituting and rearranging (9) into (2) yields a poisson equation for the pressure:

$$\mathscr{D}\left(\alpha_f^{-1}\nabla p\right) = \mathscr{D}\left(\alpha_f^{-1}\left(h_{tf} + h_{sf}\right)\right) \tag{10}$$

with

$$\alpha_f = \left(\frac{\mathscr{I}(\rho)}{\Delta t} + \mathscr{I}\left(\frac{1}{V} a_c \right) \right) \tag{11}$$

$$h_{tf} = -\frac{\mathscr{I}(\rho)}{\Delta t} \mathscr{I}(\hat{\mathbf{u}}_c) \tag{12}$$

$$h_{sf} = -\mathscr{I}(\rho)\mathbf{g} + \mathscr{I}\left(\frac{1}{V} \sum_n a_n \mathbf{u}_n - \mathbf{q} \right) \tag{13}$$

and the interpolation operator $\mathscr{I}()$ which interpolates a face field from a volume field using a central differencing interpolation scheme. To avoid a steady state solution depending on the timestep Δt the term for the time is excluded from the interpolation and built on the face directly. Therefore each term has to be interpolated on its own. This is not done in the original interFoam solver, where all terms are interpolated mutually. After solving the pressure equation the fluxes can be updated with

$$\Phi = \frac{\mathbf{s}_f}{\alpha_f} \cdot (h_{tf} + h_{sf} - (\nabla p)_f) \tag{14}$$

The equations are solved in a segregated algorithm shown in figure 1.

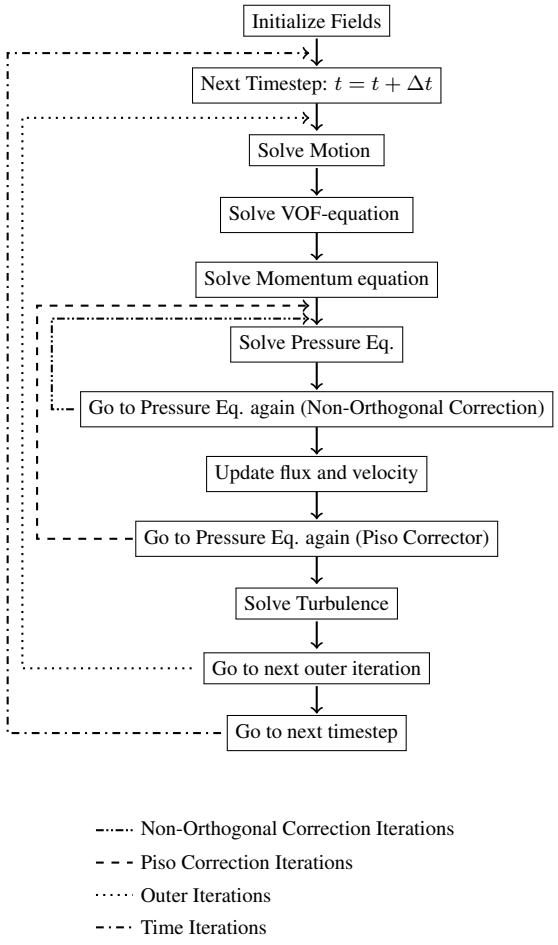


Figure 1: Solution Algorithm

High Resolution Schemes

Keeping a sharp interface using the unmodified transport-equation for the volume-fraction (5) requires a special discretisation scheme for the convective term. The scheme should transform every gradient into a step function to guarantee sharpness. The solution for α has to be bounded between zero and unity. Additionally, the scheme should allow local Courant-Friedrichs-Lewy Numbers (Cos) larger than unity to decrease computational costs without losing stability. Furthermore the scheme should prevent numerical oscillation. These requirements result in a huge family of discretisation schemes. We implemented three schemes, the High-Resolution Interface Capturing (HRIC) scheme presented in Muzaferija et al. (1998), the Blended Interface Capturing Scheme (BICS) described in Queutey et al. (2007) and the Blended Reconstructed Interface Capturing Scheme (BRICS) presented in Wackers et al. (2010). Tests showed that the latter has small advantages.

This chapter first describes the principle of the Normalized Variable Diagram (NVD). Afterwards the high resolution scheme BRICS is being described.

Normalized Variable Diagram

The NVD is a common way to define High-Resolution schemes. Considering three neighboring cells the scheme for a generic quantity ϕ can be formulated as

$$\phi_f = f(\phi_U, \phi_C, \phi_D). \tag{15}$$

Here the subscripts U, C and D stands for the upwind, central and downwind cell and depend on the flow direction, see figure 2. The subscript f stands for the face between the central and downwind cell. On structured grids the Normalized Variable Approach (NVA) defines the normalized variable $\tilde{\phi}$ as:

$$\tilde{\phi} = \frac{\phi - \phi_U}{\phi_D - \phi_U} \tag{16}$$

Based on this formula one can calculate the normalized values for the three cells:

$$\tilde{\phi}_U = 0 \tag{17}$$

$$\tilde{\phi}_C = \frac{\phi_C - \phi_U}{\phi_D - \phi_U} \tag{18}$$

$$\tilde{\phi}_D = 1. \tag{19}$$

The basic idea of the NVD is to define a scheme depending on $\tilde{\phi}_C$. Now the definition is simplified to

$$\tilde{\phi}_f = f(\tilde{\phi}_C) \tag{20}$$

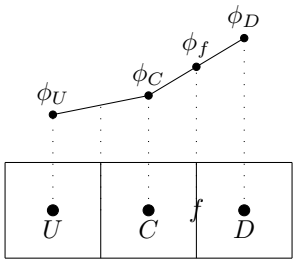


Figure 2: NVD cell notations

but still considers the three neighboring cells. The final face value is obtained by denormalizing $\tilde{\phi}_f$

$$\phi_f = \tilde{\phi}_f(\phi_D - \phi_U) + \phi_U. \tag{21}$$

This principle can be implemented easily for structured grids. For arbitrarily unstructured meshes it is not obvious how to determine the far upwind cell and the concept becomes quite complicated. The calculation of the upwind cell is explained at the end of this section.

Blended Interface Capturing Scheme

The BICS scheme starts from the Inter-Gamma Differencing Scheme (IGDS).

$$\tilde{\phi}_{IGDS} = \begin{cases} \tilde{\phi}_C & \text{if } \tilde{\phi}_C < 0 \\ h_{IGDS} & \text{if } 0 \leq \tilde{\phi}_C < \beta_{IGDS} \\ 1 & \text{if } \beta_{IGDS} \leq \tilde{\phi}_C \leq 1 \\ \tilde{\phi}_C & \text{if } 1 < \tilde{\phi}_C \end{cases} \tag{22}$$

with

$$h_{IGDS} = -\frac{\tilde{\phi}_C^2}{\beta_{IGDS}} + (1 + \frac{1}{\beta_{IGDS}})\tilde{\phi}_C \tag{23}$$

The IGDS is very similar to the base function of the HRIC scheme presented in Muzaferija et al. (1998). For a sharp interface it introduces downwind differences. To fulfill the Convective Boundedness Criterion (CBC) it blends to upwind differences depending on the normalized variable. The scheme parameter β_{IGDS} is usually set to 0.5 to guarantee stability. The IGDS is limited to a local Courant Number of 0.3. Therefore the BICS blends to the Gamma Differencing Scheme (GDS) (Jasak et al., 1999), see equation (24), depending on the local Courant number.

$$\tilde{\phi}_{GDS} = \begin{cases} \tilde{\phi}_C & \text{if } \tilde{\phi}_C < 0 \\ h_{GDS} & \text{if } 0 \leq \tilde{\phi}_C < \beta_{GDS} \\ 0.5\tilde{\phi}_C + 0.5 & \text{if } \beta_{GDS} \leq \tilde{\phi}_C \leq 1 \\ \tilde{\phi}_C & \text{if } 1 < \tilde{\phi}_C \end{cases} \tag{24}$$

with

$$h_{GDS} = -\frac{\tilde{\phi}_C^2}{2\beta_{GDS}} + (1 + \frac{1}{2\beta_{GDS}})\tilde{\phi}_C \tag{25}$$

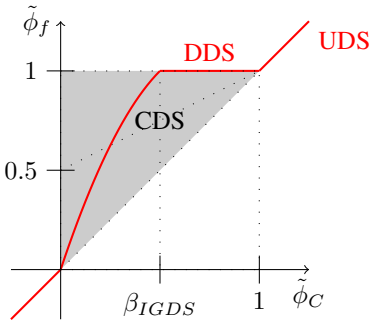


Figure 3: Inter-Gamma Differencing Scheme

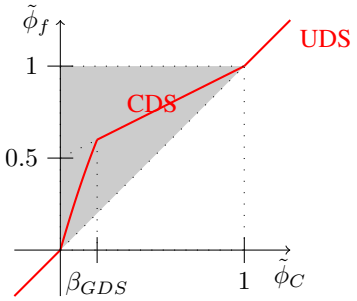


Figure 4: Gamma Differencing Scheme

As one can see in Figure 4 the GDS is second order accurate like the Central Differencing Scheme (CDS) but blends to upwind differences to fulfill the CBC. It has no Courant number limitations. Usually the scheme parameter β_{GDS} is set to 0.1. Lower values may lack stability and higher values will reduce the accuracy. The Courant number dependency of the BICS is given in following equation.

$$\tilde{\phi}_{BICS^*} = \begin{cases} \tilde{\phi}_C & \text{if } \tilde{\phi}_C < 0 \\ h_{BICS^*} & \text{if } 0 \leq \tilde{\phi}_C < \beta_{BICS} \\ p\tilde{\phi}_C + (1-p) & \text{if } \beta_{BICS} \leq \tilde{\phi}_C \leq 1 \\ \tilde{\phi}_C & \text{if } 1 < \tilde{\phi}_C \end{cases} \tag{26}$$

with

$$h_{BICS^*} = -\frac{1-p}{\beta_{BICS^2}}\tilde{\phi_C}^2 + (p + \frac{2(1-p)}{\beta_{BICS}})\tilde{\phi}_C \tag{27}$$

and with the slope p depending on the local Courant number Co, see equation (28), and the scheme parameter β_{BICS} depending on the slope p, see equation (31). The scheme is using a quadratic variation for the part below β_{BICS} and a linear variation for the region above β_{BICS}.

The slope p is calculated by

$$p(Co) = \alpha_p(Co)p_{IGDS} + (1 - \alpha_p(Co))p_{GDS} \tag{28}$$

with

$$\alpha_p(Co) = \begin{cases} 1 & \text{if } Co \leq 0.3 \\ \frac{Co-0.3}{e^{(Co-0.3)}-1} & \text{if } 0.3 < Co \end{cases} \tag{29}$$

and

$$p_{GDS} = 0.5; \quad p_{IGDS} = 0\,. \tag{30}$$

The scheme parameter β_{BICS} is calculated by

$$\beta_{BICS}(p) = a_0 + a_1 p(Co) \tag{31}$$

with

$$a_1 = \frac{\beta_{GDS} - \beta_{IGDS}}{p_{GDS} - p_{IGDS}}; \quad a_0 = \beta_{IGDS} - a_1 p_{IGDS} \tag{32}$$

with

$$\beta_{GDS} = 0.1; \quad \beta_{IGDS} = 0.5 \tag{33}$$

as the standard parameters.

Finally the classical angle correction from the HRIC scheme is applied.

$$\tilde{\phi}_{BICS} = \tilde{\phi}_{BICS^*}(\cos\theta)^{C_0} + \tilde{\phi}_{GDS}[1-(\cos\theta)^{C_0}] \tag{34}$$

where θ is the angle between the normal of the face and the normal of the free-surface. Here, the difference between the BICS and HRIC scheme is that the BICS reduces to the GDS instead of the upwind scheme for the angle correction.

Calculation of the Upwind Cell for the Normalized Variable Approach

For arbitrarily unstructured grids the calculation of the upwind value U is not straight forward. The upwind cell is not known explicitly. One might have multiple cells or no existing cell for the upwind region. This chapter will describe three ways for the calculation of the upwind value.

The classical approach is to extrapolate to an imaginary upstream node U by the use of the gradient projection method.

$$\phi_U = \phi_C - \vec{CU} \cdot \vec{\nabla}\phi_C, \quad with \quad \vec{CU} = -\vec{CD}. \tag{35}$$

Another approach given in Jasak et al. (1999) renounces the upwind value completly and uses a modification of the NVD criterion. Instead of using the upwind and downwind cell, one uses the upwind and downwind face, see figure 5. The new definition of $\tilde{\phi}_C$ is

$$\tilde{\phi}_C = \frac{\phi_C - \phi_f^-}{\phi_f^+ - \phi_f^-} = 1 - \frac{\phi_f^+ - \phi_C}{\phi_f^+ - \phi_f^-}. \tag{36}$$

After some transformation which is explained in detail in Jasak et al. (1999) one obtains the final equation

$$\tilde{\phi}_C = 1 - \frac{\phi_D - \phi_C}{2(\nabla\phi)_C \cdot \vec{CD}}. \tag{37}$$

For both methods shown the extrapolation may lead to values outside the interval [0, 1]. Therefore the values will be bounded to the given interval once they have been calculated. The methods are satisfactory for continuous quantities. For the discontinuous volume fraction these approaches may lead to big errors and stability problems.

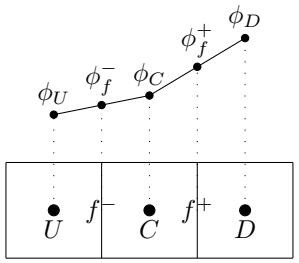


Figure 5: Modified approach for the NVD criterion

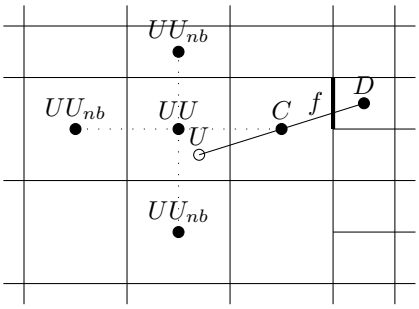


Figure 6: Reconstruction of the upstream node U

Another way of obtaining the upstream node U is based on a reconstruction. It is introduced as an Extension of the BICS in Wackers et al. (2010). The resulting scheme is called BRICS. The difference to the original BICS lies in the reconstruction of the upstream node U as explained below.

The BRICS first searches the cell UU containing the imaginary point U, see figure 6. The search consist of a loop over the neighboring cells of C. The cells UU can be precomputed. If the mesh is not changing the search algorithm has to be called only once. After the cell UU is identified the value for the point U can be calculated with a weighted interpolation based on the cell UU and its neighboring cells UU_{nb}

$$\phi_U = \sum_{N=UU,UU_{nb}} \frac{\phi_N}{||\vec{NU}||} \Big/ \sum_{N=UU,UU_{nb}} \frac{1}{||\vec{NU}||}. \tag{38}$$

Implementation of the High-Resolution Schemes

The High-Resolution schemes allow maximal Cos larger than unity without losing a sharp interface. Indeed the solver lacks stability with higher Cos. To stabilize the solution the scheme is implemented as a deferred correction scheme.

Pressure Gradient Reconstruction

The total pressure p of equation (1) consist of the static pressure $\rho g h$ and dynamic pressure p_d.

$$p = \rho g h + p_d \tag{39}$$

with the z-component g of the gravity vector $\mathbf{g}$ and the z-coordinate h. Substituting the total pressure of (1) with (39) leads to a momentum equation depending on the dynamic pressure

$$\begin{aligned} &\frac{\delta \rho \mathbf{u}}{\delta t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \nabla \cdot \mu_e \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) \\ &= -\nabla(p_d) - gh\nabla(\rho) \end{aligned} \tag{40}$$

Both momentum equations presented here can be used for calculation. The interFoam solver uses equation (40) and OurSolver uses equation (1). The reasons for our decision are explained in this section.

Considering the behavior of pressure and its gradient at the free surface it is not possible to use the dynamic pressure for the momentum equation. Doing this will lead to a term containing the gradient of the dynamic pressure and a gradient of the density, see equation (40). The pressure gradient has a jump at the free surface and the density gradient is undefined for a sharp interface. Still the dynamic pressure is used for the original interFoam solver.

Numerical smearing at the interface avoids the jump behavior. For cells containing the lighter fluid the error is especially large. For density ratios typical for water and air the values could be around 500 times larger. These wrong values lead to unphysical high velocities in the free surface cells containing the lighter fluid. Having a satisfying sharp interface, due to a High Resolution Scheme and/or due to grid refinement, will enforce this behavior. The velocity overshoots decrease the stability of the solver. Often, simulations with grids fulfilling the requirements for a sufficient sharp interface diverge using the standard interFoam solver. Sometimes a solution is possible with adjusted solver settings (smaller time step, more iterations) but then the solver is slow compared to other well known commercial solvers. One solution is to limit the velocities to a user specified value every time it is calculated. This leads to a more stable solver but still does not address the root cause. Therefore we have implemented a method for the reconstruction of the pressure and its gradient at the face for arbitrary unstructured grids given in Queutey et al. (2007).

Four equations, the momentum equation (1), the pressure equation (10), the velocity equation (9) and the equation for the flux (14), contain the surface normal gradient of the pressure which requires the reconstruction. This gradient is on the explicit right hand side three times and one time (for the pressure equation) it is in the implicit part.

The reconstruction is explained in detail in Queutey et al. (2007) and only the important parts of our implementation are described here. The method assumes that the free surface Γ is exactly at the face f. It requires a known function, which is continuous on each subdomain Ω^+ and Ω^- and discontinuous across the interface. Here the density fulfills these requirements, if a High-Resolution scheme is used for the discretisation of the volume fraction. The reconstruction is only possible using the total pressure. Otherwise in case of dynamic pressure we will get the undefined density gradient at the free surface. This is the reason why our solver is using the total pressure and not the dynamic pressure like interFoam.

Reconstructing $\left(\frac{\nabla p \cdot \mathbf{n}}{\rho}\right)_f$ instead of $(\mathbf{n} \cdot \nabla p)_f$ is the only solution. The equation given for the surface normal gradient is

$$\left(\frac{\nabla p \cdot \mathbf{n}}{\rho}\right)_f = \frac{1}{\hat{\rho}} \frac{p_R - p_L}{h} \boxed{+ \frac{\nabla p_L \cdot \mathbf{e}^- + \nabla p_R \cdot \mathbf{e}^+}{\hat{\rho} h}} \tag{41}$$

with the explicit distance vectors $\mathbf{e}^+$ and $\mathbf{e}^-$ as shown in figure 7 for the correction of grid non-orthogonality. The boxed term is treated explicitly. The equation given for the face value is

$$\begin{aligned} p_f &= \frac{h^+ \rho^+ p_L + h^- \rho^- p_R}{h^+ \rho^+ + h^- \rho^-} \\ &\boxed{+ \frac{\rho^+ \rho^-}{\hat{\rho}} \left(\frac{h^- \mathbf{e}^+ - h^+ \mathbf{e}^-}{h}\right) \cdot \left(\frac{h^+}{h}\left(\frac{\nabla p}{\rho}\right)_L + \frac{h^-}{h}\left(\frac{\nabla p}{\rho}\right)_R\right)} \end{aligned} \tag{42}$$

with the face value $\hat{\rho}$ calculated by a reversed linear interpolation

$$\hat{\rho} = \frac{h^-}{h}\rho^- + \frac{h^+}{h}\rho^+ \tag{43}$$

Again, the boxed term is treated explicitly. It is not obvious how to calculate the explicit, boxed part of equation (41). The pressure gradients can be calculated using the Gauss Theorem $\int_v \nabla \phi dv = \int_s \phi d\mathbf{s}$. The Gauss Theorem requires the face values. Using the linear interpolated face values is possible, but numerical tests showed a small negative influence on to the result. Therefore we use the reconstructed face values calculated with (42). But, Equation (42) itself requires the cell gradients (calculated with the Gauss Theorem). Thus, it is a coupled system of equations and the implementation is not straightforward. Here it is implemented with the following iterative approach:

1. Calculate p_f^n using ∇p^{n-1}
2. Calcuate ∇p^n using p_f^n
3. Limit ∇p^n
4. Update iterator $n = n + 1$. Go to step 1

The limitation at Step 3 is essential for convergency. Unfortunately this limitation has a significant influence on the

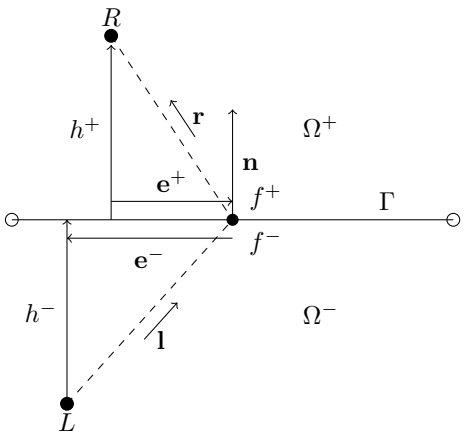


Figure 7: Notations for discontinuity reconstruction

computational time. Numerical tests with different numbers of iterations were done and showed that one iteration only is enough for a steady state result.

The final implementation consist of only one iteration and calculates the reconstructed face value with the old pressure gradient. Since there are outer iterations of the SIMPLE-algorithm there is still an iterative process updating the face value and cell gradient. It should be mentioned that a method without this gradient reconstruction also requires a limitation of the pressure gradient for sufficient stability.

For the reconstruction the flow equations have to be modified as follows (modifications are highlighted in red):

Momentum equation:

$$\begin{aligned}&\frac{\rho}{\Delta t}\mathbf{u}_c + \frac{1}{V}a_c\mathbf{u}_c + \frac{1}{V}\sum_n a_n\mathbf{u}_n \\ &= \mathbf{q} + \rho\mathbf{g} - \rho\mathscr{R}((\nabla p)_f \cdot \mathbf{s}_f/\hat{\rho}) + \frac{\rho}{\Delta t}\hat{\mathbf{u}}_c\end{aligned} \tag{44}$$

Pressure equation:

$$\mathscr{D}\left(\frac{\mathscr{I}(\rho)}{\hat{\rho}}\alpha_f^{-1}\nabla p\right) = \mathscr{D}\left(\alpha_f^{-1}\left(h_{tf} + h_{sf}\right)\right) \tag{45}$$

Velocity equation:

$$\begin{aligned}&\mathbf{u}_c = \left(\frac{\rho}{\Delta t} + \frac{1}{V}a_c\right)^{-1} \\ &\left(\mathbf{q} + \rho\mathbf{g} - \rho\mathscr{R}((\nabla p)_f \cdot \mathbf{s}_f/\hat{\rho}) + \frac{\rho}{\Delta t}\hat{\mathbf{u}}_c - \frac{1}{V}\sum_n a_n\mathbf{u}_n\right)\end{aligned} \tag{46}$$

Flux:

$$\Phi = \frac{\mathbf{s}_f}{\alpha_f} \cdot \left(h_{tf} + h_{sf} - \frac{\mathscr{I}(\rho)}{\hat{\rho}}(\nabla p)_f\right) \tag{47}$$

Motion

An important part of the drag prediction for yachts is the correct determination of flotation. While flow around a yacht is usually simulated in an Eulerian coordinate system, the yacht moves with respect to this coordinate frame. The motion itself depends on fluid forces as well as external forces and moments either fixed or resulting from fluid forces. To correctly capture the behavior of highly dynamic vessels (e.g. if foil-supported), time accuracy is of utmost importance. As the *sixDoFRBM*-Method present in OpenFOAM has been shown to be unsuitable (see Devolder et al. (2015)), a robust method has been implemented.

Time Integration

The computation of linear acceleration is based on a fully dynamic model with acceleration computed from instantaneous force:

$$\mathbf{f} = \mathbf{M} \cdot \mathbf{a} \tag{48}$$

Time integration to compute velocities and position is achieved by the trapezoidal rule (see Hadžić et al. (2005)):

$$\mathbf{v}_{n+1}^{i+1} = \mathbf{v}_n + \frac{\mathbf{f}_n + \mathbf{f}_{n+1}^{i+1}}{2m}\Delta t \tag{49}$$

$$\mathbf{r}_{n+1}^{i+1} = \mathbf{r}_n + \frac{\mathbf{v}_n + \mathbf{v}_{n+1}^{i+1}}{2}\Delta t \tag{50}$$

Similar equations hold for rotational motion.

To achieve (semi-) implicit coupling between flow and motions, the solution of above equations is integrated into the SIMPLE- or outer loop of the CFD solver (see Figure 1).

Stability

While implicit trapezoidal integration of the equations of motion could be considered to be generally stable, some decoupling between flow and motion takes place due to the quasi-explicit integration in the outer loop. This decoupling gives rise to a virtual added mass effect (see Förster et al. (2007)), potentially destabilizing the solution if virtual added masses are equal to or larger than the vessel's mass. Hadžić et al. (2005) have shown that a per iteration under-relaxation of velocities is comparable to the application of an added mass to the force defect between iterations. When the outer loop converges, the effect of this under-relaxation on results vanishes.

While this approach is simple and robust, if the under-relaxation factors are well chosen, even it does not guarantee optimal convergence behavior. More efficient approaches (e.g. Soeding (2001)) are currently under investigation.

To improve convergence to steady state if transient behavior is not of interest, velocity-proportional damping can be applied to the motion.

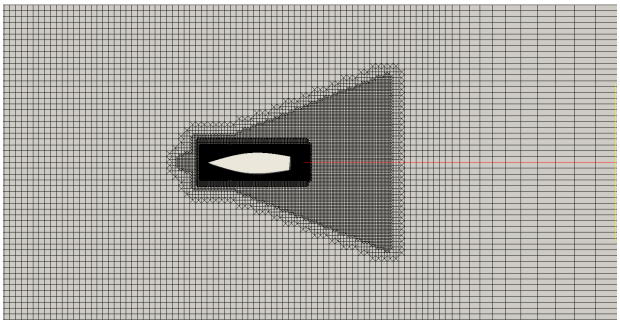


Figure 8: Sysser60 grid top view

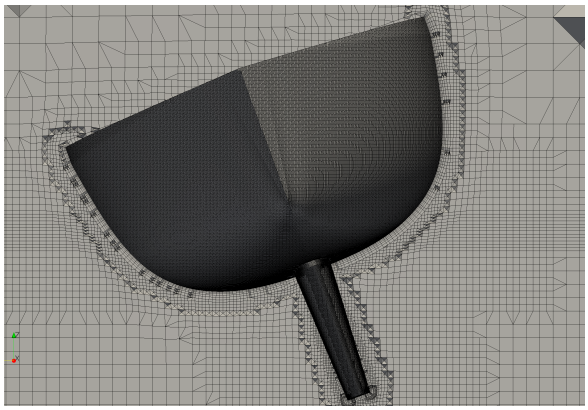


Figure 9: Sysser60 grid front view

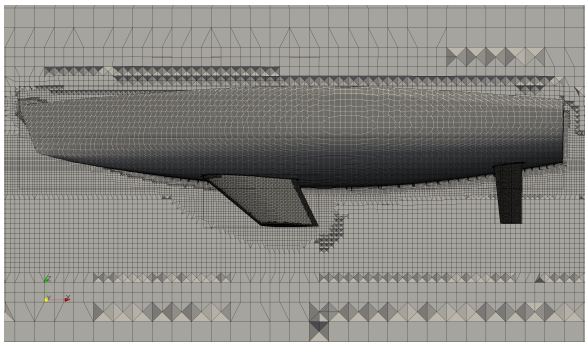


Figure 10: Sysser60 grid side view

External Forces

To properly include the effects of sail or propulsive forces and moments these can be applied at any given point (e.g. CoE), either as fixed values or depending on flow forces and moments. The point of attack and, if required, the direction are defined in boat-fixed frame of reference.

VALIDATION

Sysser60

Description

In order to compare our new solver with interFoam (of OpenFOAM 2.4.x) and the commercial solver Star-CCM+ v9.06 the Sysser 60 test case of the Delft Systematic Yacht Hull Series of Delft University of Technology (2013) has been chosen. The simulations are carried out without motion to better distinguish between phenomena induced by flow solver and motion solver. Particular focus during this test was on the phenomenon of numerical ventilation. To evaluate the solvers capabilities in this regard a flotation with the bow knuckle above water was chosen. The simulation is done in model scale. The hull has a waterline length of $L = 2.16m$. The flow speed is 1.806 m/s corresponding to a Froude no. of 0.39. The hull is heeled by 20° and yawed by 3°.

Figure 8 to 10 show the grid. It was generated with snappyHexMesh (SHM) and has 2.7 million cells. The waterline was refined anisotropically in the z direction. The cells at the outlet were stretched to avoid reflections of the waves. Figure 8 only shows a part of these stretched cells. The domain has an length of $3L$ in front of the hull and $3L$ behind the hull before stretching the cells at the outlet. The height is $1.5L$ below waterline and $1.0L$ above waterline. In the area of the Kelvin wave pattern additional refinement is applied up to a length of $1.4L$ behind the hull The domain is $4.3L$ wide. The grid was exported to Star-CCM+ so that all simulations are done on exactly the same grid.

Solver Setups

The interFoam solver uses the same algorithm as shown in figure 1. For the VOF-method an additional compressive term is used to keep a sharp interface. The equation is solved using the multi-dimensional limiter for explicit solution (MULES). It uses an implicit predictor based on upwind differencing for the convective term. Afterwards the solution is corrected with an explicit corrector using the vanLeer discretization scheme. Despite a lot of time consuming tests it was not possible to get a stable setup for the given grid with the original interFoam solver. Therefore a function for the limitation of the velocity has been added. Each time the velocity changes, the cell velocities are limited to a user specified value. In this case the velocity is limited to $15\frac{m}{s}$. In fact this modification harms the conversation of mass. But tests have shown that this has no influence onto the results. For clarity this solver is called interFoamMod in this paper. Similar to OurSolver, interFoam allows the use of the α-subcycling technique. This method allows subdivision of the timestep in smaller subtimesteps for the solution of the VOF-transport equation which improves more stability and accuracy.

Star-CCM+ also uses an algorithm similar to figure 1. It is not known if any additional iterations for the correction of

mesh non-orthogonality are applied. Also, it is not possible to prescribe any kind of α-subcycling. The HRIC scheme is used for the convective term of the VOF-equation. To supress numerical ventilation the blend to upwind differencing has been switched off as recommended in Boehm et al. (2014) for steady state simulations.

All three solvers are adjusted to the same settings as close as possible, see Table 1. The goal was to use a time step of $\Delta t = 0.02s$, 5 outer iterations, 1 Piso iteration, 1 additional iteration for the correction of non-orthogonality and no alpha-subcycling. The averaged Co is 0.123. Although the modified solver interFoamMod has been used, additional Piso iterations and α-subcycles were necessary to get sufficient solver stability here. For the sake of clarity, this is only done to get solver stability. Nevertheless additional iterations and subcycles generally have a positive influence onto the quality of the result.

In all simulations the k-ω-SST was chosen. A total time of $30s$ is simulated.

Star-CCM+ is using some unknown method to suppress an unphysical splash wave at the beginning of the simulation. For OurSolver the velocity is ramped over the first $0.2s$ to avoid the same unphysical behavior. The simulations were also done without ramping, showing that the ramping is not required for solver stability.

Wave pattern

The transversal wavecuts of figure 11 are obtained by cutting the free-surface with an increment of $\frac{x}{L} = 0.1L$. The longitudinal wavecuts of figure 12 are obtained by cutting with an increment of $\frac{y}{L} = 0.05L$. The wavecuts show good agreement in the region close to the hull. At the bow only the Star-CCM+ simulation produces a different result with three breaking waves which is discussed later in this section. Both figures show that the waves of the interFoamMod simulation are increasingly damped the further from the hull. Other simulations (not shown here) with a much longer Kelvin refinement of about $6L$ behind the hull prove this behavior. Independent of the refinement, the interFoamMod waves are much flatter far from the hull. This emphasizes that the discretization of the convective term of equation (5) requires a High Resolution scheme for a correct convection of the waves. Considering future simulations including ocean waves this is an important point to avoid damping of these waves. Figure 13 also illustrates the damping characteristics of the interFoamMod simulation. It is an important point that the interFoamMod simulation is using five alpha-subcycling due to stability reasons whereas both other solvers do not use any additional subcycles. Subcycling supports the accuracy of the simulation for both wave sharpness and wave convection. Despite this, interFoamMod is less accurate.

The waves at the bow of the Star-CCM+ simulation are significantly different than the results of OurSolver and interFoamMod. Figure 14 shows that Star-CCM+ generates a wave breaking three times. The other two solvers have one to one and a half breaking waves depending on the chosen timestep. Increasing the cell size to 11 million cells has no effect on this behavior and the wave count is still the same for OurSolver and interFoamMod. Using the finer grid for Star-CCM+ boosts this behavior to four clearly breaking waves. To the authors knowledge the four waves of Star-CCM+ have not been observed for real flow. As Star-CCM+ is a closed code, we do not have any knowledge about the implemented models beyond the description in the theory guide. Finally this phenomenon might need additional investigations.

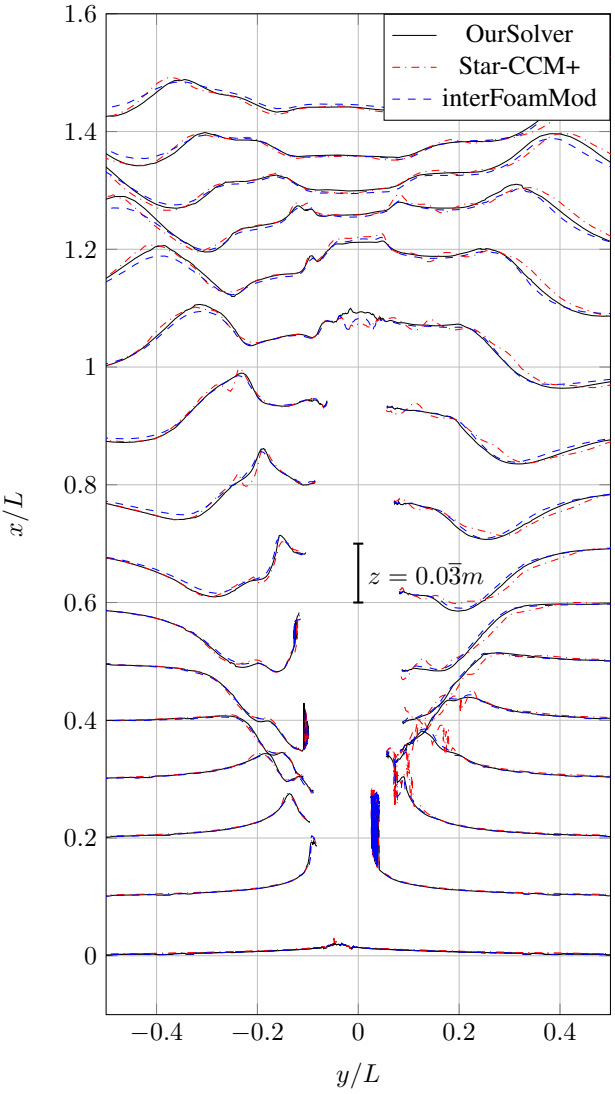


Figure 11: Sysser60 transversal wavecuts

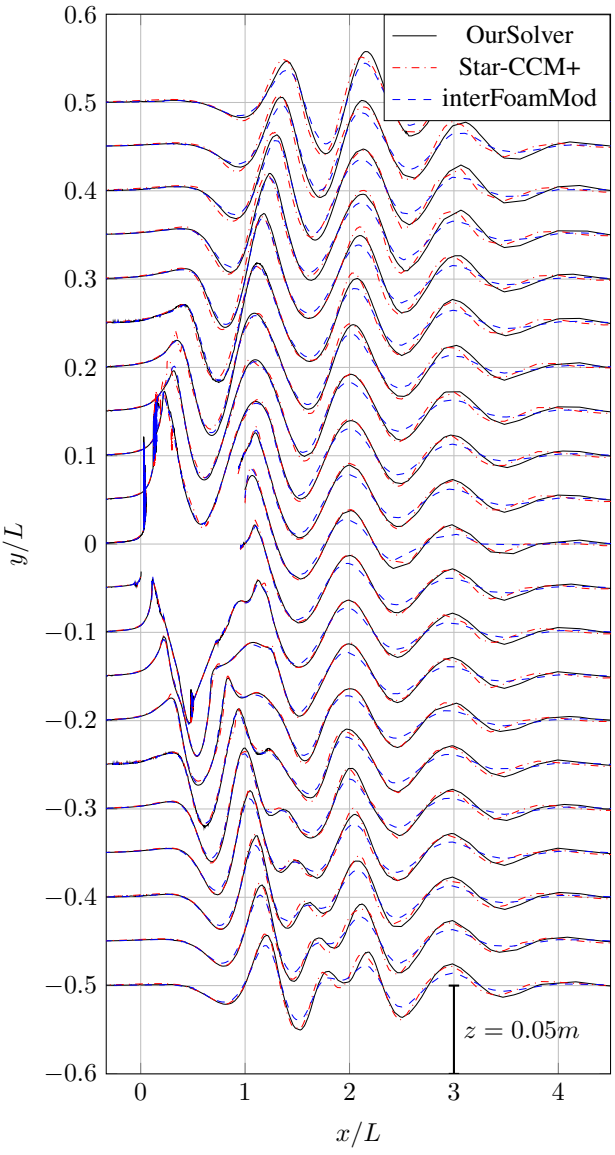
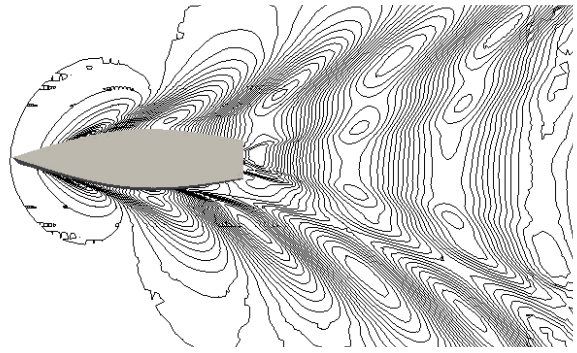
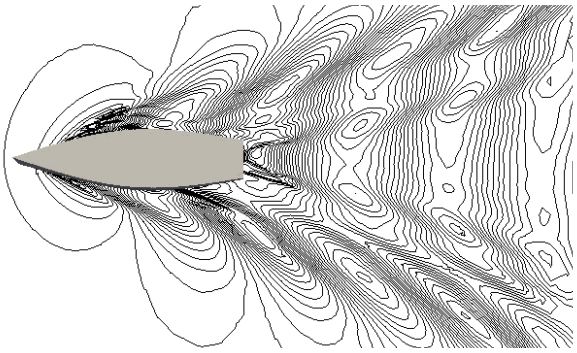


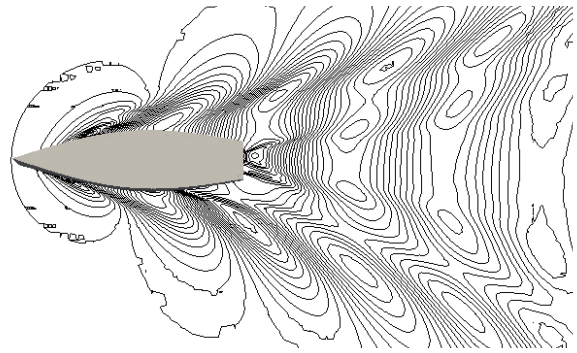
Figure 12: Sysser60 longitudinal wavecuts



(a) OurSolver

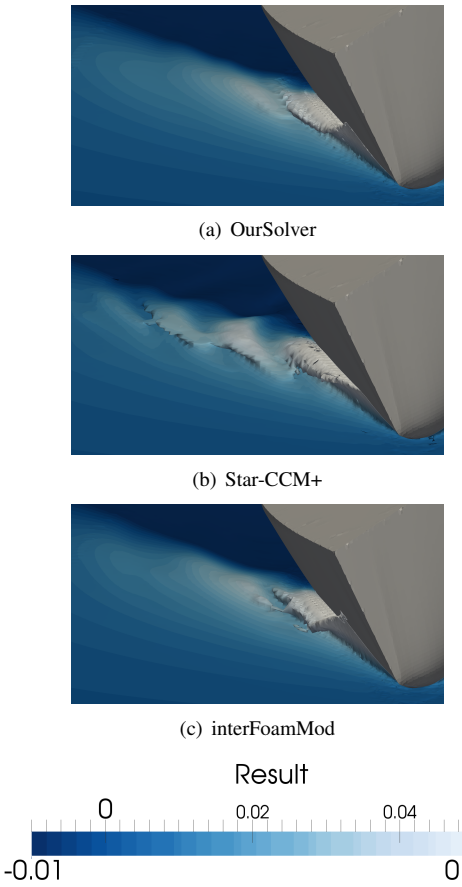


(b) Star-CCM+



(c) interFoamMod

Figure 13: Wavepattern of the Sysser60 test case

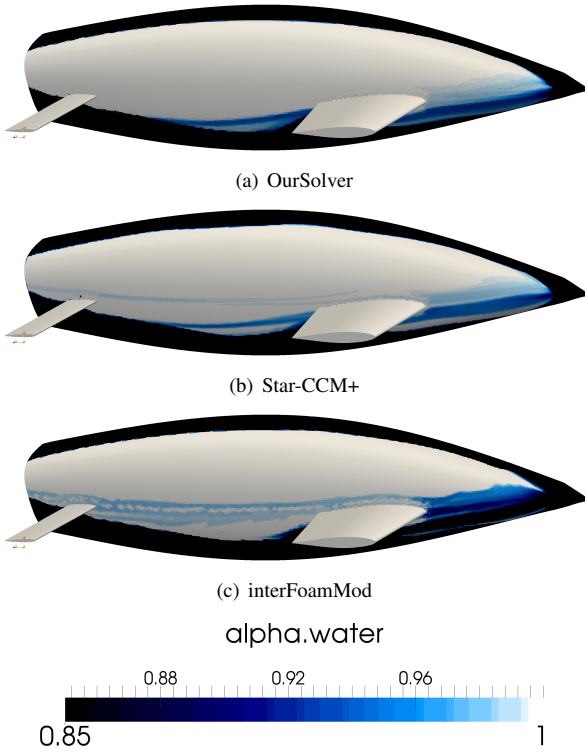


(a) OurSolver

(b) Star-CCM+

(c) interFoamMod

Figure 14: Waves at the bow of the Sysser60 test case



(a) OurSolver

(b) Star-CCM+

(c) interFoamMod

Figure 15: Numerical ventilation of the Sysser60 test case

Numerical ventilation

The amount of Numerical Ventilation (NV) is shown in figure 15 plotting the volume fraction on the hull. For a distinct differentiation the volume fraction is scaled from 0.85 to unity. While the results of OurSolver and Star-CCM+ are quite similar the interFoamMod simulation produced much more air under the hull. Figure 15 is not sufficient for an reliable predication of NV. The NV changes with the time due to a periodic oscillation of the flow and the chosen timestep might have a significant influence on the result.

To estimate the NV equation (51) is introduced.

$$N = \frac{\sum_{f=1}^{n} \lambda_f |\mathbf{s}_f||(1-\alpha_w f)|}{\sum_{f=1}^{n} \lambda_f |\mathbf{s}_f|} \tag{51}$$

where $\mathbf{s}$ is the face area vector of the face f, n is the number of faces defining the hull patch, α_w is the volume fraction of the water and λ is a switch depending on the pressure

$$\lambda_f = \begin{cases} 1 & \text{if } p_f > p_l \\ 0 & \text{if } p_f \leq p_l \end{cases} \tag{52}$$

Where the limiting pressure p_l is set to $150\frac{N}{m^2}$. The pressure switch excludes faces above and at the free surface, respectively. Therefore physically correct spray does not have a negative effect on the assessment of the NV. This equation does not produce an absolutely correct evaluation but allows a rough approximation.

Figure 16 shows the NV plotted over time. It shows that the result of interFoamMod has twice the NV compared to OurSolver and Star-CCM+.

Velocity overshoots

Figure 17 shows the maximum velocities for each timestep. The maximum velocity of the interFoamMod simulation is mostly at the prescribed limit of 15m/s. The results of OurSolver and Star-CCM+ are physically more correct. OurSolver only reaches the limiting velocity for the first timesteps. Star-CCM+ does not have such limitation and only exceeds this value for the first timesteps and at $t = 16$s. Overall the maximal velocity of Star-CCM+ is smaller than OurSolver. Concluding it can be said that the velocities of both solvers lie in a reasonable range whereas the interFoamMod velocities have unacceptably high values and never drop down to plausible values. For the sake of clarity, the velocity limitation is not included in the original interFoam solver. Without this limitation the velocities would reach much higher values and the solution will diverge until the solver crashes in the first timesteps.

The simulation with OurSolver has been done with and

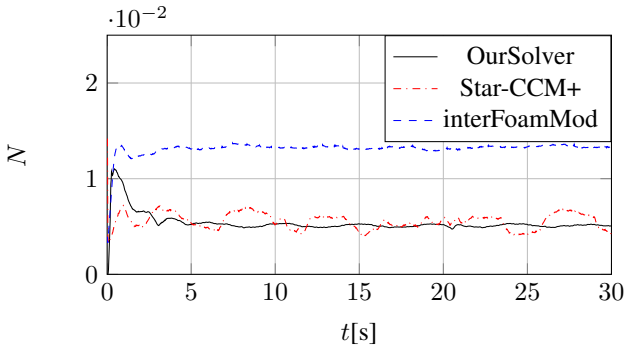
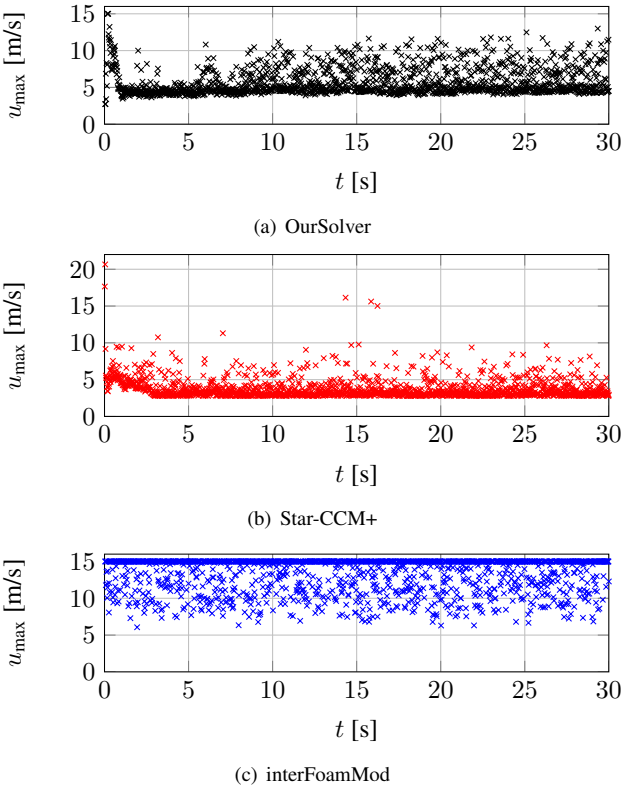


Figure 16: Numerical ventilation

without pressure reconstruction. Without pressure reconstruction the maximal velocities have the same characteristic as the velocities of the interFoamMod simulation. For this test case, besides the velocity overshoots the result looks almost identical to the result of the simulation with reconstruction. Thus, in this case the reconstruction is only necessary for the stability of the solver. It delivers a physically correct solution of the velocity overshoot problem compared to a hard limitation. Additionally it does not require a user-specified value. This simplifies automated set-ups and more complex cases.



(a) OurSolver

(b) Star-CCM+

(c) interFoamMod

Figure 17: Velocity overshoots

Computation Time

For the comparison of the computational time the case was varied in cell size and time step. The computations were done on a node with 16 cores at 2.6Ghz. Only 15 cores were used to have enough power left for background processes of the machine. Different nodes with identical construction have been used for all simulations. Overall, this is not a scientific investigation on the computational time and the time needed for the same simulation might vary about 20%. Still,the results given in Table 1 show a significant trend. The table demonstrates the solver capabilities for real engineering situations where grid or timestep studies are required.

For all grids calculated with the smallest timestep of $\Delta t = 0.02$ OurSolver is about one third slower than Star-CCM+. For the timestep variation OurSolver needs about twice the time of Star-CCM+. The reason could be the additional non-orthogonal correction iteration and the settings for the absolute residual which breaks the inner iterations of the linear solvers. Also, OurSolver is compiled with g++ which might be up to 20% slower for specific tasks than other well known compilers.

To facilitate the calculation with the interFoamMod solver its settings had to be adjusted. These adjustments are the major reason for the significant slower computations. For the grid variation the solver needs two to six times the Star-CCM+ calculation time. For the time step variation it needs up to 33 times of the calculation time. Increasing the timestep should decrease the calculation time. The cases with $\Delta t = 0.06$ and $\Delta t = 0.08$ show that this is not always the case for interFoamMod just because of stability reasons. Additionally it has to be emphasized that a typical grid or timestep study is based on scripting the simulations and then running all in one chunk. Here a lot of failing runs were necessary to get a setup with acceptable calculation time for each variation. Of course this is an unacceptable workflow. The 8.3 million cells test case forced a decrease in the timestep to $\Delta t = 0.01$. A grid with 10.8 cells has been tested, also. OurSolver and Star-CCM+ run as stable as before without modifying any settings. For interFoamMod the settings were adjusted until a simulation time of 2 weeks was reached, but the simulation still crashed.

Immersed Transom Study

Farr Yacht Design, in association with the *Sailing Yacht Research Foundation*, kindly released the results of a towing tank study on the impact of transom immersion respectively displacement and LCG on the resistance of a canonical VO70 yacht into public domain (see Farr Yacht Design (2005)). Within this study the transom immersion at rest was varied by adding weights to the yacht and sail trimming moment was applied by shifting the weights according to standard towing tank practice. The experiments were car-

Table 1: Simulation time on 15*2.6GHz for different case setups

	Δt [s]	nCells [E6 cells]	θ [ ]	κ [ ]	ξ [ ]	ϵ [ ]	Computation Time [h]	Deviation to Star-CCM+ [%]
Star-CCM+	0.02	2.7	5	1	1	-	8.39	0.0
OurSolver	0.02	2.7	5	1	1	1	10.71	+27
interFoamMod	0.02	2.7	5	3	5	1	27.95	+233
Star-CCM+	0.04	2.7	5	1	1	-	2.68	0.0
OurSolver	0.04	2.7	5	1	1	1	5.92	+121
interFoamMod	0.04	2.7	5	3	10	1	22.53	+772
Star-CCM+	0.06	2.7	5	1	1	-	1.77	0.0
OurSolver	0.06	2.7	5	1	1	1	2.92	+64
interFoamMod	0.06	2.7	10	3	15	1	42.51	+2301
Star-CCM+	0.08	2.7	5	1	1	-	1.24	0.0
OurSolver	0.08	2.7	5	1	1	1	2.41	+94
interFoamMod	0.08	2.7	10	3	20	1	42.07	+3392
Star-CCM+	0.02	4.5	5	1	1	-	14.91	0.0
OurSolver	0.02	4.5	5	1	1	1	18.91	+26
interFoamMod	0.02	4.5	5	3	5	1	48.23	+223
Star-CCM+	0.02	8.3	5	1	1	-	29.94	0.0
OurSolver	0.02	8.3	5	1	1	1	40.21	+34
interFoamMod	0.01	8.3	5	3	5	1	213.12	+611

θ = number of outer iterations, κ = number of Piso corrector iterations,
ξ = number of α subcycles, ϵ = number of non-orthogonal correction iterations

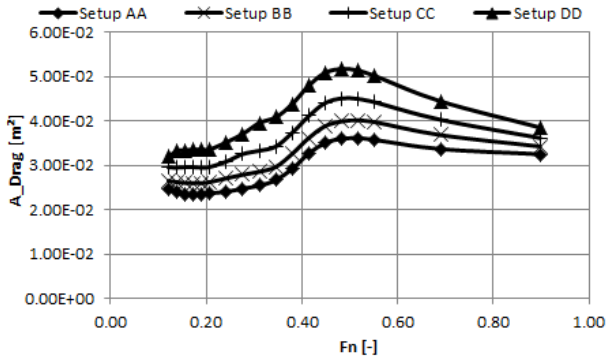


Figure 18: Resistance of canonical VO70 design (Towing tank results)

ried out at a scale of 3.2, and four basic flotations were tested at Froude numbers from 0.121 to 0.898. All tests were run in upright condition without yaw. Experimental results are given in Figure 18, the mid-forward (BB) and mid-aft (CC) LCGs are selected for validation.

Simulation Setup

The flow around the yacht is simulated at selected velocities corresponding to Fn from 0.345 upwards for LCG / displacement combinations denoted BB and CC. As only upright resistance is of interest only one half of the yacht is modeled with a domain size of 1 length ahead, 3 astern, 2 aside, 1.5 below and 0.5 above the yacht. The computational mesh is generated using a combination of snappyHexMesh (SHM) and related utilities, resulting in a domain size of about 1.8E+06 cells. Analogous to the experiments the only appendages considered are keel and bulb. The free surface region perpendicular to the undisturbed free surface is resolved down to a thousandth of length over a total height of 3% of length.

An incident turbulence level of 1% is prescribed with turbulence modeled by the kω-SST-model using wall functions. Average y^+ is about 50. Velocity is ramped over 5s, body motion is released at 1s with forces and moments ramped over another second.

Results

Figure 19 gives a comparison of drag areas (drag divided by dynamic pressure head) for the evaluated cases. As can

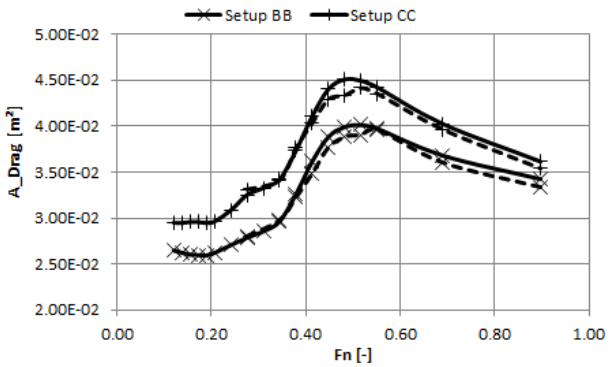


Figure 19: Drag areas from simulation and experiment

be seen, the agreement is very good with maximum deviation of about 3% (turbulence stimulation in experiment is not accounted for in the diagram). In particular, the effect of the LCG and displacement variation is correctly captured. Figures 20 to 22 give a comparison of near hull wave patterns from simulation and tank test photographs at selected velocities. Here as well, good agreement can be observed.

APPLICATION

A good example to show the code's capabilities is the simulation of the flow around a VO65OD yacht (geometry kindly provided by *Farr Yacht Design*, see figure 23). This particular design features an extensive set of appendages, comprising canting keel, twin dagger boards and twin rudders. The keel root and canting axis are located in a recess on the hull. At typical operating conditions the leeward daggerboard is lowered, the windward rudder piercing the free surface and the keel is fully canted to windward with the recess at the free surface. Due to the inclined keel pin the canted keel generates significant amounts of vertical lift, especially at larger Froude numbers, having significant impact on the wave pattern and resulting in strongly dynamic behavior of the boat.

Case Setup

The flow around the yacht is simulated at full scale at a velocity of 17kts (Fn = 0.63), leeway angle of 5°, heel angle of 26°. Domain size is 1 length in front, 3 astern, 2 aside, 1.5 below and 0.5 above the yacht with a body fitted split cartesian mesh of 6.96E+06 cells generated using SHM. The capabilities of SHM and associated utilities, using proper settings and scripting, to provide appropriate free surface resolution and capture complex geometries can be seen in figures 24 and 25. Velocity is ramped over 5s (total simulation time 20s), body motion is released at 1s with force / moment ramping over 2s. Sail forces / moments depending on resistance and side forces are taken into account by an appropriate model.

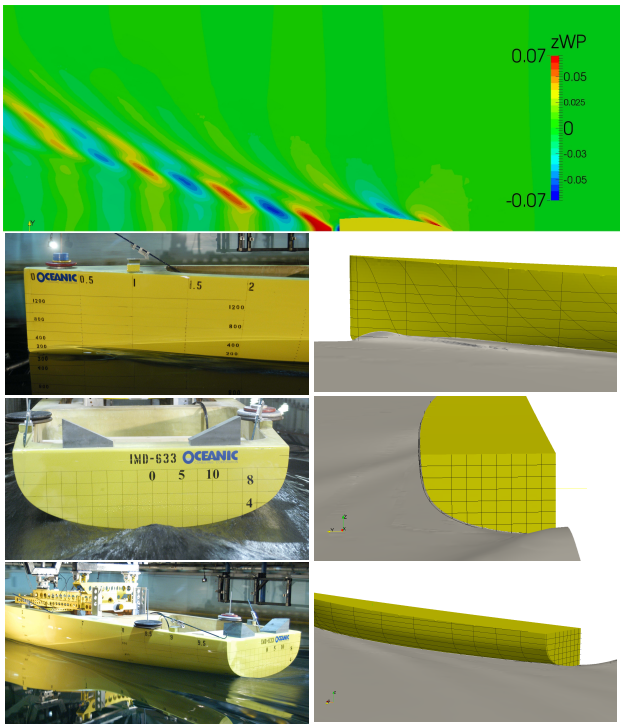


Figure 20: Wave pattern and near hull waves at Fn = 0.345

Results

Figure 26 shows the global wave field generated by the yacht. Even here the impact of the appendages on the hull-generated wave system can already be clearly seen. Figures 27 to 31 give closer views of flow features of particular interest. The breaking bow wave with rising sheet of water as well as the wave systems generated by the keel and recess close to the free surface and the surface-piercing rudder can clearly be seen.

Of particular interest for the correct prediction of resistance and trim is the volume fraction underneath the hull. In many simulations using various codes significant amounts of air can be observed here (aka numerical ventilation or streaking), strongly affecting viscous as well as pressure forces and moments. As can be seen in figure 32, this numerical phenomenon is successfully suppressed in this implementation.

Figures 33 and 34 show the evolution of forces and body motion. The rise of resistance, side force, yawing and righting moment as well as the evolution to a steady flotation state are clearly observable.

CONCLUSION

A new OpenFOAM-based solver for the calculation of free surface flow based on state of the art methods has been described. The important changes to the standard OpenFOAM solvers interFoam / interDymFoam have been presented. A modern discretisation scheme for the convective term of the

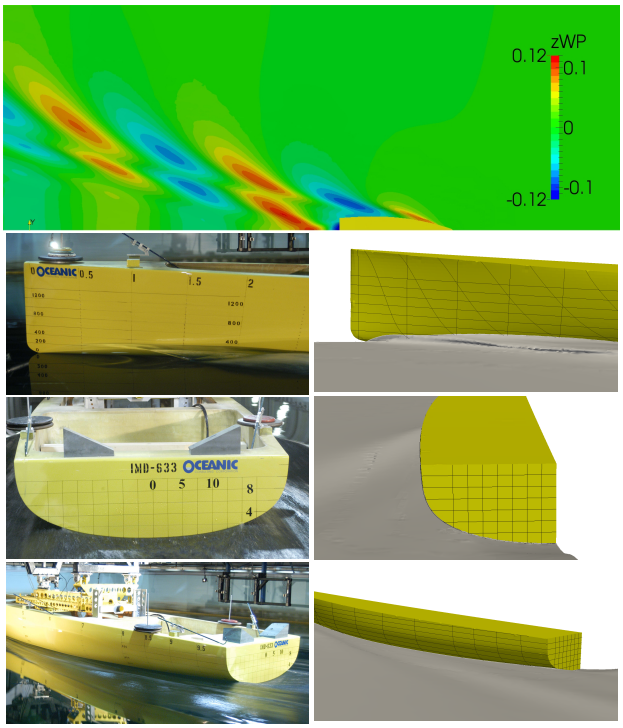


Figure 21: Wave pattern and near hull waves at Fn = 0.484

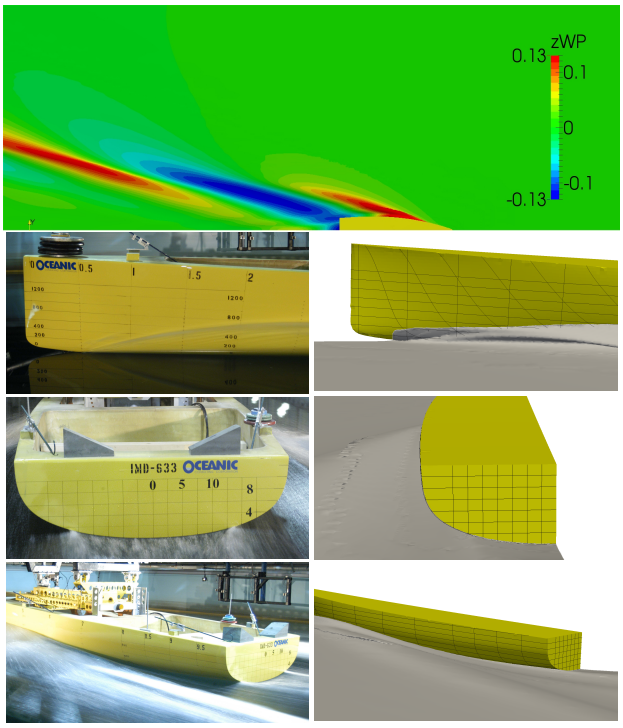


Figure 22: Wave pattern and near hull waves at Fn = 0.898

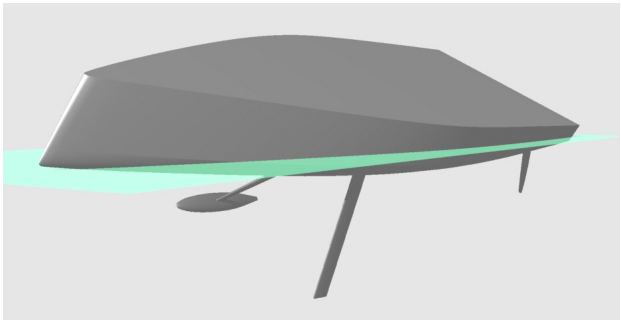


Figure 23: Rendering of VO65 OD

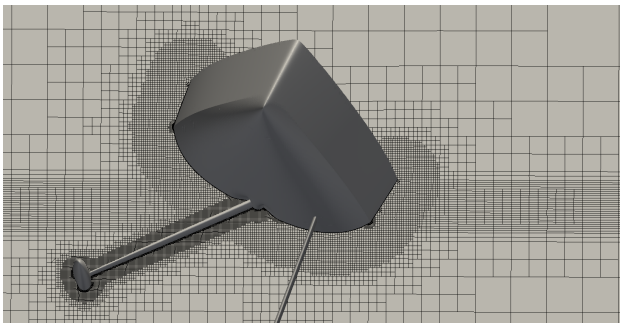


Figure 24: Global and free surface resolution

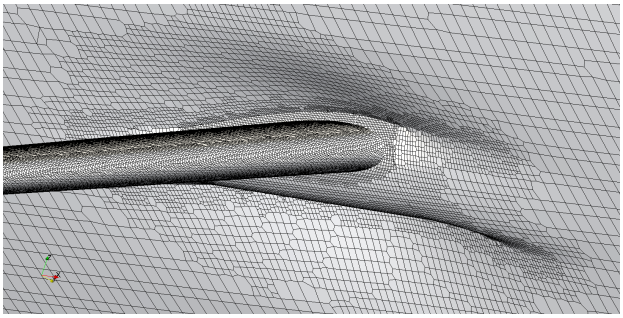


Figure 25: Resolution of keel detail

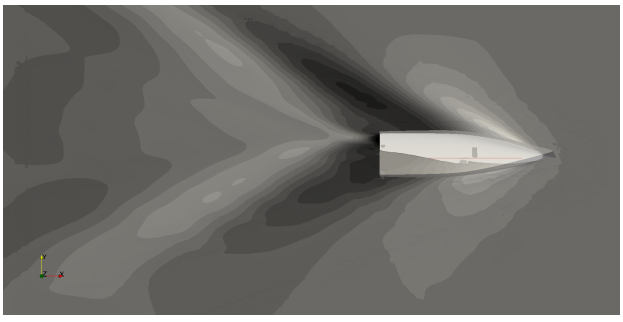


Figure 26: Global wave pattern, weather side at bottom of picture

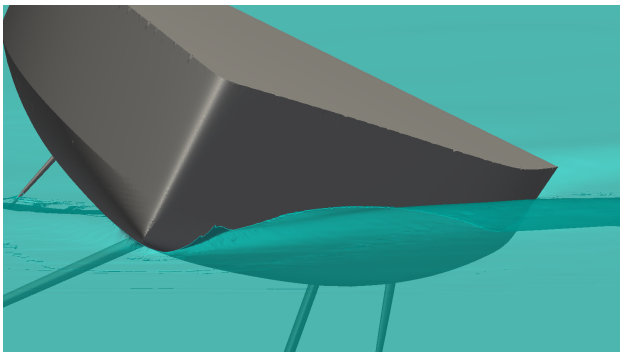


Figure 27: Bow wave system, leeward

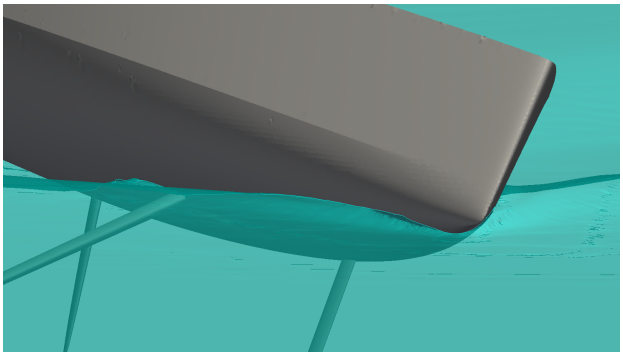


Figure 28: Bow wave system, windward

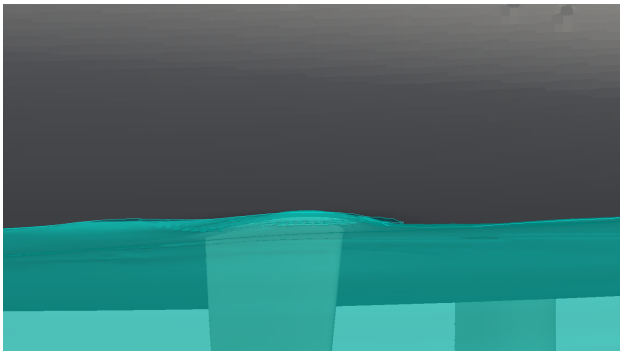


Figure 29: Local free surface at keel root (above)

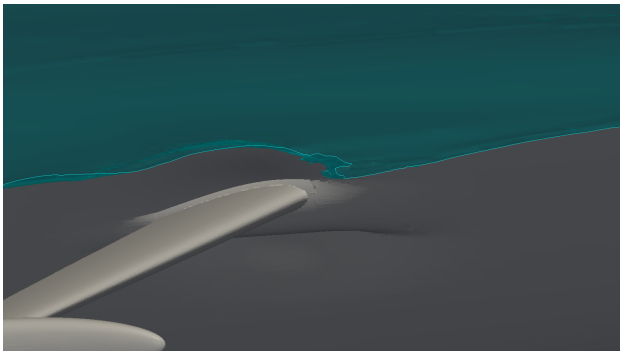


Figure 30: Local free surface at keel root (below)

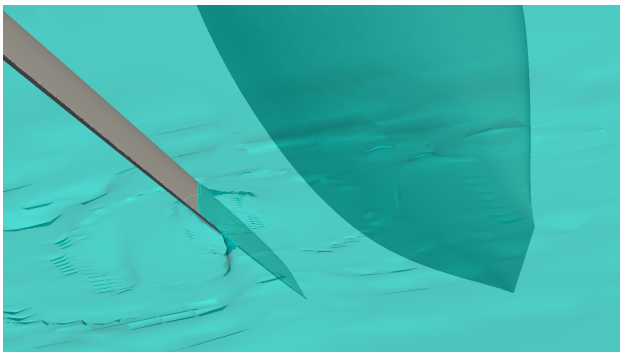


Figure 31: Local wave system from windward rudder

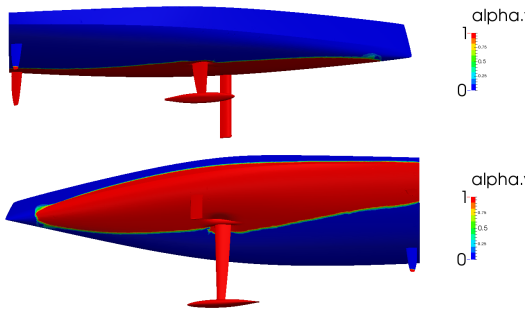


Figure 32: Volume fraction below hull

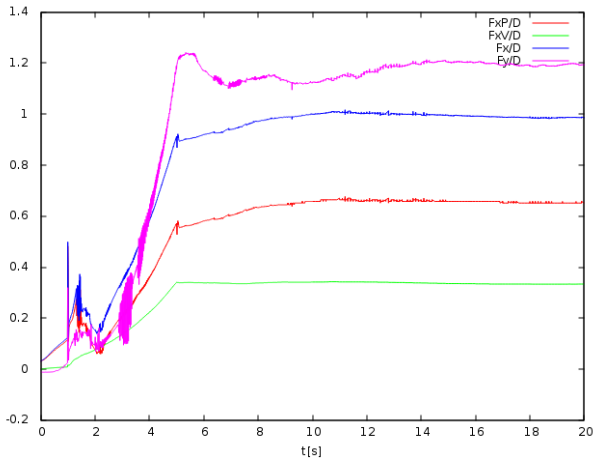


Figure 33: Evolution of resistance and side forces divided by displacement over simulated time

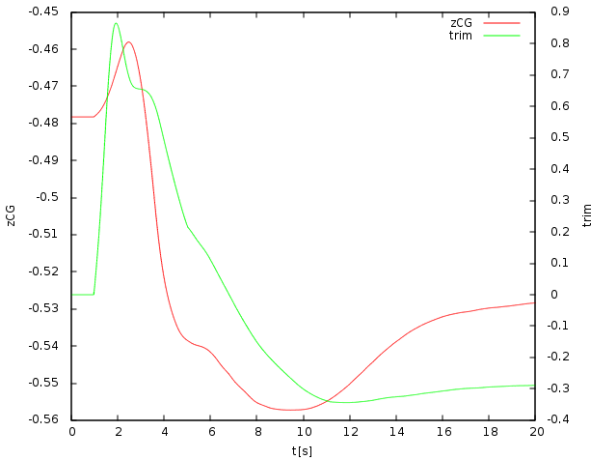


Figure 34: Evolution of trim and sink over simulated time

transport equation of the volume of fluid method has been implemented. Additionally a method for the reconstruction of the pressure discontinuity at the interface has been implemented. A robust body motion solver has been implemented and tested, a development path for further improvements of efficiency is described. First simulations for the validation have been done. The Sysser 60 case of the Delft Systematic Yacht Hull Series showed results of quality and computational time comparable to another commercial free surface code, Star-CCM+. It has been shown that the results of the new solver have better quality than the original interFoam solver. Also the stability of the solver was improved significantly compared to interFoam. On the Volvo 70 validation case good agreement to tank test results has been shown for a range of Froude numbers. The solver's capabilities to capture complex flows of different scales have been shown on the VO65OD case.

REFERENCES

BÖHM, C., GRAF, K.: *Advancements in free surface simulations for sailing yacht applications*, The Third International Conference on Innovation in High Performance Sailing Yachts, Lorient, France, (2014).

DELFT UNIVERSITY OF TECHNOLOGY: *Delft Systematic Yacht Hull Series*, http://www.dsyhs.tudelft.nl/dsyhs.php, Delft, Netherlands, (retrieved 2013).

DEVOLDER, B., SCHMITT, P., RAUWOENS, P., ELSAESSER, B., TROCH, P.: *A Review of the Implicit Motion Solver Algorithm in OpenFOAM to Simulate a Heaving Buoy*, 18th Numerical Towing Tank Symposium, Cortona-Italia, 2015.

FARR YACHT DESIGN: *Farr Volvo70 Immersed Transom Study*, Sailing Yacht Research Foundation [distributor], http://sailyachtresearch.org/19-library/44-farr-volvo70-immersed-transom-study, 2005

FÖRSTER, C., WALL, W., RAMM, E.: *Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows*, Computational Methods in Applied Mechanics and Engineering, **196**, (2007), 1278 –1293.

HADŽIĆ, I., HENNIG, J., PERIĆ, M., XING-KAEDING, Y.: *Computation of flow-induced motion of floating bodies*, Applied Mathematical Modeling, **29**, (2005), 1196 –1210.

HIRT, C.W., NICHOLS, B.D.: *Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries*, Journal of Computational Physics, **39**, (1981) 201-225.

JASAK, H., WELLER, H.G., GOSMAN, A.D.: *High resolution NVD differencing scheme for arbitrarily unstructured meshes*, International Journal for Numerical Methods and Fluids, **31**, London, (1999), 431-449.

MUZAFERIJA, S., PERIĆ, M.: *Computation of free surface flows using interface-tracking and interface-capturing methods*, Computational Mechanics Publications, Southampton, (1998), 59-100.

QUEUTEY, P., VISONNEAU, M.: *An interface capturing method for free-surface hydrodynamic flows*, Computers&Fluids, **36**, (2007), 1481-1510.

SOEDING, H.: *How to integrate free motion of solids in fluids*, 4th Numerical Towing Tank Symposium, Hamburg-Germany, 2001.

WACKERS, J., KOREN, B., RAVEN, H.C., VAN DER PLOEG, A., STARKE, A.R., DENG, G.B., QUEUTEY, P., VISONNEAU, M., HINO, T., OHASHI, K.: *Free-surface viscous flow solution methods for ship hydrodynamics*, Archives of Computational Methods in Engineering, **18**, (2010), 1-41.

B. A new adjustment-free damping method for free-surface waves in numerical simulations

VII International Conference on Computational Methods in Marine Engineering
MARINE 2017
M. Visonneau, P. Queutey and D. Le Touzé (Eds)

A NEW ADJUSTMENT-FREE DAMPING METHOD FOR FREE-SURFACE WAVES IN NUMERICAL SIMULATIONS

JANEK MEYER*, KAI GRAF† AND THOMAS SLAWIG§

*Yacht Research Unit Kiel
R&D-Centre Univ. Applied Sciences Kiel
Schwentinestrasse 24, 24149 Kiel, Germany
e-mail: Janek.Meyer@yru-kiel.de, web page: http://www.yru-kiel.de

† University of Applied Sciences Kiel
Department of Mechanical Engineering
Grenzstrasse 3, 24149 Kiel, Germany
e-mail: kai.graf@fh-kiel.de - Web page: http://www.fh-kiel.de

§ Kiel University
Algorithmic Optimal Control
Christian-Albrechts-Platz 4, 24118 Kiel, Germany
e-mail: ts@informatik.uni-kiel.de - Web page: http://www.algopt.informatik.uni-kiel.de

Key words: Damping of free-surface waves, Volume-of-Fluid method, RANSE, OpenFOAM

Abstract. Simulating free-surface flow around ships in sea waves using RANSE-Methods usually requires damping of the waves in front of the outlet to avoid reflections. It has been shown, that common damping methods like sponge layer methods deliver a reliable damping for monochromatic waves, but require a parameter adjustment by the user for different wave scales. The paper describes a new wave damping method which delivers results of same accuracy (reflections less than 2%) but does not require a manual user adjustment. The method is based on damping the vertical velocity component to reduce wave propagation. This is done by implicitly relaxing this component to zero. The relaxation is implemented with the deferred correction approach. The method is implemented in our own in-house OpenFOAM solver, which is a RANSE code using the volume of fluid method and a SIMPLE-like algorithm for the solution. Verification is done in 2D for waves of different scales, steepness, computational meshes and damping zone measures. A comparison to a linear sponge layer approach is given for the different wave scales. The 2D simulations show, that the best wave damping is achieved with the same relaxation function parameters for each individual wave. A 3D application to a modern yacht in head waves is presented. All simulation results show that the new method delivers a reliable wave damping without any parameter adjustment. The method is particularly applicable for flows with waves of different scales, like sea waves superposed with the wave system generated by a yacht.

1 INTRODUCTION

Predicting the motion of and the flow around yachts in waves using RANSE-solvers is not very common for flow analysis of sailing yachts. One of the obstacles is the generation of proper waves using free-surface flow methods. Especially the outlet of the flow domain produces unwanted reflections of the waves without using proper methods. Different methods suppressing such reflections have been developed leading from satisfying to unsatisfying results.

One group of damping methods can be classified as sponge layer methods. They are based on a damping-zone next to the boundary in which a source term is added to the governing equations. The source term usually weakens the vertical component of the fluid velocity which prevents the wave of moving through this zone. This methods deliver a good result with reflections less than 2% but require a parameter adjustment by the user.

Another way using a damping zone is presented in [1]. It explicitly relaxes the velocity to zero and the volume fraction to values of an undisturbed free-surface. This method introduces some numerical problems due to the explicit manipulation of the results of the Reynolds-averaged Navier-Stokes equations.

In [2] a method suppressing reflections at the outlet based of active filtering is presented. This method manipulates the outlet boundary condition without using a damping-zone. Indeed the damping quality is not satisfying and reflection up to 15% occure.

A lot of approaches preventing wave reflections have been given, but to the authors knowledge none of them deliver a reliable quality without parameter adjustment.

Wave damping is not only interesting for ships in waves. Also the simulation of free-surface flow around ships without sea waves may benefit of a wave-damping method. Here, the waves generated by the ship are reflected and inhibits a 100% steady-state solution. Stretching the grid in front of the outlet will prevent reflections, but a proper wave-damping method might reduce the effort for the user. Furthermore simulating offshore structures in waves require an adequate method to prevent wave reflections.

2 GOVERNING EQUATIONS AND SOLUTION METHOD FOR FREE-SURFACE FLOW

For the calculation of the free-surface flow the incompressible unsteady Reynolds-averaged Navier-Stokes equations are solved using the finite volume method. The Volume-of-Fluid (VOF) method introduced in [3] is used for the calculation of the free-surface. The momentum conservation equation (employ the eddy-viscosity hypothesis for closure), the mass conversation and the conservation equation for the transport of the volume fraction α are defined as

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \nabla \cdot \mu_e \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) = -\nabla p + \rho \mathbf{g} \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\alpha_i \mathbf{u}) = 0 \tag{3}$$

with the volume fraction α_i for the ith fluid of the two phases water and air, the velocity vector $\mathbf{u}$, the pressure p, the gravity vector $\mathbf{g}$, the density ρ and the effective dynamic viscosity μ_e. The flow properties are then calculated by $\rho = \sum_i \rho_i \alpha_i$, $\mu = \sum_i \mu_i \alpha_i$ and $1 = \sum_i \alpha_i$. The free-surface

is defined by the volume fraction $\alpha = 0.5$. The linearized, semi-discretized momentum equation can be written as

$$a_\mathrm{d}\mathbf{u}_\mathrm{d}^{q+1} + \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^{q+1} = -\nabla p^q + \mathbf{s}_{\mathrm{w/o\ p}} \quad . \tag{4}$$

Here, a represents the elements of the coefficient matrix $\mathbf{A}$ and the subscripts d and n mark the main diagonal- and neighbor-elements. All sources and contributions to the right hand side except the pressure gradient are included in $\mathbf{s}_{\mathrm{w/o\ p}}$. The solution of the current SIMPLE-iteration is marked with $q+1$ and the solution of the last iteration with q. We will not distinguish between the first prediction of the velocity and the corrected velocity of the same iteration. Rearranging (4) to $\mathbf{u}_c$ yields the velocity equation:

$$\mathbf{u}_\mathrm{d}^{q+1} = \frac{1}{a_\mathrm{d}} \left(-\nabla p^{q+1} + \mathbf{s}_{\mathrm{w/o\ p}} - \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^q \right) \quad . \tag{5}$$

Substituting and rearranging (5) into (2) yields the poisson equation for the pressure:

$$\nabla \cdot \left(\frac{1}{a_\mathrm{d}} \nabla p^{q+1} \right) = \nabla \cdot \frac{1}{a_\mathrm{d}} \left(\mathbf{s}_{\mathrm{w/o\ p}} - \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^q \right) \quad . \tag{6}$$

After integrating over the volume, the Gauss Theorem is used to transform the volume-integrals to surface-integrals. Face-variables are then interpolated with second-order discretization schemes. The pressure and its gradient is reconstructed with a method for arbitrary unstructured grids given in [4]. This prevents smearing of the density induced jump behavior at the free-surface. This includes the jump in the characteristic of the surface normal gradient of the pressure at the free-surface. Discretization in time is done with OpenFOAM's backward scheme, which is a full-implicit, second-order scheme based on quadratic interpolation using the values of two old time steps. The convective-term of the volume-of-fluid equation is discretized with the *Blended Interface Capturing Scheme with Reconstruction* (BRICS) as described in [5]. The rigid body motion is calculated with an in house motion-solver for OpenFOAM considering virtual added mass as described in [6]. This guarantees solver stability and a optimal convergence behavior even for relatively light ships like sailing-yachts. The motion-solver has been implemented and shared with us by FluidEngineeringSolutions GmbH & Co. KG, Germany. The equations are solved in a segregated algorithm and following steps are done in each time step; (A) if required solve motion eqs., (B) solve VOF-equation, (C) solve momentum predictor, (D) solve pressure eq., (E) update flux and velocity, (F) solve turbulence equations. Inside step (D) the pressure eq. may be solved a number of times for non-orthogonal correction. Steps (D) to (E) may be repeated to apply a Piso-like Correction. Steps (A) to (F) are repeated and yield the typical outer iterations of the SIMPLE-algorithm. The momentum and VOF equations are relaxed implicitly. The new solution of the pressure is relaxed explicitly before correcting the velocity. A detailed description of the solver and solution method is given in [7].

3 COMMON WAVE DAMPING METHODS

Two widely used reliable wave damping methods have been described in [8] and [9], whereby the first will be described here. The method is based on a sponge layer which can be derived

by the typical equations for porous media. The damping is achieved by adding a source term to the momentum equation inside a user-defined damping-zone in front of the outlet. The source term is added to the vertical z-component of the momentum equation and can be written as

$$s_{\mathrm{z}}^{\mathrm{d}} = -\rho \left(f_1 + f_2 |u_{\mathrm{z}}| \right) w u_{\mathrm{z}} \tag{7}$$

with the weight-function

$$w = \frac{e^{\kappa} - 1}{e^{1} - 1} \tag{8}$$

and the character of the blending function

$$\kappa = \left(\frac{x - x_{\mathrm{sd}}}{x_{\mathrm{ed}} - x_{\mathrm{sd}}} \right)^{\zeta} \tag{9}$$

with ζ usually set to 3.5. Here ρ is the density of the fluid and u_z is the the vertical velocity component. The parameter f_1 gives the amount of linear damping, f_2 the amount of quadratic damping. The weight factor w depends on the location inside the domain and helps to smoothly fade in the source term in the damping zone. The wave propagation direction is given by x with x_{sd} as the start and x_{ed} as the end x-coordinate of the damping zone.
This method is implemented in commercial codes like STAR-CCM+ or in a slightly different form in ANSYS Fluent. It has been deeply investigated in [10] and it has been shown, that the method delivers a reliable damping with satisfying damping quality. Nevertheless the parameters f_1 and f_2 have to be adjusted for different waves. Scaling laws for adjusting these parameters are also given in [10]. Assuming optimal chosen parameters for a regular and monochromatic wave adjustment is necessary if the wave changes in its scale. Where no adjustment is required if the computational mesh or wave steepness is changed. Also no adjustment is required for different lengths of the damping zone. However, the maximal achievable damping quality depends on this length and at least two wavelengths are recommended.

4 DERIVATION OF THE NEW WAVE DAMPING METHOD

In [1] wave damping is achieved by relaxing the velocity $\mathbf{u}$ and the volume fraction α explicitly after solving for the volume fraction. Explicit relaxation is done with the generic equation

$$\phi_{\mathrm{relaxed}} = r\phi + (1 - r)\phi^{\mathrm{t}} \ . \tag{10}$$

Here ϕ is an generic quantity, the superscript t signifies the target value. The relaxation factor r depends on above mentioned damping weight w:

$$r = 1 - w \ . \tag{11}$$

The method delivers a good damping quality but has significant disadvantages. All three velocity components are relaxed to zero. The volume fraction is relaxed to values which assume an undisturbed flat free-surface at constant height. This forbids additional current or boat speed superposing with the orbital velocity of the waves. Additionally this delivers some kind of Dirichlet boundary condition (BC) at the outlets, whereby a Neumann BC is desirable. Furthermore

such an explicit relaxation will prevent the convergence of the SIMPLE-algorithm. Still, this relaxation approach and the sponge layer approach described in section 3 are the inspiration for our method. Our goal is to reduce the vertical velocity component by the use of an implicit relaxation included in the momentum equation. Considering the discretized momentum equation $\mathbf{A} \cdot \mathbf{u} = \mathbf{s}$ with the coefficient matrix $\mathbf{A}$, the velocity vector $\mathbf{u}$ and all explicitly treated terms on the right hand side included in the source $\mathbf{s}$, implicit relaxation of all velocity components can be done with

$$\frac{1}{r} a_\mathrm{d} \mathbf{u}_\mathrm{d}^{q+1} + \sum_n a_\mathrm{n} \mathbf{u}_\mathrm{n}^{q+1} = \mathbf{s} + \frac{1-r}{r} a_\mathrm{d} \mathbf{u}_\mathrm{d}^\mathrm{t} \quad . \tag{12}$$

Here, the subscripts d marks the main diagonal- and n the neighbor-elements of $\mathbf{A}$. As the limiting case, where the relaxation factor r tends to zero the convergence of the equation system is not obvious. Adopting L'Hôpital's rule twice one can show that $\mathbf{u}_\mathrm{d}^{\mathrm{q}+1}$ tends to $\mathbf{u}_\mathrm{d}^\mathrm{t}$ as aspired. Modifying equation (12) to relax only the vertical velocity component is not straightforward. Using this approach the coefficient matrix $\mathbf{A}$ has to be modified. Indeed it is common practice to reuse this matrix for all three components. Therefore manipulating $\mathbf{A}$ only for the vertical component will produce additional calculation effort and a lot of programing effort to implement this methods into existing numerical codes like OpenFOAM. To solve this problem our idea is to implement the implicit relaxation with the help of the deferred correction approach. That means, the product of the modified matrix and the velocity is treated explicitly on the right hand side. Additionally the product of the unmodified matrix and the velocity is added on both, explicit and implicit, sides leading to the original unmodified left hand side. If the equation system is converging, the terms with the unmodified matrix are canceling each other out and the solution depends only on the modified matrix. This allows to modify only the right hand side, more precisely only the z-component of the right hand side. In the following we will describe two approaches to use the deferred correction. The first approach might be the obvious way to go but leads to a diverging equation system, as it will be shown. The second approach leads to our final damping method and a converging equation system.

4.1 First Approach (divergent)

In the following equations all terms on the left hand side consider the unknown velocity from the current iteration $q+1$ and all terms on the right hand side use the known values of the last iteration q. For convergence the velocity u^q should tend to $u^{\mathrm{q}+1}$. Starting from the z-component of the vector equation (12)

$$\frac{1}{r} a_\mathrm{d} u_\mathrm{zd}^{q+1} + \sum_n a_\mathrm{n} u_\mathrm{zn}^{q+1} = s_\mathrm{z} + \frac{1-r}{r} a_\mathrm{d} u_\mathrm{zd}^\mathrm{t} \tag{13}$$

and applying $a_\mathrm{d} u_\mathrm{zd}$ on both sides leads to

$$\frac{1}{r} a_\mathrm{d} u_\mathrm{zd}^{q+1} + a_\mathrm{d} u_\mathrm{zd}^{q+1} + \sum_n a_\mathrm{n} u_\mathrm{zn}^{q+1} = s_\mathrm{z} + \frac{1-r}{r} a_\mathrm{d} u_\mathrm{zd}^\mathrm{t} + a_\mathrm{d} u_\mathrm{zd}^q \quad . \tag{14}$$

Putting the term $\frac{1}{r} a_\mathrm{d} u_\mathrm{zd}^{q+1}$ from the left to the right hand side changes its iteration index and leads to

$$a_\mathrm{d} u_\mathrm{zd}^{q+1} + \sum_n a_\mathrm{n} u_\mathrm{zn}^{q+1} = s_\mathrm{z} + \frac{1-r}{r} a_\mathrm{d} u_\mathrm{zd}^\mathrm{t} + a_\mathrm{d} u_\mathrm{zd}^q - \frac{1}{r} a_\mathrm{d} u_\mathrm{zd}^q \quad . \tag{15}$$

Simplifying this equation delivers the final equation for the first approach

$$a_{\mathrm{d}} u_{\mathrm{zd}}^{q+1} + \sum_n a_{\mathrm{n}} u_{\mathrm{zn}}^{q+1} = s_{\mathrm{z}} + \frac{1-r}{r} \left(a_{\mathrm{d}} u_{\mathrm{zd}}^{\mathrm{t}} - a_{\mathrm{d}} u_{\mathrm{zd}}^{q} \right) \quad . \tag{16}$$

The whole relaxation of the z-component for the velocity is included in one source term on the right hand side. Therefore it is no more necessary to manipulate the coefficient matrix $\mathbf{A}$ and it is possible to include the relaxation of the z-component in the typical vector form of the momentum equation used in codes like OpenFOAM.

4.2 Investigation on the convergence behavior - 1st approach

Using the Jacobi method to solve equation (16) leads to

$$u_{\mathrm{zd}}^{q+1} = \frac{1}{a_{\mathrm{d}}} \left(s_{\mathrm{z}} + \frac{1-r}{r} \left(a_{\mathrm{d}} u_{\mathrm{zd}}^{\mathrm{t}} - a_{\mathrm{d}} u_{\mathrm{zd}}^{q} \right) - \sum_n a_{\mathrm{n}} u_{\mathrm{zn}}^{q} \right) \quad . \tag{17}$$

For wave damping ($u_{\mathrm{zd}}^{\mathrm{t}} = 0$) with full relaxation ($r \to 0$) the source term s_{z} becomes negletable and the solution of the equation is $u_{\mathrm{zd}}^{q+1} \to -\infty u_{\mathrm{zd}}^{q}$. Therefore no convergence is possible for small r.

4.3 Second Approach

For the second approach equation (13) is multiplied with the relaxation factor r before applying the deferred correction

$$a_{\mathrm{d}} u_{\mathrm{zd}}^{q+1} + r \sum_n a_{\mathrm{n}} u_{\mathrm{zn}}^{q+1} = r s_{\mathrm{z}} + (1-r)\, a_{\mathrm{d}} u_{\mathrm{zd}}^{\mathrm{t}} \quad . \tag{18}$$

In eq. (13) of the first approach the main diagonal elements of $\mathbf{A}$ are multiplied with $\frac{1}{r}$. Now, in eq. (18) the neighbor elements of $\mathbf{A}$ and the right hand side term s_{z} are multiplied with r. Applying the deferred correction method to eq. (18) to get rid of the relaxation factor on the left hand side leads to

$$a_{\mathrm{d}} u_{\mathrm{zd}}^{q+1} + \sum_n a_{\mathrm{n}} u_{\mathrm{zn}}^{q+1} = r s_{\mathrm{z}} + (1-r) \left(a_{\mathrm{d}} u_{\mathrm{zd}}^{\mathrm{t}} + \sum_n a_{\mathrm{n}} u_{\mathrm{zn}}^{q} \right) \quad . \tag{19}$$

Here, the original right hand side term s_{z} is manipulated. This needs to be considered in the derivation of the pressure equation. Our solution how to consider this as easy as possible is described in subsection 4.5.

4.4 Investigation on the convergence behavior - 2nd approach

Using the Jacobi method again to solve eq. (19) gives

$$u_{\mathrm{zd}}^{q+1} = \frac{1}{a_{\mathrm{d}}} \left(r s_{\mathrm{z}} + (1-r) \left(a_{\mathrm{d}} u_{\mathrm{zd}}^{\mathrm{t}} + \sum_n a_{\mathrm{n}} u_{\mathrm{zn}}^{q} \right) - \sum_n a_{\mathrm{n}} u_{\mathrm{zn}}^{q} \right) \quad . \tag{20}$$

For full relaxation ($r \to 0$) one gets the aspired behavior $u_{\mathrm{zd}}^{q} \to u_{\mathrm{zd}}^{\mathrm{t}}$ where $u_{\mathrm{zd}}^{\mathrm{t}}$ is zero for wave damping.

4.5 Manipulation of the original right hand side

Using equation (19) requires to manipulate the original right hand side s_z including the terms depending on the pressure p. Therefore it is necessary to build a new pressure equation. This can be done straightforward but is not our finally chosen way. For the sake of completeness this way will be given here, first. Afterward a simpler way leading to our final method will be described.

First way:
The vector form of the momentum equation considering the wave damping method can be written as:

$$a_\mathrm{d}\mathbf{u}_\mathrm{d}^{q+1} + \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^{q+1} = \mathbf{R}\cdot\left(-\nabla p^q + \mathbf{s}_{\mathrm{w/o\ p}}\right) + \mathbf{s}^* \quad . \tag{21}$$

Here, $\mathbf{s}_{\mathrm{w/o\ p}}$ is the original right hand side without the pressure gradient ($\mathbf{s} = -\nabla p + \mathbf{s}_{\mathrm{w/o\ p}}$). The tensor $\mathbf{R}$ allows to manipulate only the z-component of the equation and is defined as $\mathbf{R} = \begin{bmatrix} r_\mathrm{x} & 0 & 0 \\ 0 & r_\mathrm{y} & 0 \\ 0 & 0 & r_\mathrm{z} \end{bmatrix}$ with the relaxation factors $r_\mathrm{x} = r_\mathrm{y} = 1.0$ and $r_\mathrm{z} = r$ for the three Cartesian directions. The source term $\mathbf{s}^*$ is defined as

$$\mathbf{s}^* = (\delta_{ij} - \mathbf{R})\cdot\left(a_\mathrm{d}\mathbf{u}_\mathrm{d}^\mathrm{t} + \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^q\right) \tag{22}$$

with the Kronecker-delta $\delta_{ij} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Rearranging eq. (21) for $\mathbf{u}_\mathrm{d}^{q+1}$ leads to the new velocity equation allowing to calculate the corrected velocity $\mathbf{u}_\mathrm{d}^{q+1}$

$$\mathbf{u}_\mathrm{d}^{q+1} = \frac{1}{a_\mathrm{d}}\left(\mathbf{R}\cdot\left(-\nabla p^{q+1} + \mathbf{s}_{\mathrm{w/o\ p}}\right) + \mathbf{s}^* - \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^q\right) \quad . \tag{23}$$

Substituting the velocity of the continuity equation (2) with (23) and rearranging gives the adopted pressure equation

$$\nabla\cdot\left(\frac{1}{a_\mathrm{d}}\mathbf{R}\cdot\nabla p^{q+1}\right) = \nabla\cdot\frac{1}{a_\mathrm{d}}\left(\mathbf{R}\cdot\mathbf{s}_{\mathrm{w/o\ p}} + \mathbf{s}^* - \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^q\right) \quad . \tag{24}$$

Second way:
For the second way the momentum equation is written in a way containing the unmodified right hand side $\mathbf{s}$ and an additional source term

$$a_\mathrm{d}\mathbf{u}_\mathrm{d}^{q+1} + \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^{q+1} = \underbrace{-\nabla p^q + \mathbf{s}_{\mathrm{w/o\ p}}}_{\mathbf{s}} + \mathbf{s}_\mathrm{wavedamping} \tag{25}$$

with

$$\mathbf{s}_\mathrm{wavedamping} = (\delta_{ij} - \mathbf{R})\cdot\left(\nabla\tilde{p}^q - \mathbf{s}_{\mathrm{w/o\ p}} + a_\mathrm{d}\mathbf{u}_\mathrm{d}^\mathrm{t} + \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}^q\right) \quad . \tag{26}$$

It is important to notice, that the pressure gradient is used in two terms. To build the pressure equation it is only solved for the pressure of the original right hand side $\mathbf{s}$. The pressure gradient inside the source term for the wave damping $\mathbf{s}_{\text{wavedamping}}$ is fixed and therefore marked with a tilde. The equation for the corrected velocity becomes

$$\mathbf{u}_{\text{d}}^{q+1} = \frac{1}{a_{\text{d}}} \left(-\nabla p^{q+1} + \mathbf{s}_{\text{w/o p}} + \mathbf{s}_{\text{wavedamping}} - \sum_n a_{\text{n}} \mathbf{u}_{\text{n}}^{q} \right) \tag{27}$$

and the pressure equation becomes

$$\nabla \cdot \left(\frac{1}{a_{\text{d}}} \nabla p^{q+1} \right) = \nabla \cdot \frac{1}{a_{\text{d}}} \left(\mathbf{s}_{\text{w/o p}} + \mathbf{s}_{\text{wavedamping}} - \sum_n a_{\text{n}} \mathbf{u}_{\text{n}}^{q} \right) \quad . \tag{28}$$

Because $\mathbf{s}_{\text{wavedamping}}$ contains the pressure of the last iteration p^q the solution of the pressure could be interpreted as deferred inside the wave damping zone.

This is our final wave damping method made up of only one additional source term. The original right hand side $\mathbf{s}$ can be pre-calculated and then reused to build the wave-damping source term $\mathbf{s}_{\text{wavedamping}}$. The weight function w included in the relaxation factor r has not the final form and will be substituted with the optimized function w_{opt} as described in chapter 6.3.

5 WAVE GENERATION

At the inlet the values for the velocities and volume fractions are prescribed according to the wave theory. An equation for the pressure value itself or the pressure gradient is missing, which is a significant problem. Our solution is using a Neumann boundary condition with a zero gradient for the pressure. Certainly, the assumption of a zero gradient is wrong in the presence of a wave and for many 3D simulations the wave will collapse behind the inlet. Therefore we are using an relaxation zone behind the inlet and apply full implicit relaxation for the momentum and volume-of-fluid equation. The relaxation factor has the reversed characteristic of the relaxation factor of the damping-zone. Therefore full relaxation is applied at the inlet going to no relaxation at the end of the generation-zone. The target values $\mathbf{u}^{\text{t}}$ and α^{t} are calculate with the chosen wave theory for example 5th-order Stokes. All three velocity components are relaxed to the target value by using equation (12). The relaxation for the volume fraction is applied in the same way

$$\frac{1}{r} a_{\text{VOF d}} \alpha_{\text{d}}^{q+1} + \sum_n a_{\text{VOF n}} \alpha_{\text{n}}^{q+1} = s_{\text{VOF}} + \frac{1-r}{r} a_{\text{VOF d}} \alpha_{\text{d}}^{\text{t}} \tag{29}$$

with the diagonal elements $a_{\text{VOF d}}$ and neighbor elements $a_{\text{VOF n}}$ of the coefficient matrix of the volume-of-fluid transport equation. The term s_{VOF} contains all possible source terms or contributions to the right hand side.

We have not systematically tested different zone length, but a length of 1.0λ produces a reliable wave generation for all of our simulations.

The free-surface is initialized wave-less in the hole domain. For the wave generation we are using the T-soft-parameter of the waves2Foam library. This generates a wave growing from a smooth surface to the full height over the given time T_{soft} in the generation-zone. The growing time T_{soft} is set to the wave-period T. Due to the relaxation approach, the relaxation factor

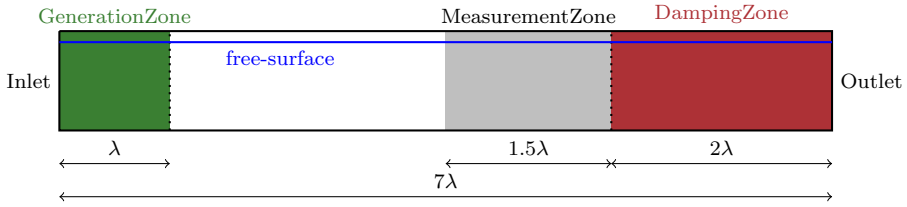


Figure 1: 2d Simulation Setup - clinched in z-direction

becomes small at the end of the generation zone. Therefore, while growing the wave gets smeared at the end of the zone, because the values are set to the new height only partially which leads to very small oscillations in the wave. Still this is acceptable and our best method to produce waves. The oscillations are quantified in chapter 6.

6 VERIFICATION AND WEIGHT FUNCTION OPTIMIZATION

To estimate the damping quality of our method a test-case very close to the one presented in [10] has been chosen. The test-case is also used to optimize the function for the relaxation factor for monochromatic waves. Additionally we have implement the typical sponge layer method for comparison.

6.1 Numerical Simulation Setup

The test-case is based on a 2D wave tank. Figure 1 shows the domain and its measurements. The wave is generated in the generation zone with a length of 1λ. Afterward the wave is propagated through a region of length 4.0λ. Finally the wave damping is applied in the damping zone of length 2.0λ. The Reflections are measured inside the measurement zone in front of the damping zone as explained later. The domain has a height of 4.5λ filled with water up to 4.0λ. The grid has a resolution of 20 cells per wave-height and 100 cells per wave-length as recommended in [10] which refers to [11]. The grid is coarsened with growing distance to the wave. The waves are generated by prescribing volume fractions and velocities of 5th-order Stokes waves at the inlet boundary and in the generation zone as explained in chapter 5. A Neumann boundary condition (BC) with zero gradient is applied for the pressure at the inlet. A free slip wall is applied at the bottom. Indeed, instead of using the hydrostatic pressure gradient a Dirichlet BC with given hydrostatic pressure is applied for the pressure at the bottom. This is done to prevent problems with the preservation of the position of the free-surface. Same BC is applied at the top with the exception, that the velocity BC is set to a zero gradient BC to get a behavior as an open water tank. At the outlet a zero gradient BC is applied for all variables. This allows a lowered free-surface at the outlet or a wave traveling through the outlet.

The solver is using a time step of about $\frac{1}{500}$ of the wave period and 10 SIMPLE-Loops. The pressure is relaxed by 0.3, the velocity by 0.7 and the volume fraction by 0.9. The simulation is assumed to be laminar and no turbulence-model is applied.

Simulations have been done for different wave scales; a medium sized wave with $\lambda = 4.0$m and $H = 0.16$m, a small wave with $\lambda = 0.04$m and $H = 0.0016$m and a big wave with $\lambda = 400$m and $H = 16.0$m. To compare the dependency of the wave steepness a steep wave with $\lambda = 0.4$ and $H = 0.16$ has been simulated. The grid has been scaled accordingly.

The influence of the grid is investigated by using a coarse grid with 10 cells per wave height and 50 cells per wave length for the medium sized wave. The influence of the damping zone length is investigated by also using a damping zone with a length of 1λ for the medium sized wave. The grid is the same as for the initial, medium sized wave simulation. The measurement zone is also kept at the initial position.

6.2 Quantifying the damping quality

To quantify the damping quality, the maximal and minimal wave heights $H_{\max}$ and $H_{\min}$ are measured in the measurement zone in the time interval $[22.0T, 24.0T]$ at 40 evenly distributed times. These wave heights are used to build a reflection coefficient as suggested in [12]

$$C_{\mathrm{R}} = \frac{H_{\max} - H_{\min}}{H_{\max} + H_{\min}} \ . \tag{30}$$

For perfect wave reflection of 100% the coefficient C_{R} becomes 1.0 and zero for no reflection. Considering that the wave train moves with the half speed of a single wave, the chosen time interval allows, that the wave train propagates from the inlet to the outlet and after reflection back to the measurement zone. The wave height is estimated by measuring the minimal and maximal z-Position of the free surface inside the measurement zone. Please notice, that the reflection coefficient includes the wave reflections, but also wave oscillations originating from the wave generation. Additionally a wave, flattening inside the measurement zone due to an insufficient discretization of time or of the convective term will influence the reflection coefficient. The superposition of all these phenomenons may influence the coefficient negatively, but also positively. Reflections may occur at the outlet but also inside or at the beginning of the damping zone, due to too much dampening. To estimate the background oscillations (BO) originating from the wave generation and the influence of flattening all simulations have been done without wave damping but with a long domain of length 25.0λ.

6.3 Optimizing the weight function

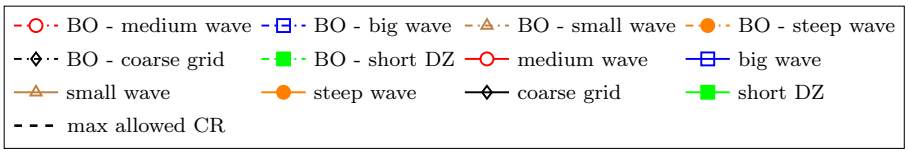
The weight function given in equation (8) does not necessarily guarantee the least reflections. Looking at equation (7) of the sponge layer method one can interpret the parameters f_1 and f_2 as a scaling of the weight function w for the different waves. To allow an optimization for our method we are also scaling the weight function with the new introduced scale factor χ

$$w_{\mathrm{opt}} = \chi w \ . \tag{31}$$

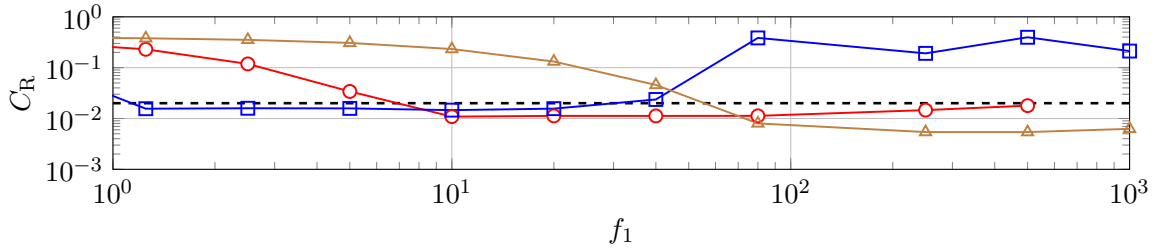
The scale factor χ is varied from 0.008 to 1.0 for each wave to achieve the least reflections and to show the dependency of our wave damping method from the wave characteristics.

6.4 Results

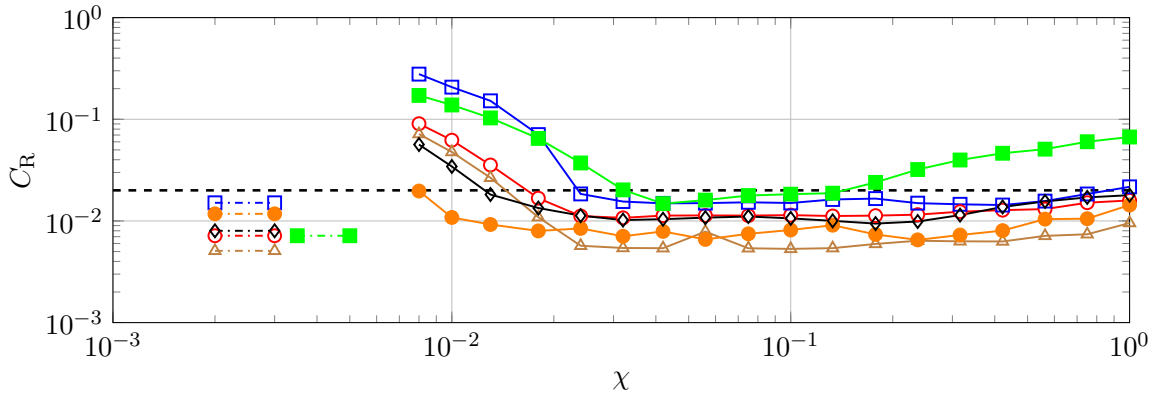
Figure 2 shows the results of all simulations. Sub-figure b shows the results of the common linear sponge layer method (with $f_2 = 0.0 = const$), also implemented in our solver. The results are remarkable close to the results presented in [10] using the same method but implemented in StarCCM+. The black, dashed line shows the limit of $C_{\mathrm{R\,lim}} = 0.02$ for an acceptable



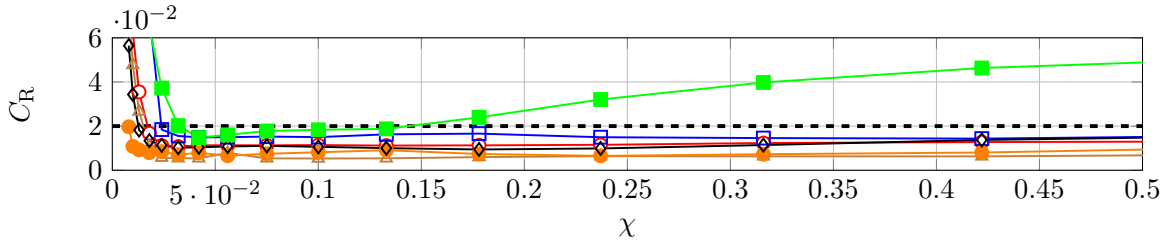
(a) Legend



(b) Sponge Layer Method for different wave scales



(c) New method log scale



(d) New method linear scale

Figure 2: Reflection coefficient for different wave damping method and different waves

small wave: $\lambda = 0.04$m, $H = 0.0016$m; medium wave: $\lambda = 4.0$m, $H = 0.16$m; big wave: $\lambda = 400$m, $H = 16.0$m; steep wave: $\lambda = 0.4$m, $H = 0.16$m

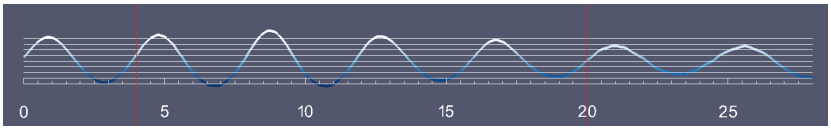


Figure 3: 2D-Wave at $t = 40s$, no damping

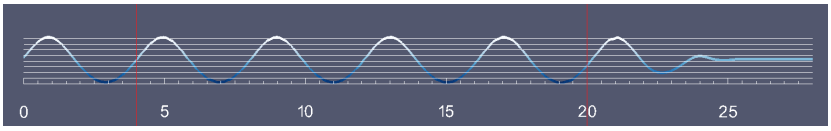


Figure 4: 2D-Wave at $t = 40s$, damping with $\chi = 1.0$ (too much damping)

reflection. Here, the sponge layer method has only been used for different wave scales and one can clearly see that f_1 has to be adjusted to achieve an acceptable reflection. For the quadratic sponge layer method we would like to refer to the results of [10], showing that f_2 has to be adjusted, too. Sub-figure c shows the results of our new method with both axis scaled logarithmically. The initial background oscillations (BO) are marked with a dashed line in the left area of the diagram. All background oscillations are below the maximal allowed value. This shows that the oscillations produced by the wave generation method mentioned in chapter 5 are acceptable. All curves show the same characteristics and an optimal wave damping is achieved for approximately $0.03 \leq \chi \leq 0.15$. The absolute minimum of each curve corresponds with the background oscillations. Only the simulation with the short damping zone clearly shows a worse damping quality. Additionally the absolute minimum for the steep wave deceeds the corresponding background oscillations. A clear reason can not be given but it is possible that the background oscillations and the reflections are canceling each other out.

The characteristics of the curves can be interpreted in that way, that the increasing reflections in the area $\chi < 0.03$ are reflections which arises predominantly at the outlet due to too little damping and that the increasing reflections in the area $\chi > 0.15$ are reflections which arises predominantly at the beginning or inside the wave damping zone due to too fast wave damping.

Sub-figure d shows the same results as c but with a linear scale. This should emphasize graphically that all curves have the same characteristics. Therefore our wave damping method allows to use the same weight function for all different monochromatic waves and no user adjustment is required. To make short damping zones practical we recommend a scale factor χ of about 0.09. Nevertheless we recommend a damping zone with a length of $2.0\lambda_{\max}$ where $\lambda_{\max}$ should be the largest wave length of a given wave spectrum. One can assume that the evaluation of the damping quality would turn out much better without the presence of the background oscillations.

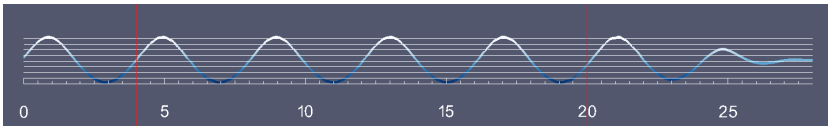


Figure 5: 2D-Wave at $t = 40s$, damping with $\chi = 0.1$ (optimal damping)

Figures 3 to 5 show the wave shapes for different damping parameters. The end of the generation-zone (left) and the beginning of the damping-zone (right) are marked with vertical red lines. Figure 3 shows the wave after $40s$ without any wave-damping applied, leading to reflections of 42%. One can clearly see the reflections superposing with the original wave. Figure 4 shows the result arising due to too much damping, which still leads to only 1.6% reflection. Here, the free-surface becomes completely flat in front of the outlet. Figure 5 shows the wave shape arising due to optimal wave-damping leading to only 1.1% reflection. Here, still a very small wave reaches the outlet. Compared to 4, this will even more reduce reflections at the beginning of the damping-zone. Furthermore, possible reflections from the outlet will be damped while traveling back through the damping-zone.

7 APPLICATION TO A YACHT IN HEAD WAVES

This section shows an application of the new wave damping method to a yacht in head waves. This is a good example for the superposition of the sea waves with the small waves of the yacht and furthermore with the velocity of the yacht.

The yacht is a Class 40 (C40) designed by VPLP, France, and has a length over surface of LoS = 12m. The waves have a length of $\lambda = 18$m, a height of $H = 0.4$m and a period of $T = 3.489$s. Second order Stokes theory is used for wave generation. The yacht is accelerated to $u = 4.4$m/s in the first second. This leads to an encounter frequency of $0.53\frac{1}{\text{s}}$. The yacht is free in pitch and heave.

The solution-domain uses a wave-generation zone of 1λ and a wave-damping zone of 2λ. The free flow begins 1LoS in front of the yacht and ends 2LoS behind the yacht. The grid is generated using the OpenFOAM mesher snappyHexMesh, but with the use of some small in-house code and a lot of scripting to get a mesh of good quality with anisotropic refinement. The grid uses 100 cells per wave length and 20 cells per wave height. Depending on the part of the yacht, seven to nine prism-layers are applied leading to $y+$ values of 40 to 90 underwater. The kelvin-refinement ends in front of the wave-damping zone. The cells inside the kelvin-refinement have a size less or equal to 0.125m. The final mesh has 8.5E6 cells.

A modified $k-\omega$-SST turbulence model is used. The modification consist of a different production term as described in [1] and a correct consideration of the density-derivations for free-surface flows, compared to OpenFOAM's standard turbulence models. The timestep was

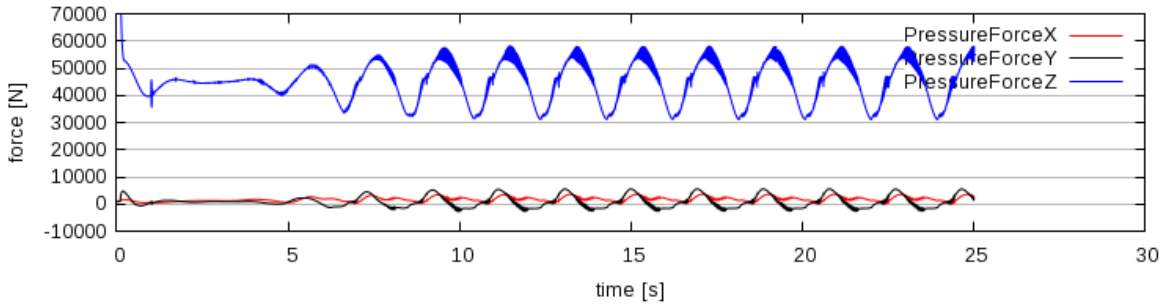


Figure 6: C40-design in head waves - pressure forces

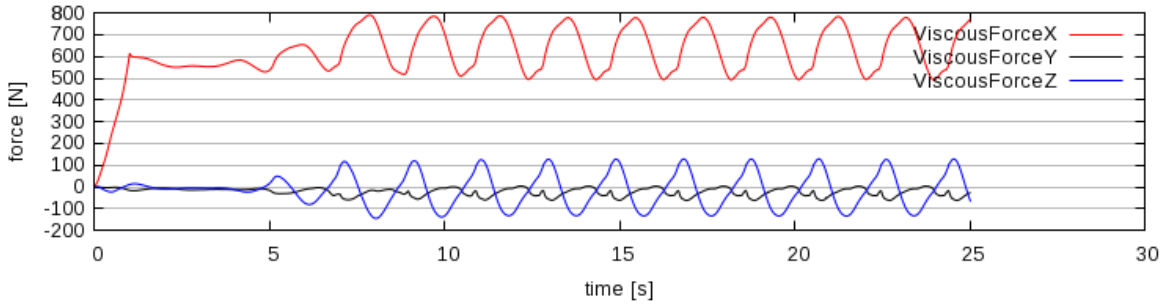


Figure 7: C40-design in head waves - viscous forces

set to 0.005s and the solver uses 5 SIMPLE-Loops. The simulation was calculated on a cluster with three knots, each with two 6-core CPUs. The simulation took about 65 hours.

Figures 6 and 7 show the pressure and the viscous forces. One can clearly see a periodic behavior and no significant disturbance due to wave reflections. The viscous force in x-direction has the most variation. This can be explained with ventilation under the hull composed of correct ventilation due to encountering a trough and incorrect numerical ventilation due to smearing of the free-surface. First simulations using a coarser grid (4.4E6 cells, three prism layers, coarser kelvin refinement) show much more variations in the viscous forces in the x-direction.

Figures 8 and 9 show a close view onto the hull and its wave system for two different times. Figure 8 shows the hull diving into the sea wave, whereas Figure 9 shows the hull with the bow knuckle significantly above the free-surface. In both cases, one can clearly see a smooth and good resolved kelvin wave pattern and furthermore breaking waves at the hull.

Figures 10 and 11 show wave patterns at different times. The end of the generation-zone and the beginning of the damping-zone are marked with two red y-planes in Figure 11. As one can see, the waves get damped in front of the outlet. The large scaled sea waves are not dampened completely at the outlet as expected for the chosen χ of 0.09. Apart from that, the small scaled wave-system of the yacht seems to be dampened completely at the outlet. The reason for this is first the length of the damping-zone, which is significantly longer than twice the wave length of the yacht induced waves. Therefore the damping for this small-scaled waves should be of a quality better than required. The second reason is the kelvin refinement ending in front of the damping-zone. At the y-max and y-min domain borders (the sides of the domain) the free traveling waves show no significant differences to each others. This underscores first the good wave propagation due to the 2nd order time discretization and second the suppression of wave reflections.

8 CONCLUSION

A new wave damping method was derived and investigated for monochromatic waves. The simulations show an overall good damping quality with reflections less than 2%. For different wave characteristics the same parameters leads to the best damping quality. Therefore the method can be seen as adjustment-free in the scope of the investigated waves. The damping quality for irregular or breaking waves has not been investigated and is an open topic for future investigations. Nevertheless, especially for irregular waves, which can be seen as a superposition

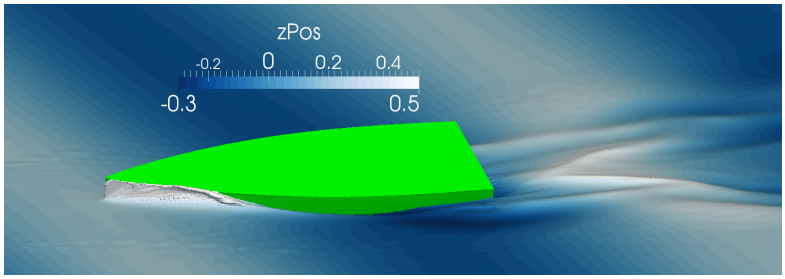


Figure 8: C40-design close-up at $t = 23.1$

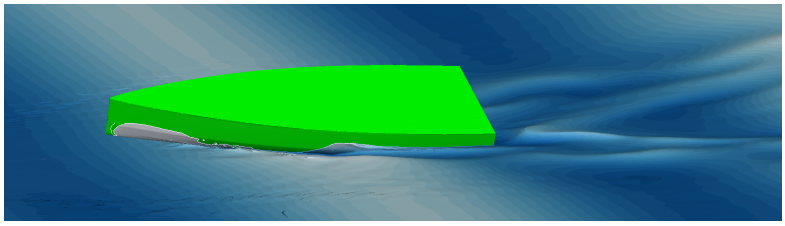


Figure 9: C40-design close-up at $t = 23.9$

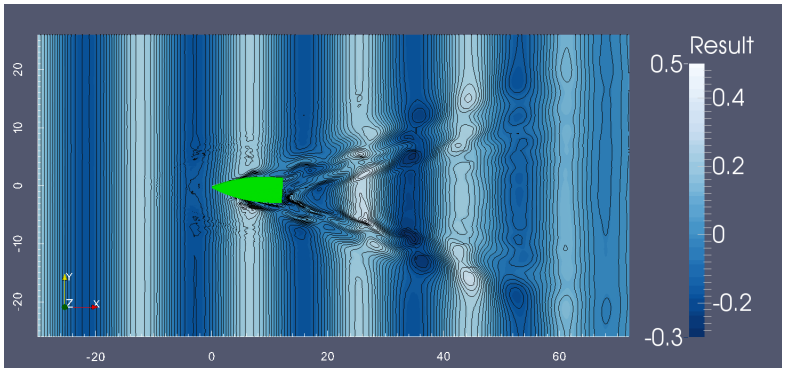


Figure 10: C40-design wave pattern at $t = 23.9$

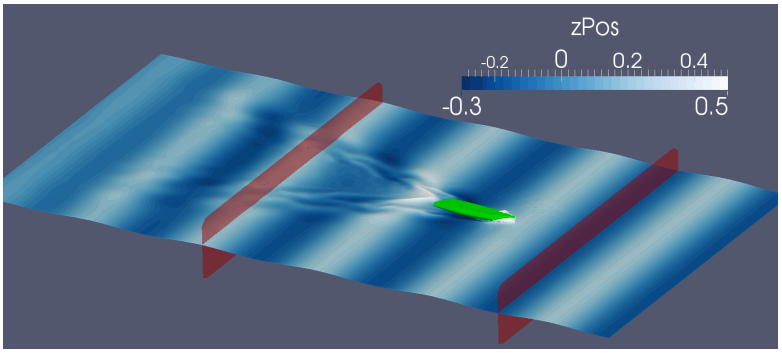


Figure 11: C40-design 3D view at $t = 23.1$

of regular waves, the authors expect a good damping quality, due to the adjustment-free behavior for monochromatic regular waves. The application of the method to a yacht in head waves emphasizes the user-friendly applicability, as it delivers a periodic solution with less variation just by activating the damping method with our optimized parameters.

REFERENCES

[1] Jacobsen, N.G., Fuhrman, D.R. and Fresdøe, J., A wave generation toolbox for the open-source CFD library: OpenFOAM®, *International Journal for Numerical Methods in Fluids*, (2012), **70**:1073–1088.

[2] Higuera, P., Lara, J.L. and Losada, I.J., Realistic wave generation and active wave absorption for Navier-Stokes models: Application to OpenFOAM®, *Coastal Engineering*, (2013), **71**:119–134.

[3] Hirt, C.W. and Nichols, B.D., Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries, *Journal of Computational Physics*, (1981) **39**:201–225.

[4] Queutey, P. and Visonneau, M., An interface capturing method for free-surface hydrodynamic flows, *Computers & Fluids*, (2007), **36**:1481–1510.

[5] Wackers, J., Koren, B., Raven, H.C., van der Ploeg, A., Starke, A.R., Deng, G.B., Queutey, P., Visonneau, M., Hino, T. and Ohashi, K., Free-surface viscous flow solution methods for ship hydrodynamics, *Archives of Computational Methods in Engineering*, (2010), **18**:1–41.

[6] Soeding, H., How to integrate free motion of solids in fluids, *4th Numerical Towing Tank Symposium*, Hamburg-Germany, (2001).

[7] Meyer, J., Renzsch, H., Graf, K. and Slawig, T., Advanced CFD-Simulations of free-surface flows around modern sailing yachts using a newly developed OpenFOAM solver, *The 22nd Chesapeake Sailing Yacht Symposium*, Annapolis-Maryland, (2016).

[8] Choi, J and Yoon, S.B., Numerical simulations using momentum source wave-maker applied to RANS equation model, *Coastal Engineering*, (2009), **56**:1043–1060.

[9] Park, J.C., Kim, M.H. and Miyata, H., Fully non-linear free-surface simulations by a 3D viscous numerical wave tank, *International Journal for Numerical Methods in Fluids*, (1999), **29**:685–703.

[10] Perić, R. and Abdel-Maksoud, M., Reliable Damping of Free Surface Waves in Numerical Simulations, *Ship Technology Research*, (2016), **63**.

[11] Perić, R., Internal generation of free surface waves and application to bodies in cross sea, MSc Thesis, *Schriftenreihe Schiffbau*, Hamburg University of Technology, Hamburg-Germany, (2013).

[12] Ursell, F., Dean, R.G. and Yu, Y.S., Forced small-amplitude water waves: a comparison of theory and experiment, *Journal of Fluid Mechanics*, (1960), **7**:33–52.

C. Simulation of scour around arbitrary offshore foundations based on the Volume-of-Fluid method combined with a Bingham model

Simulation of scour around arbitrary offshore foundations based on the Volume-of-Fluid method combined with a Bingham model

Janek Meyer *[1], Kai Graf[2] and Thomas Slawig[3]

[1] *Yacht Research Unit Kiel, R&D-Centre University of Applied Sciences Kiel, Germany*
[2] *Institute of Naval Architecture, University of Applied Sciences Kiel, Germany*
[3] *Departement of Computer Science, Kiel University, Germany*

5th December 2020

Abstract

This paper presents a method for the simulation of scour around arbitrary offshore structures. It is based on the solution of the Reynolds-Averaged-Navier-Stokes equations implemented in the OpenFOAM framework. The sediment is simulated with the help of a Bingham model, which basically models a solid sediment behavior by introducing a very high viscosity. The relative pressure used by the Bingham model is estimated with a new approach based on the solution of a Poisson equation. The position of the sediment surface is calculated with the Volume-of-Fluid approach using a high-resolution scheme. To keep the typical wall characteristics without demanding a fine grid, the common wall functions are transferred to the domain internal sediment walls. Furthermore, additional modifications are applied to model a solid sediment wall inside the solution domain. The new internal wall function implementation is validated with a 2D test case. The results show a very good agreement to common wall functions and a significant improvement compared to its negligence. Furthermore the solver is used to simulate the scour downstream of an apron and the scour around a vertical cylinder in current. The results are compared to experiments presented in the literature and show good agreement. The applicability onto arbitrary structures is demonstrated by applying the solver onto a vertical cylinder with a mudplate. The current development state is able to resolve all important physical flow and scour phenomena. The results also unveil that modeling of the suspension and the treatment of the internal wall need additional attention.

Keywords:
Scour, Sediment, Volume-of-Fluid, Bingham, Finite Volume Method, Internal Wall Function, OpenFOAM

1 Introduction

Scour is the erosion of sediment on the seabed near offshore structures like offshore wind power stations. To avoid an excessive weakening of the structural strength the driving depth of respective offshore foundings is increased and scour protection systems might be applied. Furthermore the classification society may dictate the application of scour monitoring systems. As all measures result in substantial extra costs it is of large interest to be able to predict the magnitude of the scour. Currently most common prediction methods are based on very simple empirical equations, which are restricted to be applied to simple obstacles like circular piles. They may overestimate the scour many times.

In the last decades one has begun to develop simulation methods for the scour prediction. In this context it has been shown that the long time scale of the scour development in combination with the small time scale of the vortices shedded by the offshore structures is challenging and leads to high computational costs. Furthermore, the structures can be of complex shape (for example mudplates). In addition the forces are not easy to model as they are often generated by a combination of a current and irregular waves.

The flow and scour around a vertical cylinder exposed to current were investigated experimentally and numerically by Roulund et al. (2005). The sediment shape is represented by a domain boundary. The numerical model is based on the Exner equation and a grid morphing approach to calculate the sediment motion. The numerical simulation captures all the main features and agrees reasonably well with the experiment.

The numerical simulation of the flow and scour around

*Corresponding author
Email address: info@janekmeyer.de

a vertical cylinder exposed to steady current was investigated systematically by Baykal et al. (2015). The numerical model is also based on the Exner equation and a grid morphing approach. It is shown that both, vortex shedding and the suspended load, needs to be resolved to get the correct sediment shape downstream of the pile. This study was extended to waves and backfilling by Baykal et al. (2017). It is shown that the numerical model is applicable to scouring and backfilling under waves. In advance, the vortices around a vertical wall-mounted cylinder exposed to waves were investigated by Sumer et al. (1997). Furthermore the backfilling of a scour hole around a pile in waves was investigated by Sumer et al. (2012). A first review of the applied methods has been given in Sumer et al. (2001) and second review has been given in Sumer (2007). These methods are principally based on empirical bedload formulas.

Another model based on the Exner equation and a grid morphing approach was presented by Stahlmann (2013). The model is used to simulate the scour around a tripod structure in waves. Experimental results are also given for this scenario and a good agreement between the simulation and the experiment has been observed.

A new mesh deformation method was presented by Sattar et al. (2017). The mobile bed is modeled using the Exner equation. As stated out, the model has overcome major limitations regarding mesh deformation and large amplitude bed movement.

A different modeling approach, which does not require conventional bedload and suspended load assumptions was described by Cheng et al. (2017). The method uses a multi-dimensional eulerian two-phase model with proper closure terms for the momentum exchange to model the sediment transport and it does not require empirical bedload formulas. The Volume-of-Fluid (VOF) method is used to describe the sediment shape. Furthermore, a modified version of the k-ϵ turbulence model is applied. The method has been applied to small-scaled problems like steady and oscillatory sheet flow or scour downstream of an apron. Additional enhancements to the model were presented by Chauchat et al. (2017), where also a modified k-ω turbulence model is added. In Nagel et al. (2017) and Nagel et al. (2020) this model has been used to simulate the scour around a vertical cylinder in current. The presented results show a qualitatively good agreement to the experiment on the upstream side of the pile. However, on the downstream side a sediment accumulation is reported instead of an erosion. Furthermore, the simulations require relatively long computational costs and 6,000 CPU hours were required for only 10 seconds of the simulation.

Another different method to simulate fluid-soil-body interaction was given by Voelkner et al. (2015). The soil is modeled using a Bingham model and the sediment shape is described with the Volume-of-Fluid method. The method has been applied onto a three-phase flow dam break test, a soil collapse test and offshore groundings and jacking operations in current, waves and wind. The work has its focus on the grounding and jacking operations and an application onto a scour problem is not given. Similar Bingham-based approaches to simulate the soil were given by Ulrich et al. (2013), Fourtakas and Rogers (2016) and Manenti et al. (2012). These works use the Smoothed-Particle-Hydrodynamics method to calculate the flow and to represent the shape of the sediment. In Manenti et al. (2012) the Mohr-Coulomb Erosion Criterion is used to calculate the Bingham-viscosity. The Bingham approach has been compared to the Shields Erosion Criterion by applying both onto a 2D flushing simulation. The comparison to the flushing experiment shows, that both approaches reproduce the main features. Nevertheless, Manenti et al. (2012) concludes that the Shields Erosion Criterion seems preferable for practical applications. Furthermore, the Bingham model approach was used earlier by Liu and Mei (1989) and Huang and García (1997).

Our main goal of this work is to develop a scour simulation method which is applicable onto complex arbitrary shaped offshore structures. Furthermore, the solution time should allow to use the method for industrial applications. For the final method, the environmental forces acting onto the structures should result from current, waves or a combination of both. Nevertheless, as the first developing step this paper will only consider currents. All methods based on grid morphing are systematically excluded for an application on arbitrary complex shaped structures. Whereas the Volume-of-Fluid method has a good potential for such cases and has been chosen for this work.

The sediment solver is implemented as an addition to our free-surface OpenFOAM-solver (Meyer et al., 2016). This free-surface solver has been used in a variety of industrial and scientific projects (Graf et al. (2016) and Graf et al. (2017)) and is also able to calculate high quality wave simulations (Meyer et al., 2017).

The paper first describes the numerical model with all important details. Afterward, the method is validated qualitatively against a wall-function test case, scour downstream of an apron and scour around a vertical cylinder in current. Furthermore, a first application on a complex shaped geometry is given by investigating a vertical cylinder with an additional mudplate in current.

2 Methods

2.1 Governing equations

For the calculation of the free-surface flow the incompressible unsteady Reynolds-averaged Navier-Stokes equations are solved using the finite volume method. The Volume-of-Fluid (VOF) method introduced by Hirt and Nichols (1981) is used for the calculation of the free-

surface. The momentum conservation equation, the continuity equation and the conservation equation for the transport of the volume fraction α are defined as

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \nabla \cdot \mu_e \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) = -\nabla p + \rho \mathbf{g} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\alpha_i \mathbf{u}) = 0 \quad (3)$$

with the volume fraction α_i for the ith phase, the velocity vector $\mathbf{u}$, the pressure p, the gravity vector $\mathbf{g}$, the density ρ and the effective dynamic viscosity $\mu_e = \mu_l + \mu_t$ consisting of the laminar dynamic viscosity μ_l and the dynamic eddy viscosity μ_t. The flow properties are then calculated by

$$\rho = \sum_i \rho_i \alpha_i \ , \quad \mu = \sum_i \mu_i \alpha_i \ \text{ and } \ 1 = \sum_i \alpha_i \ . \quad (4)$$

The free-sediment-surface is defined by the volume fraction $\alpha_s = 0.6$, with α_s being the volume fraction of the sediment phase. The dynamic eddy viscosity μ_t is calculated with a modified version of OpenFOAMs implementation of the k-ω-SST two equation model (Menter et al., 2003). The original implementation in OpenFOAM excludes the density from space and time derivations which is only a valid operation for one-phase flows with constant density. A suitable formulation for the turbulent kinetic energy k and the specific dissipation rate ω in a multi-phase flow is

$$\frac{\partial \rho k}{\partial t} + \nabla \cdot (\rho \mathbf{u} k) - \nabla \cdot \left(\rho \left(\nu_l + \alpha_k \nu_t\right) \nabla k\right) = \rho P_k - \rho \beta^* k \omega \quad (5)$$

and

$$\begin{aligned} &\frac{\partial \rho \omega}{\partial t} + \nabla \cdot (\rho \mathbf{u} \omega) - \nabla \cdot \left(\rho \left(\nu_l + \alpha_\omega \nu_t\right) \nabla \omega\right) \\ &= \rho \gamma S_2 - \rho \beta \omega^2 + 2 \left(1 - F_1\right) \rho \frac{\alpha_{\omega 2}}{\omega} \left(\nabla k \cdot \nabla \omega\right) \end{aligned} \quad (6)$$

with

$$S_2 = \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) : \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) \quad (7)$$

where : stands for the double inner product,

$$G = \nu_t S_2 \quad (8)$$

and the production rate for the kinetic energy

$$P_k = \min\left(G, 10.0 \beta^* \omega\right) \ . \quad (9)$$

The closure coefficients are $\alpha_{\omega 1} = 0.5$, $\alpha_{\omega 2} = 0.856$, $\alpha_{k1} = 0.85$, $\alpha_{k2} = 1.0$, $\beta_1 = 0.075$, $\beta_2 = 0.0828$, $\gamma_1 = \frac{5}{9}$, $\gamma_2 = 0.44$, and $\beta^* = 0.09$. The values for α_ω, α_k, β and γ are than calculated with the following generalized blending function

$$\phi_{\text{blend}} = F_1 \phi_1 + \left(1 - F_1\right) \phi_2 \quad (10)$$

with the blending factor

$$F_1 = \tanh\left(\left(\min\left(\min\left(F^*, \frac{4 \alpha_{\omega 2} k}{\text{CD}^+_{k\omega} y^2}\right), 10\right)\right)^4\right) \quad (11)$$

with the wall distance y,

$$F^* = \max\left(\frac{k^{0.5}}{\beta^* y \omega}, \frac{500 \nu_1}{y^2 \omega}\right) \ , \quad (12)$$

and

$$\text{CD}^+_{k\omega} = \max\left(\text{CD}_{k\omega}, 10^-10\right) \quad (13)$$

with

$$\text{CD}_{k\omega} = \frac{2 \alpha_{\omega 2}}{\omega} \nabla k \cdot \nabla \omega \ . \quad (14)$$

The blending value F_2 is calculated by

$$F_2 = \tanh\left(\left(\min\left(F^*, 100\right)\right)^2\right) \ . \quad (15)$$

Finally, the kinematic and the dynamic eddy viscosities are calculated by

$$\nu_t = \frac{a_1 k}{\max\left(a_1 \omega, F_2 S_2^{0.5}\right)} \quad (16)$$

and

$$\mu_t = \rho \nu_t \ . \quad (17)$$

For all equations the finite volume method is used to discretize the spatial derivatives. After integrating the equations over the volume, the resulting volume integrals are converted to surface integrals using the Gauss theorem. This leads to a face based method, requiring interpolation schemes, which interpolate the cell-centered variables to the faces.

2.2 Sediment simulation

2.2.1 Bingham model

The sediment phase is modeled with an approach based on Bingham models (Voelkner et al., 2015). Voelkner et al. (2015) used the method to simulate offshore grounding and jacking operations. Treating the sediment as a Bingham plastic, allows to model the sediment as a non-Newtonian fluid using a variable dynamic viscosity μ_{Bingham} in the diffusive term of the momentum equation (1). This leads to the following modified momentum equation

$$\begin{aligned} &\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \nabla \cdot \left(\mu_e + \mu_{\text{Bingham}}\right) \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) \\ &= -\nabla p + \rho \mathbf{g} \end{aligned} \quad (18)$$

If the local shear stress in the sediment is below the sediment specific yield stress the sediment should behave as

a solid. The solid behavior is modeled by using a relatively high local dynamic Bingham viscosity. If the local shear stress is higher than the yield stress the sediment begins to behave like a fluid. This behavior is modeled by reducing the local dynamic Bingham viscosity.

For the calculation of the Bingham viscosity Voelkner et al. (2015) distinguish between the Bingham viscosity μ_{soil} for the soil and the Bingham viscosity μ_{susp} for the suspension. Furthermore, we use a different interpolation scheme for the soil viscosity. Considering the later applied finite volume method and Gauss theorem, we will write the momentum equation as

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \nabla \cdot (\mu_e + \mu_{\text{susp}}) \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) - \nabla \cdot \mu_{\text{soil}} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) = -\nabla p + \rho \mathbf{g} \tag{19}$$

using two diffusive terms emphasizing the two different interpolation schemes applied later. To calculate the soil viscosity Voelkner et al. (2015) use a combination of the von Mises and Mohr-Coulomb yield criterion leading to the following equation for the dynamic soil viscosity μ_s

$$\mu^*_{\text{soil}} = \frac{\tau_f}{(4j)^{0.5}} \tag{20}$$

Here, τ_f is the yield stress and j is the second invariant of the strain rate tensor. The yield stress τ_f is estimated with the von Mises and Mohr-Coulomb yield criterion

$$\tau_f = p_{\text{rel}} \sin(\phi) + c \cos(\phi) \tag{21}$$

with the relative pressure p_{rel}, the cohesion c and the internal friction angle ϕ. The calculation of the relative pressure is not straightforward and our own approach is described in section 2.2.4. The second invariant of the strain rate tensor is calculated by

$$j = 0.5 \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) : \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) \tag{22}$$

where : is the double inner product of two tensors. For the calculation of the suspension viscosity Voelkner et al. (2015) give the following equation

$$\mu^*_{\text{susp}} = \frac{\tau_w}{(4j)^{0.5}} \tag{23}$$

with

$$\tau_w = C_f \rho_G \mathbf{u} \cdot \mathbf{u} \ . \tag{24}$$

Here C_f is the empirical friction coefficient set to 0.01 and ρ_G is the density of the granular soil phase.

The estimated sediment viscosities are bounded between a lower and an upper bound $\mu_{s_{\text{lower}}}$ and $\mu_{s_{\text{upper}}}$. For the lower and upper bounds, values of $25 \frac{Ns}{m^2}$ and $1500 \frac{Ns}{m^2}$ were suggested by Voelkner et al. (2015) . These bounds were estimated by simulating a soil collapse test with different values. This test does not reflect the main phenomenons of scouring and our own simulations have shown that the lower limit should be decreased, see section 3.3. The upper bound is adopted as it stands.

For the numerical stability of the solution algorithm, it is common practice to avoid discontinuities of the final spatial characteristics of the viscosities. Nevertheless, we decided to keep the discontinuities at the sediment surface to avoid an influence onto the result from a blending zone. In comparison to previous simulations with a blending zone applied at the sediment surface we could not observe any differences in the solver stability. Equations (25) and (26) show our final applied blending zones.

$$\mu_{\text{soil}} = \begin{cases} 0.0 & \text{if } \alpha_{s_i} \leq 0.6 \\ \mu^*_{\text{soil}} & \text{if } 0.6 < \alpha_{s_i} \end{cases} \tag{25}$$

$$\mu_{\text{susp}} = \begin{cases} 0.0 & \text{if } \alpha_{s_i} \leq 0.005 \\ \frac{\alpha_{s_i} - 0.005}{0.01 - 0.005} \mu^*_{\text{susp}} & \text{if } 0.005 < \alpha_{s_i} \leq 0.01 \\ \mu^*_{\text{susp}} & \text{if } 0.01 < \alpha_{s_i} \leq 0.6 \\ 0.0 & \text{if } 0.6 < \alpha_{s_i} \end{cases} \tag{26}$$

With respect to Nnadi and Wilson (1992) and Ulrich et al. (2013) the suspension viscosity should be blended out below a sediment volume fraction of 0.3. As our current model does not include a proper generation of the suspension layer, we have decided to decrease this value to 0.01. This guarantees an influence of the suspension, although the suspension concentration might be low.

2.2.2 Creeping

To suppress further creeping of the solid sediment Voelkner et al. (2015) suggest to damp the velocities, if the sediment viscosity reaches a specific percentage of the upper bound. Our own way to damp the velocities is based on implicitly relaxing the final momentum equation to a target velocity of $0 \frac{m}{s}$. The local relaxation factor of the ith cell is calculated by

$$r_i = \min\left(1.0, \max\left(0.0, 1.0 - \frac{\mu_i - 0.7 \mu_{s_{\text{upper}}}}{(0.9 - 0.7)\, \mu_{s_{\text{upper}}}}\right)\right) \ . \tag{27}$$

2.2.3 Interpolation of the soil viscosity

Integrating the momentum equation (19) over the volume and applying the Gauss theorem leads to a face based method, requiring interpolation schemes to interpolate the cell-centered variables to the faces. Using the typical linear interpolation for the sediment viscosity in the diffusive term might be the first choice. But, due to the jump behavior of this viscosity such linear interpolation will lead to significant problems.

The following example should explain the problem in more detail. Afterward, the solution of the problem is given. Figure 1 shows a uniform one-dimensional grid

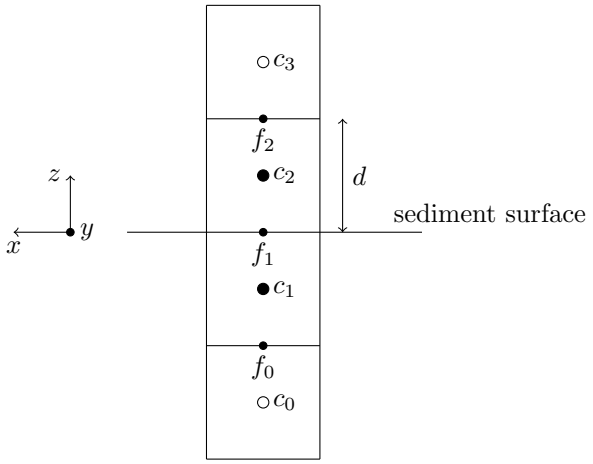


Figure 1: Cells divided by sediment-surface

with four cells in the vertical z-direction. The cells c_0 and c_3 act as fixed value boundary cells. In the two other directions x and y one can assume cells acting as a zero gradient boundary condition. The two upper cells c_2 and c_3 are filled with water and therefore have a sediment viscosity of zero. The two lower cells c_0 and c_1 are filled with sediment acting as a solid and therefore have a sediment viscosity of $\mu_{\text{soil}} = \mu_{s_{\text{upper}}} = 1500\frac{Ns}{m^2}$. The velocities are given in the boundary cells and needs to be calculated for the internal cells c_1 and c_2. The velocities at the boundaries are $u_{x_{c_0}} = 0\frac{m}{s}$ and $u_{x_{c_3}} = u_w \neq 0$. To solve the momentum equation we can make some simplifications. Assuming a steady state solution, the time derivative can be neglected. As the flow is parallel to the sediment surface, the convected term will not influence the result and can be neglected, too. Additionally, we do not consider gravitational forces, also the dynamic pressure gradient is neglected. The continuity equation is inherently fulfilled. The momentum equation (18) then reduces to the diffusive term

$$\nabla \cdot \underbrace{(\mu_e + \mu_{\text{Bingham}})}_{\mu} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) = 0 \ . \tag{28}$$

Integrating over the volume and using the Gauss theorem to convert the volume integrals to surface integrals gives

$$-\mu_{f_0} \left(\frac{\partial u_x}{\partial z}\right)_{f_0} + \mu_{f_1} \left(\frac{\partial u_x}{\partial z}\right)_{f_1} = 0 \tag{29}$$

for cell c_1 and

$$-\mu_{f_1} \left(\frac{\partial u_x}{\partial z}\right)_{f_1} + \mu_{f_2} \left(\frac{\partial u_x}{\partial z}\right)_{f_2} = 0 \tag{30}$$

for cell c_2, while the subscript f_i marks a value at the ith face. Using 2nd order accurate linear interpolation for the viscosity and building the velocity gradient directly at the face without special jump treatment lead to the following equation system

$$-\frac{\mu_{c_0} + \mu_{c_1}}{2}\left(\frac{u_{x_{c_1}} - u_{x_{c_0}}}{d}\right) + \frac{\mu_{c_1} + \mu_{c_2}}{2}\left(\frac{u_{x_{c_2}} - u_{x_{c_1}}}{d}\right) = 0 \tag{31}$$

$$-\frac{\mu_{c_1} + \mu_{c_2}}{2}\left(\frac{u_{x_{c_2}} - u_{x_{c_1}}}{d}\right) + \frac{\mu_{c_2} + \mu_{c_3}}{2}\left(\frac{u_{x_{c_3}} - u_{x_{c_2}}}{d}\right) = 0 \tag{32}$$

with $\mu_0 = \mu_1 = \mu_{s_{\text{upper}}}$ and $\mu_2 = \mu_3 = \mu_w$ and knowing that $\mu_{s_{\text{upper}}} >> \mu_w$ an approximate solution for $u_{x_{c_1}}$ and $u_{x_{c_2}}$ is

$$u_{x_{c_1}} = \frac{\mu_{s_{\text{upper}}}^2 u_{c_0} + \mu_w \mu_{s_{\text{upper}}} (u_{c_3} - u_{c_0}) + \mu_w^2 u_{c_3}}{\mu_{s_{\text{upper}}}^2 + \mu_w^2} \to\sim u_{c_0} \tag{33}$$

$$u_{x_{c_2}} = \frac{\mu_{s_{\text{upper}}}^2 u_{c_0} + \mu_w \mu_{s_{\text{upper}}} (u_{c_0} - u_{c_3}) + \mu_w^2 u_{c_3}}{\mu_{s_{\text{upper}}}^2 + \mu_w^2} \to\sim u_{c_0} \ . \tag{34}$$

The results clearly show, that the linear interpolation leads to a wrong velocity for cell c_2. This is a problem for the scour model, as the flow velocity never reaches the water cell next to the sediment. Therefore no force is acting on the sediment and consequently the sediment viscosity will never be reduced. In Voelkner et al. (2015) it is suggested to use the harmonic mean for the interpolation of the viscosities. In the following it should be shown that the harmonic mean solves the problem. Using the harmonic mean the interpolated face value for a equidistant grid is defined as

$$\phi_f = \frac{2}{\frac{1}{\phi_N} + \frac{1}{\phi_P}} \tag{35}$$

where the subscripts N and P marks the two cells sharing the face f. Interpolating the face values of equations (29) and (30) with the harmonic mean gives the following equation system

$$-\frac{2}{\frac{1}{\mu_0} + \frac{1}{\mu_1}}\left(\frac{u_{x_{c_1}} - u_{x_{c_0}}}{d}\right) + \frac{2}{\frac{1}{\mu_1} + \frac{1}{\mu_2}}\left(\frac{u_{x_{c_2}} - u_{x_{c_1}}}{d}\right) = 0 \tag{36}$$

$$-\frac{2}{\frac{1}{\mu_1} + \frac{1}{\mu_2}}\left(\frac{u_{x_{c_2}} - u_{x_{c_1}}}{d}\right) + \frac{2}{\frac{1}{\mu_2} + \frac{1}{\mu_3}}\left(\frac{u_{x_{c_3}} - u_{x_{c_2}}}{d}\right) = 0 \tag{37}$$

Solving this equation system under the previous assumptions leads to following approximate solution

$$u_{c_1} = \frac{3\mu_{s_{\text{upper}}} u_{c_0} + \mu_w (u_{c_0} + 2u_{c_3})}{3\left(\mu_{s_{\text{upper}}} + \mu_w\right)} \to\sim u_{c_0} \tag{38}$$

$$u_{c_2} = \frac{3\mu_w u_{c_3} + \mu_{s_{\text{upper}}} (2u_{c_0} + u_{c_3})}{3\left(\mu_{s_{\text{upper}}} + \mu_w\right)} \to\sim \frac{1}{3} u_{c_3} \ . \tag{39}$$

The results show that a significant amount of the flow velocity is reaching the cell next to the sediment, while the sediment cells still keep their correct velocity. To the authors knowledge, it is not allowed to use the harmonic

mean for the interpolation of the effective viscosity μ_e without additional adjustments. Therefore only the soil viscosity μ_{soil} is interpolated with the harmonic mean, which leads to the second diffusive term in equation (19) for the final implementation in OpenFOAM.

2.2.4 Relative pressure

In Manenti et al. (2012), Ulrich et al. (2013) and Fourtakas and Rogers (2016) the relative pressure used in equation (21) is calculated through the equation of state for a weakly compressible fluid. As we do not simulate a compressible fluid we are introducing a new way to estimate the relative pressure. For our estimation, we assume that the relative pressure is zero at the sediment surface and only has a vertical spatial dependency. The basic idea is to calculate the relative pressure p_{rel} with the following Poisson equation

$$\nabla \cdot (\mathbf{Z} \cdot \nabla p^*_{\text{rel}}) = \nabla \cdot (\tilde{\rho}\mathbf{g}) \tag{40}$$

with the gravitational acceleration vector $\mathbf{g}$, the effective density $\tilde{\rho}$ resulting of the sand particles without the poor-water and the tensor $\mathbf{Z}$ which decouples the pressure from the horizontal environment. The * means that this is not the final relative pressure and a final correction will follow. The effective density is calculated by

$$\tilde{\rho} = \begin{cases} \rho_{\text{rock}}\,(1-\phi_p) & \text{if } \alpha_s \geq \alpha_{\text{wall}} \\ 0 & \text{if } \alpha_s < \alpha_{\text{wall}} \end{cases} \tag{41}$$

with ϕ_p as the porosity of the sediment, ρ_{rock} as the density of the sand grain and $\alpha_{\text{wall}} = 0.6$ defining the sediment surface. The tensor $\mathbf{Z}$ is defined as

$$\mathbf{Z} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{42}$$

Using this tensor is the main trick to calculate the effective pressure. It ensures that the pressure is always zero at the sediment surface and that the pressure only depends on the sediment depth. But, using this tensor leads to numerical problems. First of all it is necessary to have a Dirichlet boundary condition on the top of the domain. For example, a Dirichlet boundary condition only at the outlet will lead to problems as the internal cells will not be influenced by this boundary condition due to the decoupling tensor $\mathbf{Z}$. Nevertheless, the solution of this equation always diverged in the first timesteps. Our solution is to implement the decoupling approach with the help of the deferred correction method. That means, the Laplace term containing the tensor $\mathbf{Z}$ is treated explicitly on the right hand side. Additionally, the same Laplace term, but without the tensor $\mathbf{Z}$ is added on both implicit and explicit sides. This leads to the following equation for the relative pressure

$$\nabla \cdot \left(\nabla p^{*^{q+1}}_{\text{rel}}\right) = \nabla \cdot (\tilde{\rho}\mathbf{g}) + \nabla \cdot \left(\nabla p^{*^{q}}_{\text{rel}}\right) - \nabla \cdot \left(\mathbf{Z} \cdot \nabla p^{*^{q}}_{\text{rel}}\right) \tag{43}$$

where the superscript q marks the already known solution of the last iteration and $q+1$ the new solution solved in the current iteration. If the equations system is converging the Laplace terms without the tensor $\mathbf{Z}$ are canceling each other out and the solution is the same as for the equation (40). The converging behavior of this equation has been satisfying in all of our simulations. Still small errors in the results for the relative pressure are possible and in very few cells containing water a noticeable high relative pressure occurred. A vertical advancing of this error in the water cells is inherently given by the zero gradient ($\nabla\tilde{\rho}\mathbf{g}$) inside the water cells. To avoid this errors, resulting in pressure peaks outside the sediment we do an explicit correction.

$$p_{\text{rel}} = \begin{cases} p^*_{\text{rel}} & \text{if } \alpha_s \geq 0.99\alpha_{\text{wall}} \\ 0 & \text{if } \alpha_s < 0.99\alpha_{\text{wall}} \end{cases} \tag{44}$$

This correction suppresses the mentioned problem successfully. Nevertheless, an explicit correction, which is not included in the equation system may destroy the converged solution. In our case, this explicit correction leads to a slow convergence and in a few simulations wrong results for the relative pressure occurred for a few timesteps leading to big errors in the result. Therefore, the last step was, to include this correction into the equation system. This can be achieved easily, using an implicit relaxation to a target pressure of zero for the sediment cells. After discretizing equation (43) the final equation system can be written as

$$a_{\text{d},p_{\text{rel}}} p^*_{\text{rel}} + \sum_n a_{\text{n},p_{\text{rel}}} p^*_{\text{rel}} = s_{\text{p}_{\text{rel}}} \tag{45}$$

with $a_{\text{d},p_{\text{rel}}}$ as the main diagonal elements of the coefficient matrix, $a_{\text{n},p_{\text{rel}}}$ as the neighbor elements of the coefficient matrix and $s_{\text{p}_{\text{rel}}}$ as the right hand side. Using the implicit relaxation one gets the following equation

$$a_{\text{d},p_{\text{rel}}} p_{\text{rel}} + r \sum_n a_{\text{n},p_{\text{rel}}} p_{\text{rel}} = r s_{\text{p}_{\text{rel}}} + (1-r)\, a_{\text{d},p_{\text{rel}}} p^t_{\text{rel}} \tag{46}$$

with the relaxation factor r, and the target pressure p^t_{rel}, which is zero. The relaxation factor r is calculated with the help of the sediment volume fraction

$$r = \begin{cases} 1.0 & \text{if } \alpha_s \geq 0.99\alpha_{\text{wall}} \\ 0.0 & \text{if } \alpha_s < 0.99\alpha_{\text{wall}} \end{cases} . \tag{47}$$

With this approach the convergence of the equation system for the relative pressure is very fast. And we did not observe any more instabilities.

2.2.5 Sliding model

As stated out in Roulund et al. (2005) observations have shown, that on the upstream side of the scour hole the slope might exceed the angle of repose by a few degrees.

As a result, shear failures occur at these locations. For the simulation, Roulund et al. (2005) suggests a sliding model, which moves the sediment until the bedslope deceeds the angle of repose by two degrees. The whole sliding process is done in the same time step and not resolved in a real transient way. As this method is based on a mesh boundary representing the sediment surface, it is not directly applicable to our Volume-of-Fluid approach. Therefore, we have developed a new sliding model based on a similar idea. The new model can be concluded as follows:

- Mark the sediment surface cells: this can be done by iterating over the internal faces and comparing the volume fraction of the sediment α_s of the two cells sharing this face.
- Calculate the gradient of the sediment volume fraction $\nabla\alpha_s$.
- For all sediment surface cells: calculate the angle between the local vector $\nabla\alpha_s$ and the gravity vector $\mathbf{g}$.
- If the local angle succeeds the angle of internal friction, set the local sediment viscosity μ_s to zero.

Compared to the sliding models presented in Roulund et al. (2005) and Stahlmann (2013), our new sliding model does not slide the sediment explicitly but it removes the local withstanding force by reducing μ_s. Therefore the external forces can act unrestricted. If the gravitational forces dominates, the sediment will slide down. If another force dominates the sediment may also be picked up by the flow, for example.

2.2.6 Internal wall function

Resolving the boundary layer above the sediment surface correctly is important as the boundary layer size has a significant influence onto the horseshoe vortex. Simulation methods based on "grid deforming" as described in Roulund et al. (2005) can use the standard well-known wall functions applied to the domain boundary representing the sediment surface. However, since we use the VOF-method to represent the sediment, its surface is not linked to a domain boundary and standard wall functions are not applicable. A potential solution could be to resolve the boundary layer with a very fine grid. But this will increase the computational costs significantly, especially because the whole region of the scour hole development needs to be resolved with the finest cell size. Therefore we have implemented a new wall function approach in a way allowing it to act on the domain-internal sediment wall. To achieve this, we use the same principle as the common wall functions. In the following we will first describe the high-level principle of OpenFOAMs standard wall function for smooth walls. Afterward, the new internal wall function implementation will be explained.

Using the k-omega-SST turbulence model in combination with OpenFOAM's *omegaWallFunction*, *nutUWallFunction* and *movingWallVelocity* the solution process of the turbulence model with an applied wall function can be concluded on the following points:

- Explicitly calculate the wall distance y.
- Explicitly calculate G (8) and the turbulent production rate P_k (9).
  At walls using a wall function:
  for the near wall cells overwrite G with a different explicitly calculated result (requires y). This also changes P_k as its calculation uses G.
- Set up the equation for ω (see equation (6)).
  At walls using a wall function:
  manipulate the coefficient matrix for the near wall cells to force a different, explicitly calculated result for these cells (requires y).
  At the wall set the previously calculated value of the near wall cell for ω. Solve the equation for ω.
- Set up the equation for k (requires P_k, see equation (5)).
  At the wall a Neumann boundary condition with a zero gradient is applied for k. Solve this equation for k.
- Calculate the turbulent kinematic viscosity ν_t explicitly (see equation(16)).
  At walls using a wall function:
  for the wall faces overwrite ν_t with a different explicitly calculated result (requires y).

Summarizing the important steps for a wall function: It is necessary to have y, a different G at the near wall cell, a different ω at the near wall cell and the face of the wall, a zero gradient boundary condition for k and a different ν_t at the face of the wall.

For our new, domain internal implementation we use the same equations as the original implementation and just redefine the wall distance y, the near wall cell and the face of the wall as shown in figure 2. Identifying the wall cells can be done by iterating over all internal faces and comparing the sediment volume fraction of both cells sharing this face. The sediment wall lies inside these cells, if one cell has a sediment volume fraction smaller than 0.6 and the other cell has a sediment volume fraction greater or equal to 0.6. As one can see, we use the half distance between the cell centers to estimate the wall distance y. It might be a more accurate approach to project the distance vector between both cells onto the surface normal of the sediment surface. But this introduces new problems, as the projected distance might become very small or zero. Therefore, we do not use such a projection.

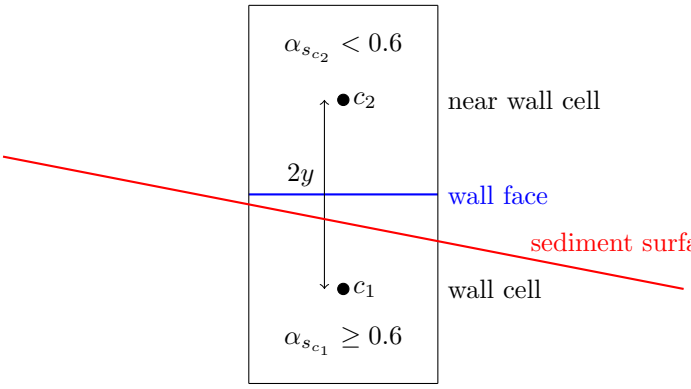


Figure 2: Wall cells at sediment-surface

The cell with a sediment volume fraction smaller than 0.6 is called the near wall cell, the cell with a sediment volume fraction greater or equal to 0.6 is called the wall cell. The face between these two cells is called the wall face. As a simplification, the wall face is assumed to be the sediment surface in both points, position and orientation, although this is not the case predominantly.

Modifications at the near wall cell:
For this cell the modified values for G and ω are used. As previously shown the equation system for ω has to be modified, so that the converged solution contains the modified value for the near wall cells. In the current implementation the equation system is modified with the help of an implicit relaxation. For the near wall cells the equation is relaxed to the individual target value with a relaxation factor of $1e-9$.

Modifications at the wall cell:
The previously modified value for ω in the near wall cell is copied to the wall cell. Therefore a linear interpolation will also lead to this value at the wall face.

To achieve a zero gradient like behavior for k at the wall face, we are modifying the coefficient matrix with an implicit relaxation. For the wall cell the equation is relaxed to the value for k of the near wall cell of the previous iteration.

As mentioned before, the kinematic eddy viscosity ν_t is modified directly at the face of the wall. In the code of OpenFOAM there exist different parts where this viscosity or its dynamic equivalent μ_t or the product of μ_t and $(\nabla \mathbf{u})^T$ are interpolated to the face with the help of a linear interpolation scheme. A modified version of the linear interpolation scheme which manipulates the values for the wall face is possible but complicated. Therefore, we decided to manipulate not the face value, but the value of the wall cell for ν_t in a way, that a linear interpolation will lead to the desired value at the face. Then, assuming an equidistant grid, the kinematic eddy viscosity in the wall cell is set to

$$\nu_{t\text{wallCell}} = \max\left(2\nu_{tf_{\text{target}}} - \nu_{t\text{nearWallCell}}, 0\right) \tag{48}$$

with $\nu_{tf_{\text{target}}}$ as the target value for the face. The limitation to a minimal value of zero is done for stability reasons.
In the current implementation we are using the equations for a smooth wall function. In Roulund et al. (2005) it has been shown that the wall roughness has a small influence onto the result. Therefore, our implementation of a smooth wall function should be a valid starting point. Nevertheless, for future work one should implement a rough wall function.

2.2.7 Special VOF-wall treatment

Using the Volume-of-Fluid method to describe the sediment surface, this surface is usually described by at least two cells. One cell contains a sediment volume fraction greater or equal to $\alpha_s = 0.6$, which we will call the sediment-surface-soil-cell (SS-S-Cell), as it is on the side of the soil. The other cell contains a sediment volume fraction lesser than $\alpha_s = 0.6$ and is called sediment-surface-water-cell (SS-W-Cell), as it is on the side of the water. Such surface description leads to two problems:

(1) Illegal sediment transport:
Using only one velocity field for both phases, as it is the case in our method, one has two problematic possibilities. (a) Applying the soil viscosity on both sediment-surface cells. For a sediment acting as a wall, this will lead to a slow or zero velocity in both cells. Therefore, the water in the SS-W-cell will be too slow and also the internal wall function will not work as desired. (b) The soil viscosity is only applied for the SS-S-cell and not for the SS-W-cell. This leads to an almost correct velocity for the water in the SS-W-cell, but to a too high velocity for the soil in this cell. Therefore, although the soil might act as a solid wall, the sediment in the SS-W-cell is being transported away, which is an illegal sediment motion.

(2) Wrong suspension detection:
As given in equation (26) the suspension is defined in the interval $[0.01, 0.6]$ for the sediment volume fraction plus a blending region. With respect to the given VOF-approach, for most SS-W-cells a suspension will be detected, although the cells contain only such a volume fraction to describe the free surface and not the suspension.

Our solution for the first problem is based on possibility (a), where the soil viscosity is applied on both sediment-surface cells. Furthermore, we do not apply the internal wall function at the sediment-surface face,

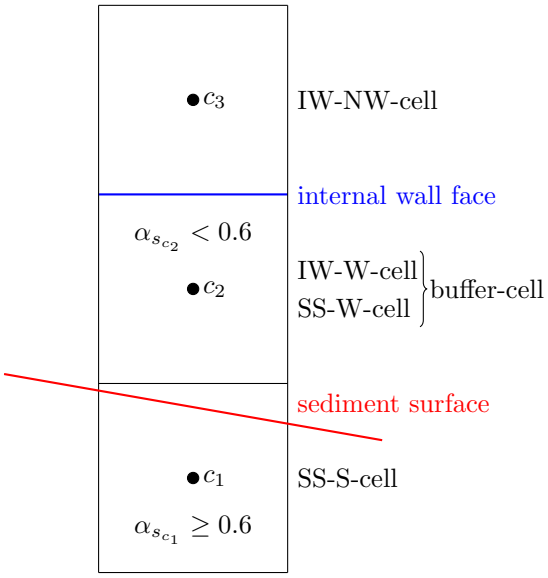


Figure 3: Buffer cell approach

but onto the face shifted one cell away from the sediment. The details are shown in figure 3. The cells c_1 and c_2 are the SS-S-cell and SS-W-cell as described previously. The wall face which is used for the internal wall function is shifted one cell away, compare with figure 2. With respect to the internal wall function, described in 2.2.6 the wall cell is now called the internal-wall wall-cell (IW-W-cell) and the near wall cell is called the internal-wall nearwall-cell (IW-NW-cell). As the SS-W-cell and IW-W-cell are the same and as they build some kind of buffer region, one can also call this cell a buffer-cell. The soil viscosity is calculated in the SS-S-cell and then copied to the buffer-cell. For a correct calculation of this viscosity the second invariant of the strain rate tensor j is calculated at the buffer-cell and copied to the SS-S-cell, before the soil viscosity is being updated.

The identification of the SS-S and SS-W cells is straightforward and can be done by comparing the sediment volume fractions of the two cells sharing one face. After the identification of these cells, the identification of the IW-W and IW-NW cells can be done by comparing the sediment volume fractions and SS-cell identification of the two cells sharing one face.

We would like to emphasize, that this approach is only applied for the simulations in section 3.3. Furthermore, it is important that the cells have a uniform size in the region, where this buffer-cell approach is applied. For future work it might be a better and cleaner solution to use two velocity fields, one for the water phase and one for the sediment face, instead of the buffer-cell approach.

For the solution of the second problem, we stay with the denotation given in figure 3, although it is not necessary to apply the buffer-cell approach to solve the second problem. To check, if the SS-W-cell contains the suspension we do not only use the sediment volume fraction of this cell, but also the sediment volume fraction of the IW-NW-cell. The SS-W-cell is than only considered as a suspension cell, if the volume fraction of the IW-NW-cell exceeds a limit of 0.001.

2.3 Discretization details

All terms of the given equations are discretized with 2nd order accurate schemes. Some local assumptions are possible, where a 1st order scheme is used to fulfill stability criteria as explained later in this chapter. The major parts of the discretization procedure in OpenFOAM are explained in Jasak (1996). In this chapter only the critical parts, which are required for a stable and accurate simulation, are described.

The time derivatives are discretized with a 2nd order scheme. OpenFOAM offers two 2nd order schemes, *backward* and *Crank-Nicolson*. The backward scheme is based on quadratic interpolation using the values of two previous time steps. The Crank-Nicolson scheme is based on the trapezoidal rule. Both schemes may lead to oscillating solutions under specific circumstances. This oscillation tendency is especially large due to a high density-jump at the interface. Knowing, that future simulations will also contain a water-air interface, this problem is severe. To avoid oscillations, we have implemented a modified version of the backward scheme. Under the assumption of a given M-Matrix characteristic as defined by Hackbusch (2017), the modified scheme locally blends to 1st order Euler, if the right hand side of the unmodified backward scheme has a different sign than the right hand side of the Euler scheme. This blending is only necessary in very few cells and numerical tests with sea waves have shown, that the solution still has the accuracy of a full 2nd order solution, while avoiding any oscillations.

The convective terms, except the one of the Volume-of-Fluid equation (3), are discretized with the 2nd order *Gamma* scheme presented in Jasak et al. (1999). The Gamma scheme is based on 2nd order linear interpolation but locally blends to 1st order upwind to fulfill the convective boundedness criterion. Additionally it redefines the normalized variable approach, in a way that it does not require the far upwind node addressing, simplifying the implementation of this scheme for unstrucutred grids.

The convective term of the Volume-of-Fluid equation (3) is discretized with the *blended interface capturing scheme with reconstruction*, presented in Wackers et al. (2011). This scheme is based on downwind differencing as the downwind scheme has a compressive character leading to a sharp interface. The scheme locally blends to the Gamma scheme to fulfill the convective boundedness criterion. Additional local blending is introduced for high Courant-Friedrichs-Lewy-Numbers and for specific angles between the face normal and the normal of

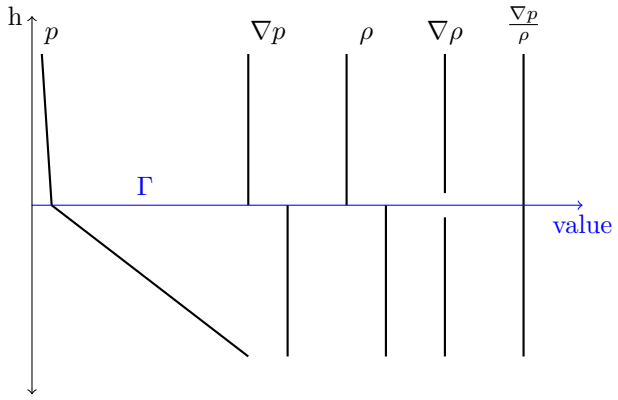


Figure 4: Characteristics of pressure and density at the interface Γ

the free-surface to avoid that the free-surface aligns with the grid. The far upwind node, required for the normalized variable approach, is estimated using a reconstruction of the corresponding value based on a search path algorithm and a weighted interpolation. The details of our implementation are described in Meyer et al. (2016). It is important to know, that such an high-resolution scheme resolves sharp jumps sufficiently but also converts smooth gradients into a sharp jump. This inherently suppresses the development of a suspension. An alternative way, which has been investigated by us, was to use the Gamma-scheme at the sediment surface. The Gamma scheme resolves smooth gradients sufficiently but smears a sharp jump into a smooth gradient. The Gamma scheme allows the development of a suspension. However, the smearing of the sediment interface inherently carries away the sediment, even if the sediment should act as an solid and a zero sediment velocity is given. For now, we have decided to use the mentioned high-resolution scheme.

The density jump at the free-surface leads to a kink in the pressure characteristics and a jump in the pressure gradient as shown in figure 4. Using standard schemes will smear this characteristics. This leads to errors resulting in overestimated velocities in the cells containing the lighter phase directly above the free-surface. This error becomes larger if the cell size is reduced, making grid independence studies unpredictable (Meyer et al. (2016)). To solve this problem the method presented in Queutey and Visonneau (2007) has been implemented. This method reconstructs the jump induced characteristics using the pressure gradient normalized with the density. Our implementation is described in Meyer et al. (2016) and it is also shown, that this approach avoids the unphysical high velocities.

Besides the calculation of the 2nd invariant of the strain rate tensor j, see equation (22), some other terms require the explicit calculation of the cell centered gradient of the velocity $\nabla\mathbf{u}$. In all cases except for the calculation of j the *cellLimited* gradient calculation of OpenFOAM is used. The limitation works as follows

- Calculate the cell centered gradient by converting the volume integrals to surface integrals based on the Gauss theorem or calculate the gradient by using a least squares fit.
- For each face:
  - Extrapolate the cell centered value to the face by using the calculated gradient.
  - Compare the extrapolated face value with the cell centered value of the neighbour cell also sharing this face.
  - If necessary limit the cell centered gradient to guarantee that no extrapolated face value exceeds the cell centered value of the neighbour cell.

This limitation is used as our standard approach to maintain convergence. It is important to notice that it is not allowed to use this gradient limitation for the calculation of j. The starting condition for the sediment always includes a zero velocity inside the sediment. Using the above gradient limitation will inherently reduce j to zero inside the sediment. As j represents the force acting onto the sediment, the gradient limitation prevents the flow from acting onto the sediment.

2.4 Solution method

The coupled equations are solved with a segregated SIMPLE-like algorithm. After integrating over the volume, the Gauss Theorem is used to transform the volume integrals to surface integrals. The linearized, semi-discretized momentum equation (19) can be written as

$$a_\mathrm{d}\mathbf{u}_\mathrm{d} + \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n} = -\nabla p + \mathbf{s}_{\mathrm{w/o\ p}} \quad . \tag{49}$$

Here, a represents the elements of the coefficient matrix $\mathbf{A}$ and the subscripts d and n mark the main diagonal- and neighbor-elements. All sources and contributions to the right hand side except the pressure gradient are included in $\mathbf{s}_{\mathrm{w/o\ p}}$. Rearranging (49) to $\mathbf{u}_d$ yields the velocity equation:

$$\mathbf{u}_\mathrm{d} = \frac{1}{a_\mathrm{d}}\left(-\nabla p + \mathbf{s}_{\mathrm{w/o\ p}} - \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}\right) \quad . \tag{50}$$

Substituting and rearranging (50) into (2) yields the poisson equation for the pressure:

$$\nabla\cdot\left(\frac{1}{a_\mathrm{d}}\nabla p\right) = \nabla\cdot\frac{1}{a_\mathrm{d}}\left(\mathbf{s}_{\mathrm{w/o\ p}} - \sum_n a_\mathrm{n}\mathbf{u}_\mathrm{n}\right) \quad . \tag{51}$$

The key points of the solution algorithm are shown in figure 5. The momentum equation is relaxed implicitly

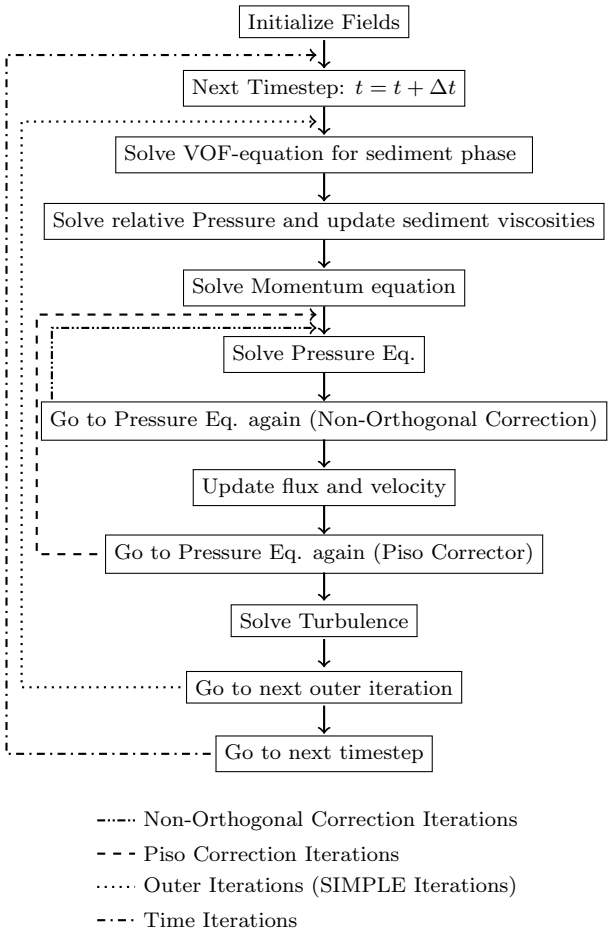


Figure 5: Solution Algorithm

with a relaxation factor of 0.7. The pressure is relaxed explicitly with a relaxation factor of 0.3. The pressure relaxation is done after updating the flux and before updating the velocity. The sediment viscosities are relaxed explicitly with a relaxation factor of 0.1. For all simulations, the relaxation of the sediment viscosities has been absolutely necessary to achieve convergence.

3 Results

3.1 Internal wall function

The internal wall function is verified with a 2D test case. In this test case, the bottom boundary is treated as a wall. The flow is initialized with zero velocity and accelerated with a constant horizontal acceleration so that a boundary layer is being built at the bottom. Three simulation setups are investigated: (i) the sediment surface is represented by a fixed no-slip wall boundary condition employing the standard wall function, (ii) the sediment surface is located inside the domain and represented by the VOF-function without using any special wall treatment and (iii) our new internal wall function is used additionally.
The first simulation is using the standard wall function for smooth walls. The domain has a length of 18.0m and a depth of 22.0m. The cells have a length of 1.0m and a height of 0.0625m at the bottom. The chosen acceleration is $0.00175\frac{\mathrm{m}}{\mathrm{s}^2}$ and the time step size is 0.02s.
For the second simulation the same setup is used, but the bottom domain is extended by 8m. The extended region is filled with sediment. No Volume-of-Fluid transport is calculated as the sediment should act like a constant wall. The sediment viscosity always equals the maximal viscosity of $1500\frac{\mathrm{Ns}}{\mathrm{m}^2}$. For the second simulation no internal wall function approach is applied.
The third simulation equals the second one, but with the new internal wall function applied. The result of the first simulation is used as a reference solution.

Figure 6 shows the velocity profiles for the three simulations at $t = 300$s. The results show, that the velocity profile of the second case is clearly underestimated. Therefore, in a real scour simulation, the forces acting on the sediment are too small. Additionally, anticipating the simulation of subsection 3.3, the horse shoe vortex will not have the correct shape, as this vortex is influenced by the size of the boundary layer. The velocity profile of the third test case using the new internal wall function is significantly better. Figures 7 and 8 show the profile for the kinetic energy k and the specific dissipation rate ω. Again, the new internal wall function improves the result significantly. Only, very close to the wall, the values differ from the reference solution and only have about the half magnitude. However, with respect to equation (16), both errors are canceling each other out, which is also noticeable in figure 9 which shows the profile for the kinematic eddy viscosity ν_t. This figure also shows, that the kinematic eddy viscosity is significantly overestimated, without using the internal wall function. Such a high viscosity will influence the flow substantially. For example the vortices behind a circular pile, which are an important detail of the simulation in subsection 3.3, will be completely different. The test cases show that the new internal wall function is improving the boundary layer flow significantly.

3.2 Scour downstream of an apron

The process of scouring of non-cohesive materials behind an apron was investigated experimentally by Breusers (1965). First equations describing this process are given which allow evaluating the conformity of the scour hole in model and prototype. Here the results are used to validate the simulation method. Figure 10 shows the setup of the test case. At the left side a velocity with a suitable boundary layer is given. The sediment is initialized as a flat bed. It is shown, that the scouring depth d_S increases exponentially according to equation (52), being valid for a wide range of velocities, water depth and

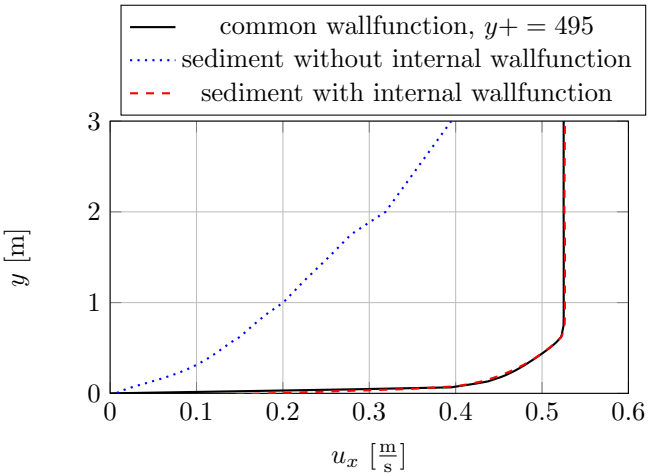


Figure 6: Boundary layer profiles of the horizontal velocity u_x for the three simulations

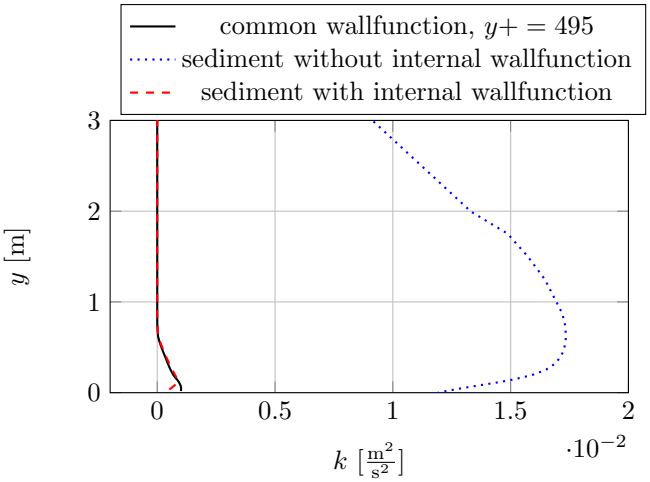


Figure 7: Boundary layer profiles of the kinetic energy k for the three simulations

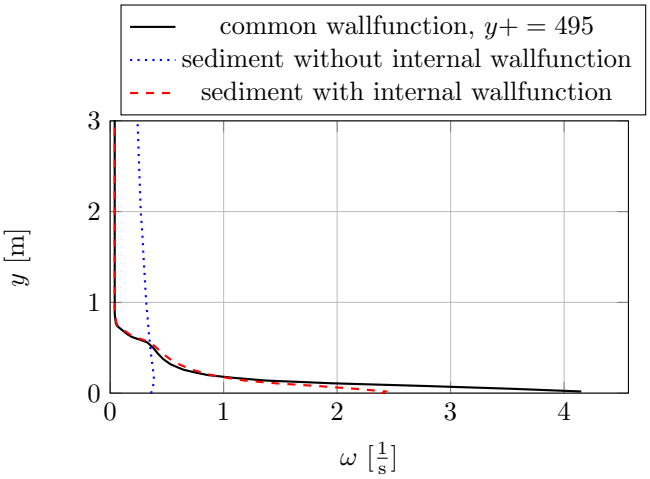


Figure 8: Boundary layer profiles of the specific dissipation rate ω for the three simulations

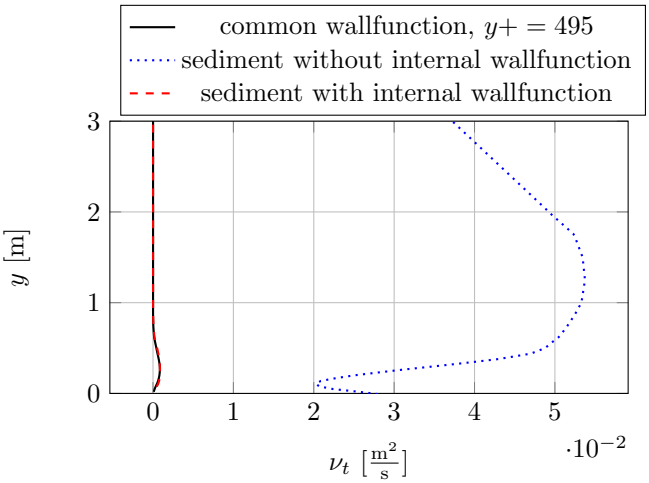


Figure 9: Boundary layer profiles of the kinematic eddy viscosity ν_t for the three simulations

materials (Breusers, 1965).

$$\frac{d_S}{h_0} = \left(\frac{t}{T_S}\right)^{n_S} \tag{52}$$

Here, h_0 is the water depth at the end of the bottom protection, T_S is the characteristic timescale of the scouring process and n_S is the exponent which characterizes the speed of the scour depth growth at the initial stage (Cheng et al., 2017). Additionally, Breusers (1967) described the behavior of the scour angle α_S with

$$\alpha_S = \alpha_{S0}\left(1 - e^{-\frac{t}{T_{\alpha S}}}\right) \tag{53}$$

where α_{S0} is the equilibrium scour angle and $T_{\alpha S}$ is the equilibrium timescale of the scour angle. For fine sand Breusers (1965) gives a range of $13.4° \leq \alpha_{S0} \leq 14.95°$ for the equilibrium scour angle. As mentioned by Amoudry and Liu (2009) the experiments of Breusers (1965), Breusers (1967) and Dietz (1969) show that the shape of the scour hole is almost independent of the flow velocity and the bed grain size if the flow velocity is significantly larger, than the critical velocity required for sediment motion.

Simulationsetup:
The test case is using a two dimensional quadratic domain, with the velocity inlet directly at the edge of the apron. The domain is 1m long and 0.2m height and has a water depth h_0 of 0.15m. The grid cells height is 0.12mm and their length is 0.85mm directly at the edge. To reduce the total number of cells, the cells are stretched with increasing distance to the edge. The grid has 300 cells in horizontal direction and 155 cells in vertical direction with a cell height of 10mm at the top and a cell length of 8.0mm at the outlet on the right side.

The part above the edge of the left side acts as the inlet with Dirichlet boundary conditions for the velocity,

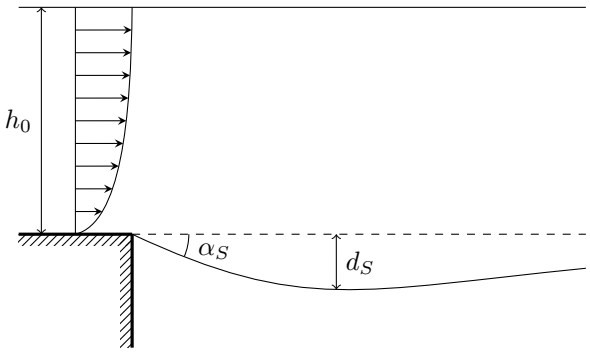


Figure 10: Testcase: scour downstream of an apron

volume fractions, turbulent kinetic energy k and specific rate of dissipation ω. The part below the edge and the bottom of the domain are treated as walls. The right side acts as an outlet with Neumann boundary conditions for all variables. At the top of the domain a Neumann boundary condition is applied for all variables except the pressure which uses a Dirichlet boundary condition. Specifying the pressure at the top instead of the right side allows that the sediment changes its height directly at the outlet. We are not using so called Open Boundary Conditions at the outlet although this has been recommended by Amoudry and Liu (2009) reporting small advantages.

Initial values:
Cheng et al. (2017) imposed a rough-wall log-law velocity profile at the inlet and an one-dimensional simulation was used to get values like the bed concentration. Here, we are using a different approach based on an initial simulation to get the initial values including the velocity and the two turbulence variables k and ω for the inlet. This simulation uses the same mesh, but with a shortened domain using only 10 cells in the horizontal direction. Furthermore, Neumann boundary conditions are given for all variables at the inlet, proper values for the affected variables are still dictated as a Dirichlet boundary condition at the walls. The sediment is fixed and acts as a fixed wall using the maximal Bingham viscosity. Therefore it is not necessary to solve the Volume-of-Fluid transport equation. On the other hand, this means, that no initial suspension layer is produced. The flow is initialized with a velocity of zero, the turbulent variables are initialized with a constant initial guess. Finally a constant acceleration is applied in the whole domain during the whole initial simulation. The simulation is executed until the changes in the velocity and turbulent profiles are negligible. In this study three initial simulations are done with accelerations of $3.5 \cdot 10^{-3} \frac{\mathrm{m}}{\mathrm{s}^2}$, $7.0 \cdot 10^{-3} \frac{\mathrm{m}}{\mathrm{s}^2}$ and $14.0 \cdot 10^{-3} \frac{\mathrm{m}}{\mathrm{s}^2}$. In all three cases the changes of the results are being seen as negligible after 2000s. These initial simulations ensures that the profiles of the affected variables fits perfectly

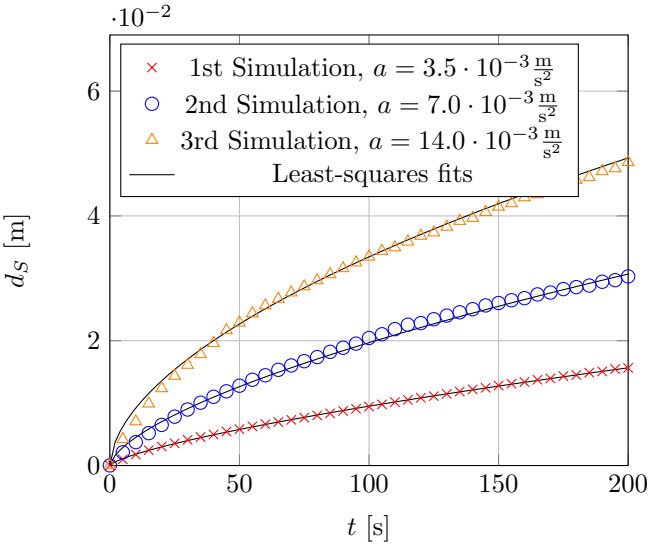


Figure 11: Comparison of simulation results and least-squares fits for the scouring depth

Table 1: Least squares parameters for d_S-function (52) estimated over 200s and α_S-function (53) estimated over 25s.

a $[10^{-3} \frac{\mathrm{m}}{\mathrm{s}^2}]$	T_S [s]	n_S []	$T_{\alpha S}$ [s]	α_{S0} [°]
3.5	4612	0.72	15.2	9.8
7	2387	0.64	5.2	8.54
14	1459	0.56	2.15	8.6

to the given wall function approach.

Scour simulations:
The parameters of the sediment model were adjusted to represent fine sand. Therefore the rock density is $2650 \frac{\mathrm{kg}}{\mathrm{m}^3}$, the saturation is 0.7, the internal friction angle is 25°, the cohesion is $0 \frac{\mathrm{N}}{\mathrm{m}^2}$, and the minimal and maximal Bingham viscosities are set to $0 \frac{\mathrm{Ns}}{\mathrm{m}^2}$ and $1500 \frac{\mathrm{Ns}}{\mathrm{m}^2}$, respectively. Furthermore, the internal wall function approach is activated.

The results are shown in figures 11 and 12. Figure 11 shows the evolution of the scour hole depth d_S. The solid black lines represent the least squares fits to equation (52). The parameters of the optimal fits are given in table 1. For all three cases a very good fit is achieved, which shows that the simulation reflects the sediment behavior reported by Breusers (1965). The estimated values for n_S are similar to the values of the model of Amoudry and Liu (2009) or the model of Cheng et al. (2017). As mentioned by Cheng et al. (2017), referring to Buchko et al. (1988), this values are at the high range but still reasonable. Figure 12 shows the evolution of the scour angle. Again, the results belong to the empirical equation (53). The equilibrium scour angles are signifi-

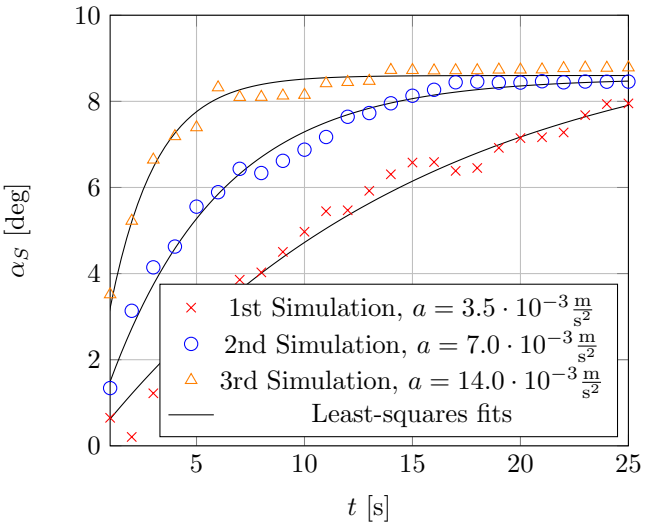


Figure 12: Comparison of simulation results and least-squares fits for the scouring angle

cantly smaller than the given range of the experiments $13.4° \leq \alpha_{S0} \leq 14.95°$ for fine sand. The equilibrium scour angle is nearly the same for all three velocities, which has been expected as the flow velocities are significantly higher than the critical velocity for the motion of the sediment. Figure 13 shows the volume fractions of the sediment and the boundary velocities for different timesteps. The shape of the sediment is plausible but seems to become too edgy with progressing time. Simulations with other approaches for the calculation of the relative pressure have shown that the relative pressure could have a significant influence onto the shape. Interim results with methods, where the relative pressure is not always zero at the sediment surface, show a much rounder shape. Therefore, we assume that the here given method for the calculation of the relative pressure is responsible for the too edgy shape.

Furthermore, one can see that no sediment is brought into suspension. Tests with a given initial suspension layer have shown that the suspension can have a huge influence onto the sediment transport. Especially the equilibrium scour angle increases, which allows to achieve angles in the correct range. Nevertheless at the current development state of the scour solver using such a manually given suspension layer can only be interpreted as *guessing a suspension* or *manipulating the results*. Therefore we are not showing such simulations. But, we assume that a better suspension treatment may lead to better results especially for the equilibrium scour angle.

Figure 14 (a) shows the relative pressure of the simulation at $t = 100$s. The solution for the relative pressure is very clean without any disturbances, as desired. Figure 14 (b) shows the soil viscosity. It shows that the critical velocity for sediment motion is significantly exceeded.

3.3 Scour around a vertical pile

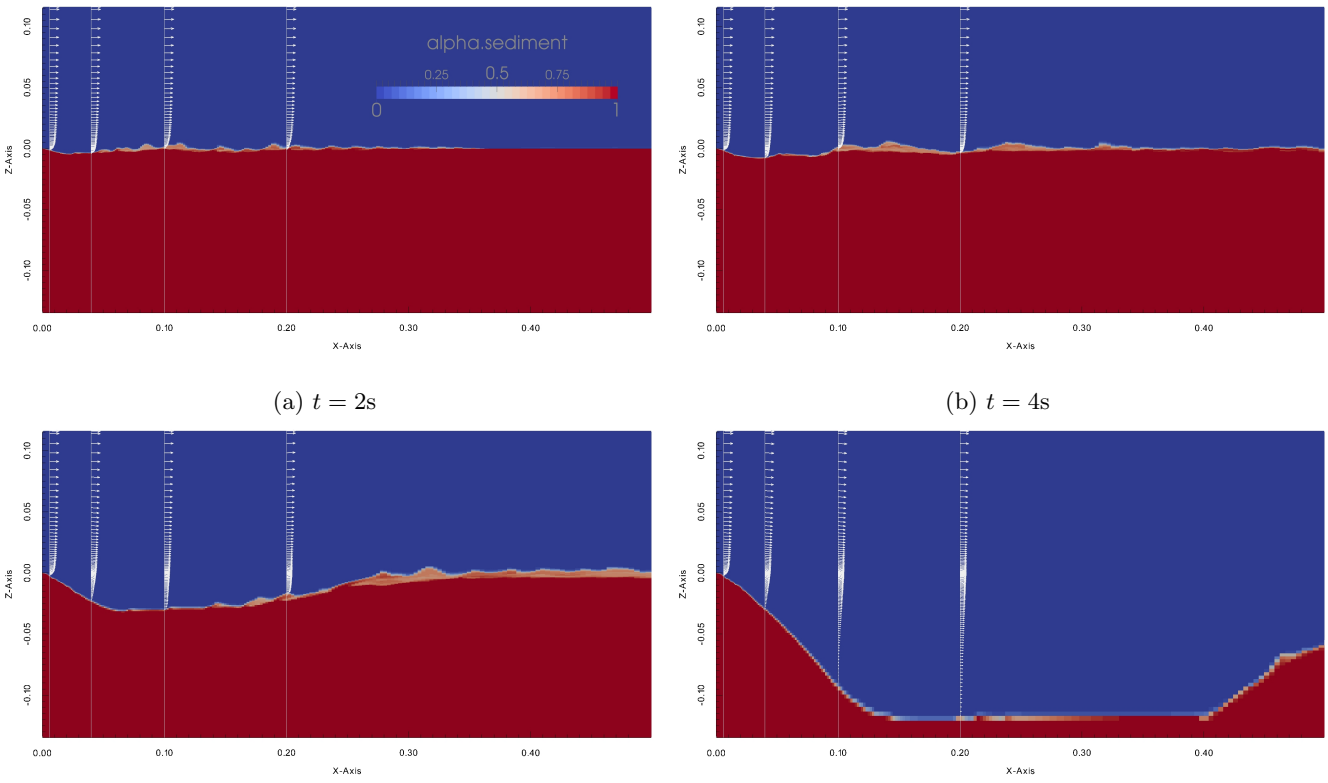
The flow around a vertical circular pile exposed to a steady current is studied numerically and experimentally by Roulund et al. (2005). In the following the main phenomena of this test case should be summarized. The steady current forms a boundary layer at the bottom surface. This boundary layer forces the flow to build a down-flow on the upwind side of the pile. This down-flow leads to a horseshoe vortex in front and around the pile, which then trails off downstream. Additionally, a lee-wake vortex is built downwind of the pile. Furthermore, the streamlines contract at the sides of the pile. All three phenomena increase the sediment transport leading to local scour around the pile.

As shown by Roulund et al. (2005) the time required to build the horseshoe vortex depends on the boundary-layer-thickness to pile-diameter ratio. The smaller this ratio, the longer the delay until the horseshoe vortex is built. For very small ratios it is possible, that no separation is formed. Furthermore, the size of the horse-vortex depends on this ratio and a smaller ratio leads to a smaller vortex.

Simulation Setup:
The pile has a diameter of 0.1m. The domain has a length of 2.1m, a width of 1.2m and a height of 0.75m. It is initialized with sediment up to 0.345m. The domain is discretized with an unstructured hexadominant grid with anisotropic mesh refinement for the sediment surface as shown in figure 15. In the region of the sediment surface, the mesh is refined anisotropically in the vertical direction. The cells have a height of $1.875E-3$m and a length and width of $7.5E-3$m at the sediment surface. The final mesh consist of $2.0E6$ cells.

The velocity, the volume fraction and the turbulence values are given as Dirichlet boundary conditions at the inlet on the left side. The pressure is given as a Dirichlet boundary condition on the right side. The initial values for these Dirichlet boundary conditions were generated using a 2D simulation based on the same principle as in subsection 3.2. The acceleration used to generate the velocity profile was $2.0E-3 \frac{\text{m}}{\text{s}^2}$. A Free-Slip wall boundary condition is applied on the top. It was not possible to achieve a stable simulation using an open lid boundary condition at the top as recommended by Roulund et al. (2005) and Baykal et al. (2015). Symmetry boundary conditions are applied at the sides.

The parameters of the sediment model were adjusted to represent the material of the experiment. The rock density is $2650 \frac{\text{kg}}{\text{m}^3}$, the saturation is 0.7, the internal friction angle is 32°, the cohesion is $0 \frac{\text{N}}{\text{m}^2}$. The maximal Bingham viscosity is set to $1500 \frac{\text{Ns}}{\text{m}^2}$. The simulation was run with different values for the minimal Bingham viscosities. The investigated minimal viscosities are $1.0 \frac{\text{Ns}}{\text{m}^2}$, $12.5 \frac{\text{Ns}}{\text{m}^2}$ and $25 \frac{\text{Ns}}{\text{m}^2}$. The internal wall function



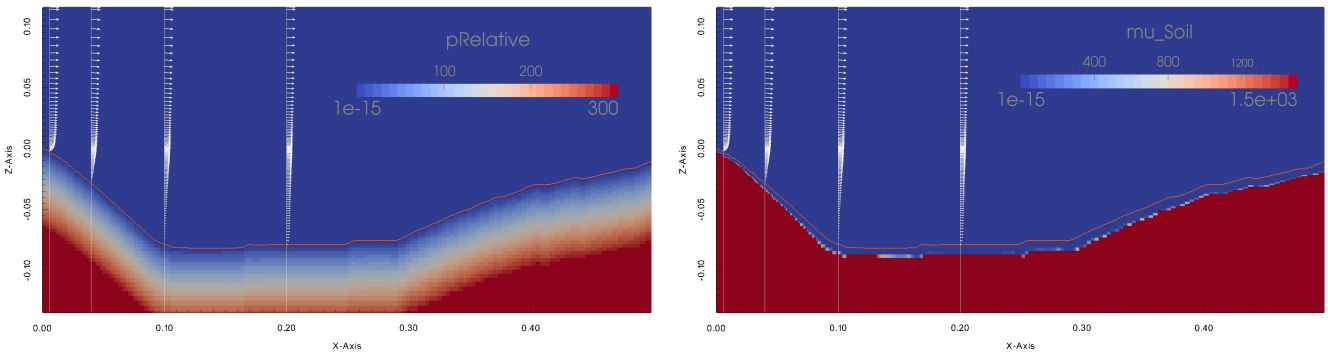
(a) $t = 2$s

(b) $t = 4$s

(c) $t = 25$s

(d) $t = 200$s

Figure 13: Sediment volume Fraction and velocity profiles for different time steps



(a) relative pressure

(b) soil viscosity

Figure 14: Relative pressure and soil viscosity at $t = 100$s. The solid, orange line represents the sediment surface.

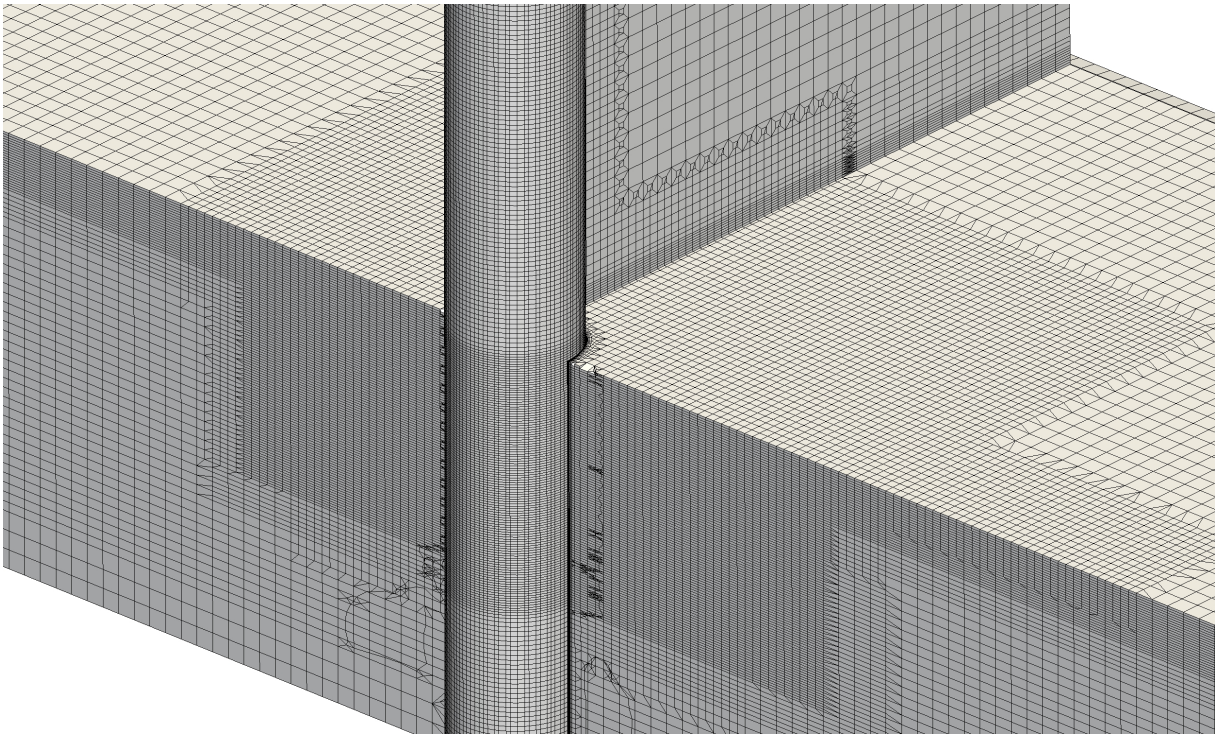


Figure 15: vertical pile in steady current: grid

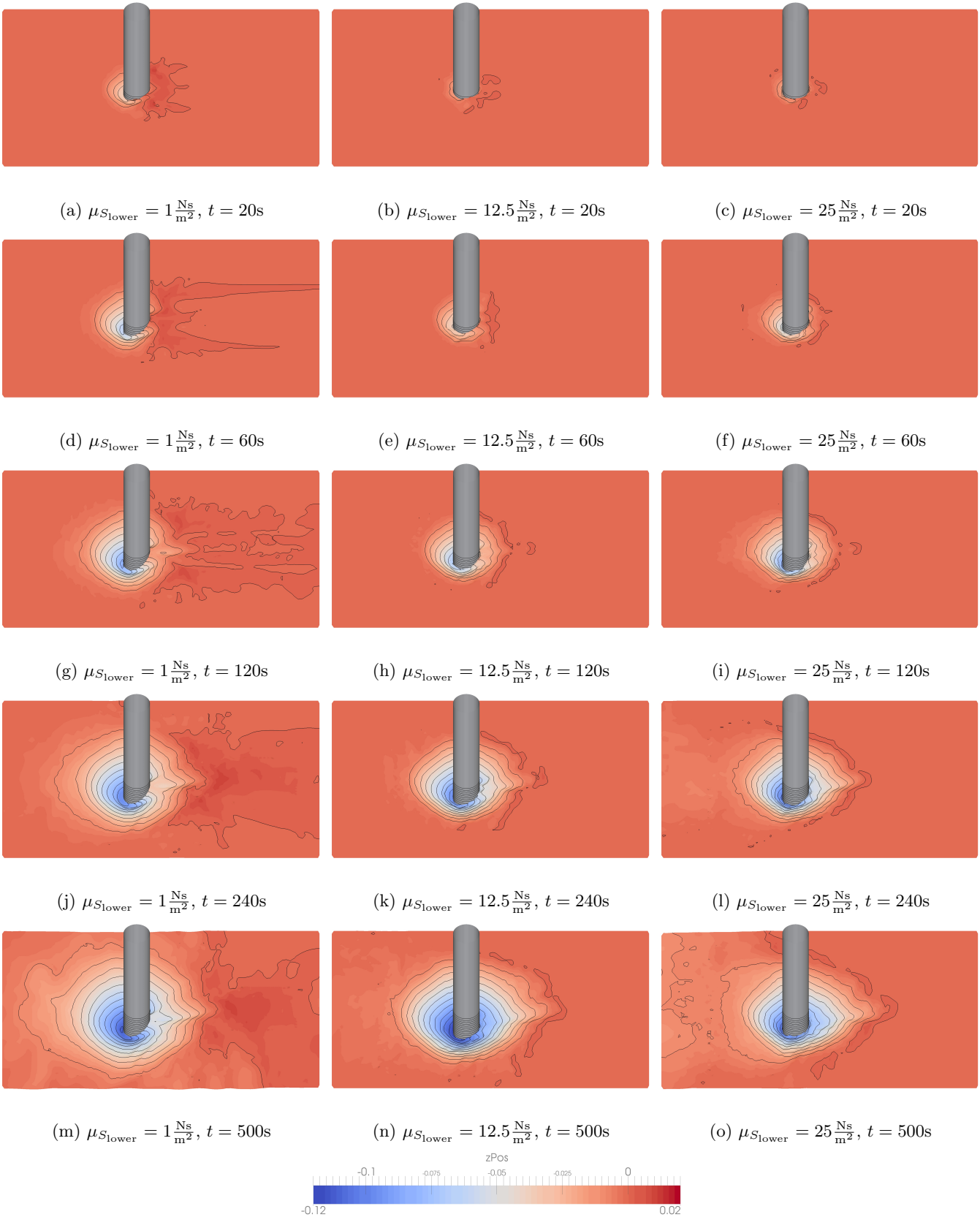
and the buffer cell approach are activated.

Results:
The simulation was executed on a compute cluster with five nodes interconnected via InfiniBand. Each node holds two 6-core CPUs (Intel Xeon E5-2643 v4 @ 3.40GHz) and the calculation time averaged 21 hours for one simulation. Figure 16 shows the sediment surface at different timesteps for the four investigated viscosities. In all cases one can see that the building of the scour begins with the horseshoe vortex on the upwind side of the pile. Subsequently the hole growth around the pile. After some time the lee wake vortex is build and also transports the sediment away from the downstream side of the pile. To emphasize the influence of the different vortices a detailed view is given in figure 17. The vortices are visualized with the help of the Q-Criterion, which is the 2nd invariant of the velocity gradient tensor. In comparison of the simulation results one can see, that the higher the minimal viscosity the steeper the scouring angle on the upstream part. At $t = 240$s the slope has an angle of 27°, 29° and 30° for the three different minimal viscosities. The experiment from Roulund et al. (2005) shows, that this angle should equal the angle of repose, which is not reached completely by our simulations.

In all three simulations, the scouring angle at the downstream side is significantly lower than at the upstream side. At $t = 240$s the slope has an angle of 16°, 18.5° and 20° at the downstream side. A smaller angle is in accordance with the experiment. Furthermore, it has been pointed out, that the scouring depth at the upstream side is higher than on the downstream side. This behavior is resolved in all three simulations till $t = 240$s. Later at $t = 500$s only the simulation with the smallest minimal viscosity is able to hold this characteristic. From comparison with the pictures of the experiment (Roulund et al., 2005) we think that the simulation with the smallest minimal viscosity represents the scouring depth very well.

In all simulations one can observe an erosion in front of the scouring hole. We assume, that the buffer cell approach, presented in subsection 2.2.7, is not sufficient to protect the sediment surface from this erosion. Furthermore, we would like to emphasize that the suspension may have a significant influence onto the result. The suspension increases the erosion at the downstream side of the pile which is in accordance with the observations represented by Baykal et al. (2015), reporting a decrease of 50% if the suspension is neglected. For the simulation with the highest minimal viscosity we observed that the suspension protects the sediment in front of the scouring hole. Without the suspension the erosion would have been much higher in that region.

Due to the sensible reaction to the suspension in combination with an insufficient representation of the suspension generation it is currently not possible to calibrate the minimal soil viscosity and give a final value. Nevertheless, for the presented results all characteristics of the experiments are represented in all three simula-



(a) $\mu_{S_{\text{lower}}} = 1\frac{\text{Ns}}{\text{m}^2}$, $t = 20\text{s}$ (b) $\mu_{S_{\text{lower}}} = 12.5\frac{\text{Ns}}{\text{m}^2}$, $t = 20\text{s}$ (c) $\mu_{S_{\text{lower}}} = 25\frac{\text{Ns}}{\text{m}^2}$, $t = 20\text{s}$

(d) $\mu_{S_{\text{lower}}} = 1\frac{\text{Ns}}{\text{m}^2}$, $t = 60\text{s}$ (e) $\mu_{S_{\text{lower}}} = 12.5\frac{\text{Ns}}{\text{m}^2}$, $t = 60\text{s}$ (f) $\mu_{S_{\text{lower}}} = 25\frac{\text{Ns}}{\text{m}^2}$, $t = 60\text{s}$

(g) $\mu_{S_{\text{lower}}} = 1\frac{\text{Ns}}{\text{m}^2}$, $t = 120\text{s}$ (h) $\mu_{S_{\text{lower}}} = 12.5\frac{\text{Ns}}{\text{m}^2}$, $t = 120\text{s}$ (i) $\mu_{S_{\text{lower}}} = 25\frac{\text{Ns}}{\text{m}^2}$, $t = 120\text{s}$

(j) $\mu_{S_{\text{lower}}} = 1\frac{\text{Ns}}{\text{m}^2}$, $t = 240\text{s}$ (k) $\mu_{S_{\text{lower}}} = 12.5\frac{\text{Ns}}{\text{m}^2}$, $t = 240\text{s}$ (l) $\mu_{S_{\text{lower}}} = 25\frac{\text{Ns}}{\text{m}^2}$, $t = 240\text{s}$

(m) $\mu_{S_{\text{lower}}} = 1\frac{\text{Ns}}{\text{m}^2}$, $t = 500\text{s}$ (n) $\mu_{S_{\text{lower}}} = 12.5\frac{\text{Ns}}{\text{m}^2}$, $t = 500\text{s}$ (o) $\mu_{S_{\text{lower}}} = 25\frac{\text{Ns}}{\text{m}^2}$, $t = 500\text{s}$

Figure 16: Vertical pile in steady current: sediment evolution for different minimal soil viscosities.

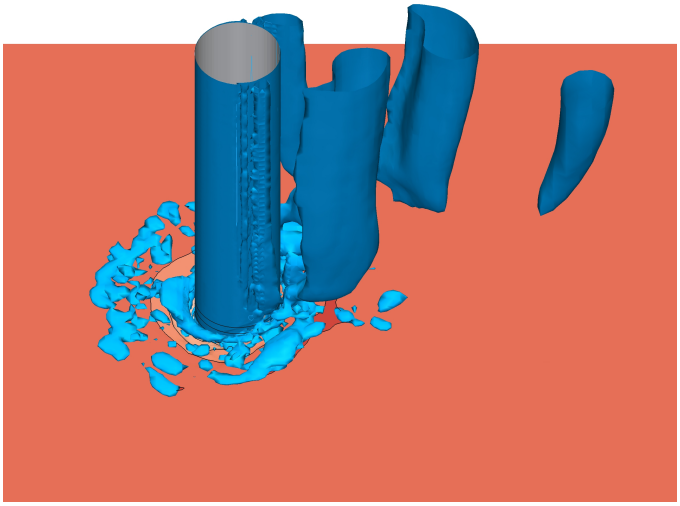
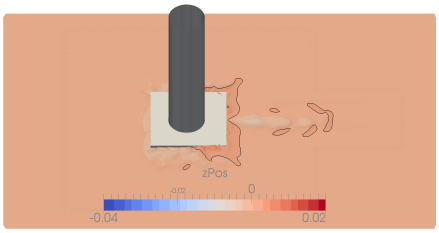


Figure 17: Horse shoe vortex and leewake vortices at $t = 40$s for $\mu_{S_{\text{lower}}} = 25 \frac{\text{Ns}}{\text{m}^2}$

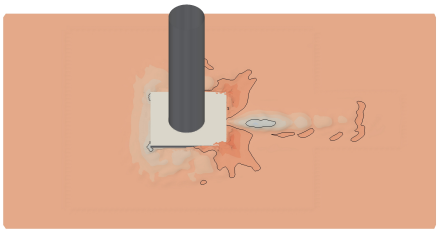
tions and a value of $1.0 \frac{\text{Ns}}{\text{m}^2}$ seems to be the best compromise.

To demonstrate the ability to simulate arbitrary structures, we have done the same simulation with an additional mudplate. Usually such a mudplate prevents scouring as the horse shoe vortex can not act onto the sediment. On the other hand, if scouring occurs, the mudplate can have a negative influence, as the flow is accelerated between the plate and the sediment. Figure 18 shows the simulation results at different timesteps. One can see that the influence of the horse shoe vortex is suppressed and only the leewake vortex can act onto the sediment. At $t = 60$s an additional erosion begins at the edges on the upstream side of the plate. As one can see, this erosion is growing until the plate gets underwashed. At this point the whole scouring process is initiated and one can assume a significant reduction of the stability, after some additional time.

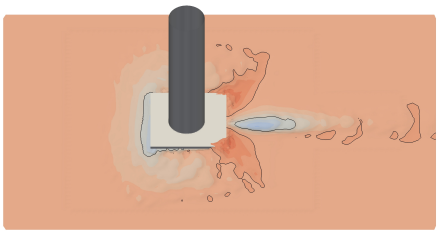
Figure 19 shows the simulation results for the single pile but without the buffer cell approach applied. The pictures clearly show the importance of such an approach. The wrong erosion in front of the pile completely suppresses the erosion directly at the pile which finally leads to totally different and wrong results.



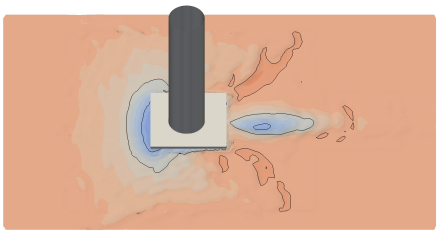
(a) $\mu_{S_{\text{lower}}} = 1 \frac{\text{Ns}}{\text{m}^2}$, $t = 20$s



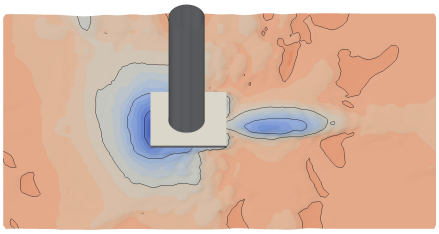
(b) $\mu_{S_{\text{lower}}} = 1 \frac{\text{Ns}}{\text{m}^2}$, $t = 60$s



(c) $\mu_{S_{\text{lower}}} = 1 \frac{\text{Ns}}{\text{m}^2}$, $t = 120$s



(d) $\mu_{S_{\text{lower}}} = 1 \frac{\text{Ns}}{\text{m}^2}$, $t = 240$s

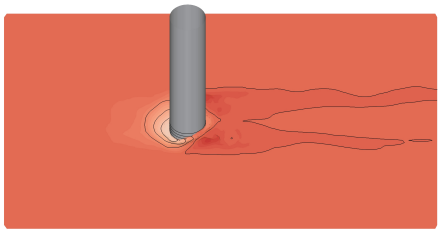


(e) $\mu_{S_{\text{lower}}} = 1 \frac{\text{Ns}}{\text{m}^2}$, $t = 500$s

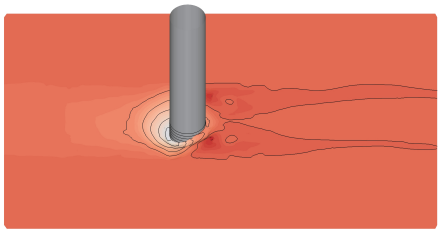
Figure 18: Scour development around a vertical pile with a mudplate in steady current

4 Conclusions

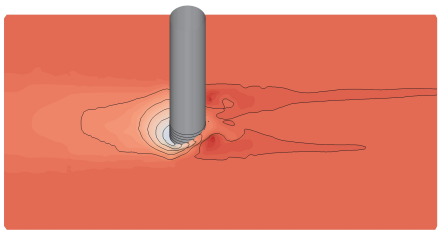
In this study a new numerical code for the simulation of scour around offshore structures has been presented. The method is based on a Bingham model for the soil and an additional model for the suspension. The sediment shape is described using the Volume-of-Fluid method. It has been shown, that the method resolves complex problems like the scour development around a circular pile in



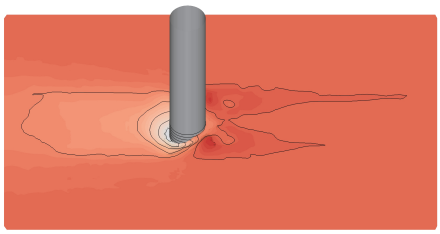
(a) $\mu_{S_{\text{lower}}} = 1\frac{\text{Ns}}{\text{m}^2}$, $t = 20$s



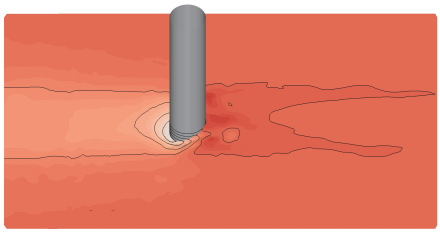
(b) $\mu_{S_{\text{lower}}} = 1\frac{\text{Ns}}{\text{m}^2}$, $t = 60$s



(c) $\mu_{S_{\text{lower}}} = 1\frac{\text{Ns}}{\text{m}^2}$, $t = 120$s



(d) $\mu_{S_{\text{lower}}} = 1\frac{\text{Ns}}{\text{m}^2}$, $t = 240$s



(e) $\mu_{S_{\text{lower}}} = 1\frac{\text{Ns}}{\text{m}^2}$, $t = 500$s

Figure 19: Vertical pile in steady current: sediment evolution without using the buffer cell approach.

current. The results are in a good agreement with experiments. Nevertheless, the treatment of the suspension requires further improvements, especially for the modeling of its generation. As the influence of the suspension is high a final calibration of the minimal Bingham viscosity can not be given at this development state.

It has been shown, that the solver is applicable to complex structures, which can not be simulated by other methods based on grid morphing. Furthermore, the calculation times are small enough to allow an industrial application. For the simulation of the scour around a vertical pile in current the solver requires approximately 25 CPU hours for 10 seconds of simulation. This is a big improvement compared to the method investigated by Nagel et al. (2020), where 6000 CPU hours are reported. However, we would like to emphasize again, that the method used by Nagel et al. (2020) is modeling the sediment in more detail.

To achieve a wall behavior at the sediment surface, while using only one velocity field for all phases, a new approach based on buffer-cells has been presented. Its application demonstrates the importance of such an method, and the wall behavior was resolved sufficient for first simulations. For future work, the authors recommend to replace this approach, by an approach based on an additional velocity field for the sediment.

To improve the wall behavior even more, the typical wall functions have been transferred successfully to the domain internal sediment surface. The results show a very good agreement with a standard smooth wall function for domain boundaries. Further improvement should be achieved using a rough wall function instead of a smooth wall function.

5 Acknowledgements

This work was supported by the *Ministerium für Energiewende, Landwirtschaft, Umwelt und ländliche Räume des Landes Schleswig-Holstein.*

References

Amoudry, L. and Liu, P.-F. (2009). Two-dimensional, two-phase granular sediment transport model with applications to scouring downstream of an apron. *Coastal Engineering*, 56(7):693 – 702.

Baykal, C., Sumer, B., Fuhrman, D., Jacobsen, N., and Fredsøe, J. (2015). Numerical investigation of flow and scour around a vertical circular cylinder. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 373.

Baykal, C., Sumer, B., Fuhrman, D., Jacobsen, N., and Fredsøe, J. (2017). Numerical simulation of scour and

backfilling processes around a circular pile in waves. *Coastal Engineering*, 122:87 – 107.

Breusers, H. (1965). Conformity and time scale in two-dimensional local scour. Technical report, Delft Hydraulics Laboratory.

Breusers, H. (1967). Time scale of two-dimensional local scour. In *12th IAHR Congress*, volume 3, pages 275–282, Ft. Collins.

Buchko, M., Kolkman, P., Pilarczyk, K., and Hydraulics, D. (1988). *Investigation of Local Scour in Cohesionless Sediments Using a Tunnel-Model*. Delft Hydraulics.

Chauchat, J., Cheng, Z., Nagel, T., Bonamy, C., and Hsu, T.-J. (2017). SedFoam-2.0: A 3-D two-phase flow numerical model for sediment transport. *Geoscientific Model Development*, 10:4367–4392.

Cheng, Z., Hsu, T.-J., and Calantoni, J. (2017). Sedfoam: A multi-dimensional eulerian two-phase model for sediment transport and its application to momentary bed failure. *Coastal Engineering*, 119:32 – 50.

Dietz, J. (1969). Kolkbildung in feinen oder elichten Sohlmaterialien bei stroemendem Abfluss. In *Mitteilungen des Theodor Rehbock Flussbaulaboratorium*, pages 1–122.

Fourtakas, G. and Rogers, B. (2016). Modelling multi-phase liquid-sediment scour and resuspension induced by rapid flows using smoothed particle hydrodynamics (sph) accelerated with a graphics processing unit (gpu). *Advances in Water Resources*, 92:186 – 199.

Graf, K., Meyer, J., Renzsch, H., and Preuß, C. (2017). Investigation of modern sailing yachts using a new free-surface RANSE code. In *Innovation in high performance sailing yachts (INNOV'SAIL 2017)*, pages 67–76, Lorient, France. Cite de la Voile Eric Taberly and Naval Academy Research Institute.

Graf, K., Renzsch, H., and Meyer, J. (2016). Prediction and optimization of aerodynamic and hydrodynamic forces and boat speed of foiling catamarans with a wing sail and a jib. In *Transactions - Society of Naval Architects and Marine Engineers*, volume 124.

Hackbusch, W. (2017). *Theorie und Numerik elliptischer Differentialgleichungen*. Springer Spektrum, Leipzig, 4 edition.

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

Huang, X. and García, M. (1997). A perturbation solution for bingham-plastic mudflows. *Journal of Hydraulic Engineering*, 123(11):986–994.

Jasak, H. (1996). *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. PhD thesis.

Jasak, H., Weller, H. G., and Gosman, A. D. (1999). High resolution NVD differencing scheme for arbitrarily unstructured meshes. *International Journal for Numerical Methods in Fluids*, 31:431–449.

Liu, K. and Mei, C. C. (1989). Effects of wave-induced friction on a muddy seabed modelled as a bingham-plastic fluid. *Journal of Coastal Research*, 5:777–789.

Manenti, S., Sibilla, S., Gallati, M., Agate, G., and Guandalini, R. (2012). SPH Simulation of Sediment Flushing Induced by a Rapid Water Flow. *Journal of Hydraulic Engineering*, 138(3):272–284.

Menter, F. R., Ferreira, J., and Esch, T. (2003). The SST Turbulence Model with Improved Wall Treatment for Heat Transfer Predictions in Gas Turbines. *International Gas Turbine Congress 2003*.

Meyer, J., Graf, K., and Slawig, T. (2017). A new adjustment-free damping method for free-surface waves in numerical simulations. In *VII International Conference on Computational Methods in Marine Engineering, MARINE 2017*, volume 2017-May, pages 296–311.

Meyer, J., Renzsch, H., Graf, K., and Slawig, T. (2016). Advanced CFD-Simulations of free-surface flows around modern sailing yachts using a newly developed OpenFOAM solver. In *THE 22nd CHESAPEAKE SAILING YACHT SYMPOSIUM*, volume 22, pages 161–177.

Nagel, T., Chauchat, J., Bonamy, C., Liu, X., Cheng, Z., and Hsu, T.-J. (2020). Three-dimensional scour simulations with a two-phase flow model. *Advances in Water Resources*, 138:103544.

Nagel, T., Julien, C., Cheng, Z., Bonamy, C., Liu, X., Hsu, T.-J., and Bertrand, O. (2017). Two-Phase Flow Simulations of Scour Around a Cylindrical Pile. In *Coastal Dynamics 2017*, pages 1758–1766, Helsinky.

Nnadi, F. N. and Wilson, K. C. (1992). Motion of Contact-Load Particles at High Shear Stress. *Journal of Hydraulic Engineering*, 38:669–679.

Queutey, P. and Visonneau, M. (2007). An interface capturing method for free-surface hydrodynamic flows. *Computers and Fluids*, 36(9):1481 – 1510.

Roulund, A., Sumer, B. M., Fredsøe, J., and Michelsen, J. (2005). Numerical and experimental investigation of flow and scour around a circular pile. *Journal of Fluid Mechanics*, 534:351–401.

Sattar, A. M., Jasak, H., and Skuric, V. (2017). Three dimensional modeling of free surface flow and sediment transport with bed deformation using automatic mesh motion. *Environmental Modelling and Software*, 97:303–317.

Stahlmann, A. (2013). *Experimental and Numerical Modeling of Scour at Offshore Wind Turbines*. PhD thesis, University Hannover.

Sumer, B. (2007). Mathematical modelling of scour: A review. *Journal of Hydraulic Research*, 45(6):723–735.

Sumer, B., Whitehouse, R. J., and Tørum, A. (2001). Scour around coastal structures: a summary of recent research. *Coastal Engineering*, 44(2):153 – 190.

Sumer, B. M., Christiansen, N., and Fredsøe, J. (1997). The horseshoe vortex and vortex shedding around a vertical wall-mounted cylinder exposed to waves. *Journal of Fluid Mechanics*, 332:41–70.

Sumer, B. M., Petersen, T. U., Locatelli, L., Fredsøe, J., Musumeci, R. E., and Foti, E. (2012). Backfilling of a scour hole around a pile in waves and current. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 139:9–23.

Ulrich, C., Leonardi, M., and Rung, T. (2013). Multi-physics sph simulation of complex marine-engineering hydrodynamic problems. *Ocean Engineering*, 64:109 – 121.

Voelkner, S., Wriggers, W., Luo-Theilen, X., and Rung, T. (2015). An Overset-Grid Three-Phase Flow Model For Offshore Operations. In *VI International Conference on Computational Methods in Marine Engineering MARINE 2015*, pages 943–954, Rome, Italy.

Wackers, J., Koren, B., Raven, H. C., van der Ploeg, A., Starke, A. R., Deng, G. B., Queutey, P., Visonneau, M., Hino, T., and Ohashi, K. (2011). Free-Surface Viscous Flow Solution Methods for Ship Hydrodynamics. *Archives of Computational Methods in Engineering*, 18:1–41.

D. Danksagung

Zu Beginn möchte ich ganz besonders Prof. Dr. Kai Graf und Prof. Dr. Thomas Slawig für die Betreuung, Begleitung und vor allem Ermöglichung dieser Doktorarbeit danken.

Während meines Studiums konnte mich Kai für die numerische Simulation begeistern. Mit seiner Vorlesung “Programming of Numerical Methods” kam ich zum ersten Mal in Kontakt mit diesem Fachgebiet. Kai bot mir immer sehr vielseitige und interessante Themen und insbesondere sein Engagement hat diese Arbeit initiiert.

Durch die vielen persönlichen Gespräche mit Thomas konnte ich meine Kenntnisse über das wissenschaftliche Arbeiten erweitern. Thomas Beratung hat mir sehr geholfen diese Arbeit anzufertigen. Auch seine Gestaltung des Oberseminars sowie der Beitrag der Teilnehmenden empfand ich als sehr hilfreich.

Auch Prof. Dr. Michael Hinze gilt besonderer Dank für die Übernahme der Zweitkorrektur dieser Arbeit.

Außerdem bedanke ich mich bei Hannes Renzsch, Christoph Böhm und Hannes Pegel. Während des Einarbeitens in OpenFOAM war Hannes Renzsch eine große Unterstützung. Zudem führten wir viele spannende fachliche Diskussionen über Simulationsmethoden für freie Wasseroberflächen, sodass sein fachlicher Input in den ersten Jahren sehr richtungsweisend war. Mit Christoph Böhm und Hannes Pegel führe ich bereits seit Beendigung meines Master-Studiums gerne Konversationen über CFD-Simulationen, welche mir insbesondere bei der Segelyacht-Simulation dienlich sind.

Zum Schluss bedanke ich mich herzlich bei meiner Familie. Unser familiärer Zusammenhalt bedeutet mir sehr viel. Meine Eltern haben mich immer unterstützt und mir in allen Lebenslagen den nötigen Halt gegeben. Auch über das exzellente Verhältnis zu meinem Bruder Moritz bin ich sehr dankbar. Moritz berät mich seit einigen Jahren bei der selbständigen Abwicklung kommerzieller Projekte unter Anwendung des hier vorgestellten Verfahrens.

E. Eidesstattliche Erklärung

Hiermit versichere ich eidesstattlich,

i. dass diese Arbeit, abgesehen von der Beratung durch die Betreuer Thomas Slawig und Kai Graf, nach Inhalt und Form meine eigene ist, außer die Abschnitte *Motion*, *Time Integration*, *Stability*, *External Forces*, *Immersed Transom Study* sowie das Kapitel *APPLICATION* der in Anhang A beigefügten Veröffentlichung “Advanced CFD-Simulations of free-surface flows around modern sailing yachts using a newly developed OpenFOAM solver”,

ii. dass diese Arbeit bereits zum Teil an anderen Stellen publiziert wurde, und zwar:

   a) Janek Meyer, Hannes Renzsch u. a. (März 2016). “Advanced CFD-Simulations of free-surface flows around modern sailing yachts using a newly developed OpenFOAM solver”. In: *The Twenty-Second Chesapeake Sailing Yacht Symposium* (Annapolis, Maryland, USA). Hrsg. von Britton R. Ward. Society of Naval und Marine Engineers, S. 161–177,

   b) Janek Meyer, Kai Graf und Thomas Slawig (Mai 2017). “A new adjustment-free damping method for free-surface waves in numerical simulations”. In: *MARINE VII : proceedings of the VII International Conference on Computational Methods in Marine Engineering* (Nantes, France). Hrsg. von Michel Visonneau, Patrick Queutey und David Le Touzé. CIMNE, S. 296–311. ISBN: 978-84-946909-8-3,

   c) Janek Meyer, Kai Graf und Thomas Slawig (4. Mai 2021). *Simulation of scour around arbitrary offshore foundations based on the Volume-of-Fluid method combined with a Bingham model*. Version 2. arXiv: arXiv:2012.03051v2,

iii. dass diese Arbeit unter Einhaltung der Regeln guter wissenschaftlicher Praxis der Deutschen Forschungsgemeinschaft entstanden ist

iv. und dass mir noch kein akademischer Grad entzogen wurde.

Janek Meyer, 2021