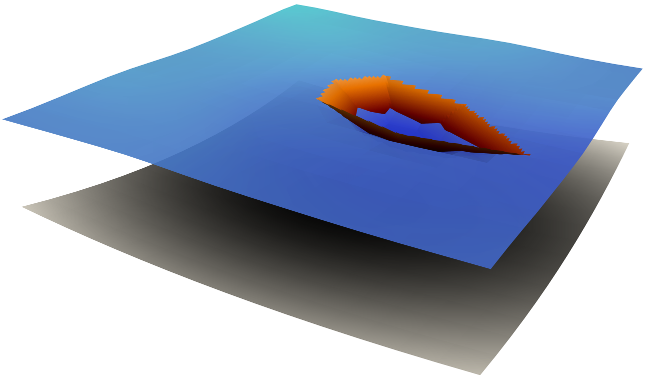
UNIVERSITY OF TRENTO

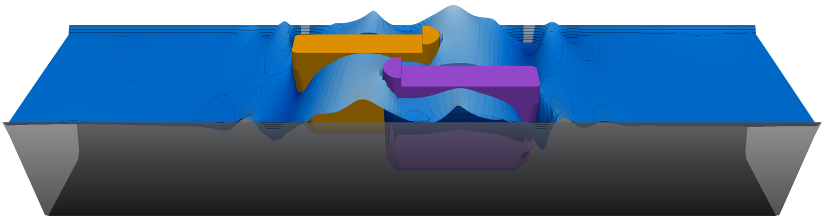
Doctoral School in Civil, Environmental and
Mechanical Engineering

Curriculum 3. Modelling and Simulation

Cristian Brutto

Development of an efficient fluid-structure interaction model for floating objects





2024 - Doctoral thesis

University of Trento

DEPARTMENT OF CIVIL, ENVIRONMENTAL AND MECHANICAL ENGINEERING

DOCTORAL PROGRAMME IN CIVIL, ENVIRONMENTAL AND MECHANICAL ENGINEERING

~ · ~

Academic Year 2022–2023

Development of an efficient fluid-structure interaction model for floating objects

Supervisor
Prof. Michael Dumbser

Graduate Student
Cristian Brutto
224393

Final examination date: 18-06-2024

University of Trento,
Trento, Italy,

Doctoral thesis in
Civil, Environmental, and Mechanical Engineering,
Cycle XXXVI

Departement of Civil, Environmental, and Mechanical Engineering
University of Trento
Academic year 2022/2023

Graduate Student	**Cristian Brutto**
Supervisor	**Prof. Dr.-Ing. Michael Dumbser**
Defended on	*June 18th 2024*

Members of the Examination Committee :

Prof. Dr. Eleuterio F. Toro

Dr. Mario Ricchiuto

Prof. Dr. Harry B. Bingham

Dr.-Ing. Jacek A. Jankowski

Prof. Dr. Elena Gaburro

Preface

This thesis has been developed during the three-year Doctoral program of the Doctoral School in Environmental Engineering at the Department of Civil, Environmental and Mechanical (DICAM) Engineering of the University of Trento.

The presented research was co-funded by the Italian Ministry of Education, University and Research (MIUR) in the frame of the Departments of Excellence Initiative 2018–2027 attributed to DICAM of the University of Trento (grant L. 232/2016) and in the frame of the PRIN 2017 project *Innovative numerical methods for evolutionary partial differential equations and applications.* Additionally, this research was co-funded by the Bundesanstalt für Wasserbau (BAW) in the framework of the R&D contract n°739530400019002 between the University of Trento and the BAW.

Most of the work was carried out in Trento under the supervision of Prof. Dr.-Ing. Michael Dumbser. I spent two months (January 2023 - March 2023) at INRIA Bordeaux in visit to the CARDAMOM team of Dr. Mario Ricchiuto, and two months (September 2023 - October 2023) at the BAW offices in Karlsruhe working alongside the W4 Schifffahrt (Shipping) sub-department of Dr.-Ing. Michael Schröder.

Trento, June 2024

Cristian Brutto

Acknowledgements

It was a dream of mine to create something new and give my contribution to the research community, and I am happy it has become a reality. Many people accompanied me during these years, but it is fair to make a few honourable mentions.

The first person I would like to thank is my supervisor Michael Dumbser, who made possible this experience. His support and guidance were decisive for the success of this PhD and taught me to keep learning, trying different paths and aiming high.

The second pillar of this project is the BAW, represented here by Jacek A. Jankowski and Michael Schröder. No words can express how having someone so interested in using the product of my work was a motivation to polish the results and pay attention to details. Even if the physical distance was an obstacle, we made the collaboration work through frequent and long email exchanges. It was a pleasure to be at your offices in Karlsruhe and meet the W4 team.

The visit to Inria Bordeaux, in addition to giving me a fresh perspective and enriching my understanding of the topic, inspired many of the solutions present in this manuscript. I then express my gratitude to Mario Ricchiuto, Martin Parisot, Elena Gaburro and the entire CARDAMOM team for welcoming me and making the visit memorable.

To all the colleagues I encountered at the university, during workshops and conferences. It was a joy to get to know many motivated people from different backgrounds. In particular, thanks to Vincenzo Casulli for the fruitful conversations we had.

Thanks to the present and past members of my research group and its visitors: Firas, Alessia, Olindo, Enrico, Davide, Laura, Ilya, Francesco, Ondrej, Simone, Walter, Andrea, Saray, Maurizio. I am glad for the lunches and trips in your company, they have been precious moments that made me know you beyond the professional level.

Finally, thanks to my family and friends for caring about me.

Contents

Abstract

This thesis gives an overview of the process that led to the development of a novel semi-implicit fluid-structure interaction model. The thesis is dedicated to the creation of a new numerical model that allows to study ship generated waves and ship manoeuvers in waterways for various vessel characteristics and speeds in different external current situations. A model like this requires a coupling between the fluid and the solid to generate the waves and the hydrodynamic forces on the hull.

Since the horizontal dimensions are significantly larger than the vertical dimension, we started by employing the shallow water equations, which are based on the assumption of hydrostatic pressure. The discretization was carried out taking only the nonlinear advective terms explicitly while the pressure terms are discretized implicitly, which makes the CFL condition milder. The price to pay for this semi-implicit discretization is an increase in the algorithm complexity compared to a fully-explicit method, but it is still much simpler than a fully-implicit discretization of the governing equations. Indeed, the mass and momentum equations couple, and finding the unknowns involves solving a system of equations with dimensions equal to the number of cells. The grid supporting the discretization is staggered, overlapping and Cartesian. Since the aimed application domain is inland waterways, it is paramount to allow wetting and drying of the cells. This was achieved by acting on the depth function, the relationship between the free-surface elevation and the water depth in the cell.

The main novelty of this research project is the two-way coupling of the PDE system for the water flow with the ODE system for the rigid body motion of the ship. The hull defines the ship region, and its shape can range from a simple box to an STL file of a real 3D ship geometry. Where the hull is in contact with the water, the cells are pressurized. This pressurized group of cells generates waves as it moves, and its motion is influenced by incoming external waves. This result is obtained by imposing an upper bound to the depth function, so that the water depth does not increase when it reaches the hull elevation, while the pressure is allowed to increase. This upper bound increases the nonlinearity of the system, which may have dry cells, wet free-surface cells and pressurized cells. The solution of this system is found by a single nested-Newton iterative solver of Casulli and Zanolli [36], in which with two separate linearizations the system is written in a sparse, symmetric, positive semi-definite form. This particular form allows us to employ a matrix-free conjugate gradient method, and efficiently get the unknown pressure. The integral of the pressure over the hull is applied for the hydrodynamic force and torque acting on the ship.

After adding the skin friction and other external forces from the propeller or the rudder, the total force is inserted in the equation of motion of the rigid body. The ODE system is discretized with a second-order Taylor method, and it is solved for the six degrees of freedom (3 coordinates for the position vector of the barycenter and 3 rotation angles), providing the next position and orientation of the ship.

The vertical translation of the rigid body is governed by the gravitational force and the restoring force from Archimedes' principle. As the ship oscillates up and down, the gravitational potential energy is partially transferred to the radiated free-surface water waves, damping and eventually stopping the motion. Also, the ship pushes and pulls the water around it, inducing the added mass force. All these elements constitute the ODE that was used for the verification of the vertical degree of freedom. The numerical simulation gave the expected results for the vertical motion.

The horizontal translation, important for the manoeuvers, presented a numerical instability unseen in our previous test cases, which is connected to the relative motion between the ship and the grid. In each time step in which the ship enters a new cell, the pressure sharply increases and decreases at the ship bow. An oscillation can build up in time and create an unphysical void below the vessel. We implemented a few ideas to attenuate the oscillations. At the heart of all the following techniques is the reduction of the time derivative of the water depth, especially for those cells transitioning to a pressurized state. All these modifications were effective at controlling the oscillations, each with a different intensity, and simulations with a horizontal motion are much more stable than without these techniques.

With the collaboration of the BAW research institute, we worked on the model validation. We used data from two separate experiments to compare the measurements with the numerical results. Specifically, we focused on the ship-generated wave height and the hydrodynamic forces on the hull. The comparison is satisfactory for the wave height. The force and torque prediction is plausible but underestimated compared to the measurements. The model seems to displace the water volume correctly during the ship passage, while the force and torque response might need additional work to be trusted in applications.

Even though the hydrostatic assumption is mostly correct in our range of applications, the presence and the motion of a ship could generate strong vertical accelerations of the flow, which may not be negligible. For this reason, we implemented an algorithm that corrects the velocity field, introducing also dispersive effects due to a non-hydrostatic pressure. The correction consists of a higher-order Bousinnesq-type term in the momentum equation and the solution

of the resulting system. The non-hydrostatic update has a small influence on the wave generation, while it alters significantly the reaction forces.
The subgrid method implementation allowed to benefit from high-resolution bottom descriptions while keeping the grid size coarse. The same subgrid can also be used for a refined definition of the hull, which makes the volume computations more accurate. Furthermore, the subgrid introduces new possible states for the cells, as they can be partially dry or partially pressurized. These intermediate states translate into smoother transitions from one state to the other when the free-surface is close to the bathymetry or to the hull.
Concerning the software implementation of the developed scheme, in order to improve the execution performance of the prototype script formulated initially in Matlab, the numerical method was rewritten as a Fortran program. Also, thanks to the domain decomposition technique and the MPI standard, each simulation can run in parallel on multiple CPUs, leveraging the computational power of supercomputers.
The coupling of the PDE and ODE system, together with an appropriate redefinition of the depth function, proved to be a valuable method for studying fluid-structure interaction problems. The combination of efficient numerical techniques led to the development of a tool with a potential to be applied in the practice for the simulation of floating objects in wide domains.

1 Introduction

1.1 Motivations for this study

In waterways engineering there is a growing need to improve the methodology with which the passage of ships in rivers affects the neighbouring structures. Indeed, the shipping industry pushes for the adoption of bigger means of transport, which would increase the height of the ship-generated waves and the ship sinkage as it travels. This is especially true in shallow water environments, typical for inland waterways. In alternative, where the physical constraints of the waterway restrict the size of the ships, the traffic increases. The horizontal motion of these floating bodies is an aspect of great importance because they are massive objects. While traveling through a body of water, a ship generates wave systems which have potential impacts on the surroundings. In the engineering practice, these impacts are particularly relevant in confined waterways and coastal environments, where the ship-generated waves interacts with other natural or man-made structures. For example, we mention the destruction of groynes, the erosion of river banks and shorelines, the increased sediment transport, the damage to mooring systems, the alteration of the ecosystems.

The altered wave and flow field around the ship is also significant for the ship itself. First and foremost, the water exerts a pressure on the hull that is responsible for the floating condition and for a resistance that has to be countered by the propeller. As on the two sides of a coin, exactly the same pressure causes the energy input in the waves formation. In addition, the depressed water-level profile around the ship lowers its equilibrium position, causing a vertical displacement called sinkage and a rotation called trim. The sum of the sinkage and trim effects forms the squat [42, 117], which reduces the under-keel clearance and is thus an important navigation parameter to avoid grounding. When the vessel moves near and parallel to a structure, such as a river bank, the local acceleration of the flow between them causes a yaw torque that attracts the ship towards the bank. Finally, the shear stress between the water and the hull generates a resistance that contrasts the forward motion of the boat. This resistance is stronger in confined waterways.

A detailed description of these and more loads is given in the review made by Dempwolff et al. [47] and in the rich literature mentioned there.

1.2 Ship-generated wave systems

Given the importance of ship-generated waves, it is worthwhile to provide a general description of the typical pattern of these waves. A ship moving relatively to the water displaces a volume of water which is proportional to the size and the speed of the vessel. At the bow, the water is pushed away and builds up at the front of the ship, while the space left at the stern is filled by water. In these two regions two stagnation points determine the height of the bow wave and stern wave. The difference in free-surface elevation induces a pressure gradient which creates a so-called return current that goes opposite to the ship velocity and flows in the region at the sides of the boat. The intensity of the return current depends on the dynamics of the vessel but also on the relative dimensions of the ship and the waterway cross section. The energy balance requires that this velocity-field alteration in the lateral region is compensated by a lowering of the free-surface elevation, which is called drawdown. Together, the bow wave, stern wave and drawdown form the primary wave system, which mainly affects the area close to the ship. In confined waterways and shallow water, the primary wave system is characterized by a higher wave amplitude. Additionally to the primary wave system, a secondary wave system forms at the bow and at the stern, where the cross section of the hull changes rapidly along the main axis. These waves are short and travel far from the origin. Finally, if the ship is accelerating or moving close to the critical speed, long-period solitons travel at the front of the ship. For more detail on the wave systems we refer to the report of BAW [120] and the article of Dempwolff et al. [47].

1.3 Solutions for this type of FSI problems

In the engineering practice, different tools complement each other to design appropriate interventions:

- analytical solutions make clear what are the important variables at play, but they work under restrictive conditions.

- laboratory or field experiments allow us to consider most of the forces and circumstances, but they are expensive and the data is measured only at specific points, times and setups, making it difficult to generalize the results for different conditions.

- numerical simulations offer great flexibility in the definition of the bathymetry, the initial conditions and the vessel characteristics, but the accuracy they provide is often linked to the time required to make the computations.

This thesis is focused on the numerical methods. We mention here a few examples of numerical methods that could help in studying the ship-waterway interaction, without the pretense of making a complete list.

The range of solutions available is broad, and usually the choice of the numerical implementation is a trade-off between accuracy, speed of computation, and complexity. The most expensive but most accurate models are those that solve the Navier-Stokes equations directly or in the Reynolds-averaged form [56]. These models are widely used to compute ship-focused variables, but they become really demanding at mesoscale.

Bradford [19] proposed a model in which the structure is the void resulting from the vertical subdivision of the domain in two regions of water, and each region is discretized by a σ-coordinate transformation of the grid. In the context of potential-flow theory, Shao et al. [104] include a floating body using a higher-order boundary element method, while Tong et al. [114] implement an immersed boundary method combined with a harmonic polynomial cell method. Banks et al. [6–8] developed a fluid-structure-interaction (FSI) partitioned scheme and put the emphasis on analyzing the stability related to added-mass and added-damping effects. Ferrari and Dumbser [55] developed a semi-implicit finite volume scheme for the free-surface equations written in a conservative form that treats the non-hydrostatic pressure exerted against a fixed rigid body; in particular they use the diffuse interface approach, in which for each cell along the vertical the volume is limited and subdivided in the liquid, solid and void phases, while the pressure is unbounded. Similar diffuse interface models in the context of weakly compressible flows have been forwarded, for example the works of Gaburro et al. [58], Kemm et.al [73] and literature citations therein.

Numerical methods for pressurized flows are often derived for stormwater systems, in which the flow transits from a free-surface condition to a pressurized regime, where it is then called mixed flow. The numerical solutions for this type of problem differ in how many sets of equations are adopted, either one working in both free-surface and pressurized conditions, or two. Some examples of these schemes are the Preissmann slot technique [45, 95] and the two-component pressure approach [119]; for more details, see the review by Bousso et al. [18].

Inspiring works can also be found in problems related to the aerodynamics theory. The connection to the shallow-water theory is made by the aerodynamic equivalence principle [117], which recognizes the similarity in the governing equations. Indeed, the shallow water flows are equivalent to certain compressible gas flows. Caiden et al. [27] make two phases interact by separating the domain in compressible and incompressible regions. They use the level-set method to track the interface and the Ghost Fluid Method (GFM) to impose the proper boundary conditions at the interface. In the same spirit, throughout this text

the unbounded part of the domain is also called the compressible region, while the incompressible region indicates the part of the domain where the ship forces the free-surface position.
In this thesis the floating body is moved also in the horizontal directions, which means that the compressible and incompressible parts of the domain vary in time. This change can result in undesired pressure oscillations. This problem is already known in the application of the immersed-boundary method (IBM), and there are works that aim to smooth the transition of the cells from one kind of domain to another, such as the ones of Pan and Shen [88], Liao et al. [82], Seo and Mittal [103], Luo et al. [83] and Kontos [75].

1.4 Dispersion effects

One very important element to consider is if dispersion effects are required, or not, and if the region near the floating body needs to be treated differently from the rest of the domain. Bosi et al. [17] show that when there are no rotational degrees of freedom and under the assumptions of the Boussinesq theory, high-order dispersive terms become negligible and therefore a simple hydrostatic shallow water model is appropriate in the body region. Work in this direction has been also done by Lannes [78] and Godlewski et al. [63, 64], who coupled a shallow water model with the motion of a rigid body. Parisot [90], while studying congested flows, also introduced the air-pocket dynamics. Bocchi et al. [15] used the congested-flow theory to simulate a wave energy converter (WEC) [60], where the structure is fixed.
The waves generated by the fluid-stucture interaction can travel far from the body and may also be dispersive in nature; consider, for example, precursor solitons or the Kelvin wake [106]. In these cases, it is advisable to introduce dispersion effects outside the near-body region to take into account non-hydrostatic effects as well. This can be done either by employing a multi-layer non-hydrostatic model [39, 107], or Boussinesq-type models [84–86]. It is interesting to point out that there is a link between the nonhydrostatic models in multi-layer form and in Boussinesq form, as found by Bai and Cheung [5]. An example of multi-layer model can be found in the work of Rijnsdorp and Zijlema [98]: in their extension of the non-hydrostatic SWASH model [123], they integrate a body with a fixed position into the domain. In a recent work, Rijnsdorp et al. [97] use SWASH to simulate moored floating structures. The non-hydrostatic pressure is included thanks to the use of the Keller-box scheme, as shown by Stelling and Zijlema [108]; only a few vertical layers are required to get an accurate propagation of short waves. Examples of application of the Boussinesq-type approximation can be found in the works of Bingham

[13], Karambas and Loukogeorgaki [70], Beck and Lannes [10], Filippini et al. [57]. A high order discontinuous Galerkin finite element method for the simulation of the coupling of dispersive water waves with structures can be found in the work of Engsig-Karup et al. [51]. For recent and very efficient hyperbolic reformulations of nonlinear dispersive shallow water flows, see e.g. Bassi et al. [54], Busto et al. [52], Escalante et al. [53], Escalante and Morales [9], Favrie and Gavrilyuk [25] and references therein.

1.5 The SIFSI model

In the inland waterways, a floating object interacts with - and moves in - a body of water which covers a relatively larger area. We can imagine a ship with a beam of ten meters and a length of a hundred meters sailing in a river with a width of a few hundred meters and a length of tens of kilometers. Figure 1.1 illustrates a real-world example of the typical scales.

If the main goal is to capture phenomena in the ship surroundings, numerical simulations require small computational cells; consequently, the large domain needs many cells and the excessive computational time becomes the main challenge. For this reason, it is imperative to use efficient tools to solve the problem. This principle, together with the goal of creating an instrument suitable for real-world applications, guided many decisions during the development of the model, starting from the choice of the foundational model.

The method used in this thesis for the pressurized part of the flow was developed by Casulli and Stelling [41]. The main idea is to write the water depth in the mass conservation equation as a piecewise linear function limited from both below and above. The limit below could represent the bathymetry of a river bed or the bottom of a pipe, while the limit on top may be a floating structure or the roof of a pipe. With this boundary from above, when the flow is pressurized, the wet cross section is limited while the pressure can increase. Other works that laid the foundation for this thesis are those of Casulli and Cheng [38], Casulli and Walters [34], and Casulli and Stelling [40] on staggered semi-implicit finite volume / finite difference schemes for free surface flows. The main contributions of this thesis are the extension of these type of semi-implicit methods to work with floating objects and the development of numerical techniques to make them stable, which is then supported by the verification of the vertical motion and the validation of the ship-induced waves and hydrodynamic forces. For recent results on alternative staggered semi-implicit finite volume and finite element schemes for the shallow water and Navier-Stokes equations, the reader is referred to the works of Busto and Dumbser [111], Busto et al. [26], Ioratti and Dumbser [68], Del Río et al. [24], Tavelli and Dumbser [99].

Figure 1.1. Traffic of cargo ships in the Rhine river. Image courtesy of BAW.

In the new method proposed in this thesis the hydrodynamics is solved in terms of conservative variables, in order to deal also with bores and hydraulic jumps. As starting point, we assume a hydrostatic-pressure distribution in water, and in this context we develop a new and efficient semi-implicit finite volume model for fluid-structure interaction problems (SIFSI). The model originates from the shallow water equations written in a conservative form, and the water flow is coupled with the equations of motion of a rigid floating body. The resulting sparse mildly nonlinear system for free-surface and pressurized flows is linearized by the nested Newton method of Casulli et al. [21, 22, 35, 36], and it is then solved by a matrix-free conjugate gradient method since the resulting pressure systems are symmetric and positive definite. Mass is exactly conserved at each time step, even when cells in the computational domain are undergoing wetting and drying. The floating body, which we will refer to also as the "ship", "boat" or "vessel", has six degrees of freedom, three for its translational and three for its rotational degrees of freedom. The forces acting on it are those resulting from gravity, from the hydrostatic pressure pushing on its external surface and the resistance due to the shear stress. As a result, the water waves affect the movement of the ship, but the equation coupling works both ways, meaning that the ship changes the water depth in the space below it, generating radiated waves as it moves. The grid supporting the discretization is staggered, meaning that some variables are defined in the cell center, while others are defined on the cell interfaces. This setup is convenient in a finite-volume setup since the variables affecting the fluxes are right where they are needed. The grid is stationary so that the motion of the ship does not impose to recompute a new mesh. Finally, the grid is Cartesian, which has the benefit of maintaining the second-order space accuracy for central terms, like the pressure gradient. By the introduction of the subgrid technology [37] we can efficiently integrate the details of the bathymetry and of the floating body, improving the accuracy of volume computations at a reasonable computational cost. In addition, we make use of the theta method to reduce numerical dissipation and thus preserve wave height.

The foundations of the model just mentioned have already been published on an international journal by Brutto and Dumbser [23]. This article contains a description of the mathematical model and the verification of the vertical motion of the floating body. From this point, the work proceeded by addressing some important missing aspects as the dispersive effects in the water flow, the horizontal motion of the ship and the validation of waves and hydrodynamic forces.

The introduction of dispersion effects is done by means of a Boussinesq-type term in the momentum equation. At each time iteration, the system is first solved in its hydrostatic form, and the pressure and velocities updated. Then, a Poisson-type equation is solved to update the velocities and include dispersion effects.

Similarly to the models based on the immersed-boundary method, the horizontal motion of the ship can cause instabilities of the pressure, especially at the ship bow. The subgrid was already useful to dampen the oscillations, however more solutions were devised: with an alternative volumes computation algorithm, the concentration of a problematic term in specific time iterations was removed. The transition between the hyperbolic and elliptic parts of the domain was smoothed out by acting on the depth function. The impact of vertical walls was reduced by rounding the hull shape. The displaced volume was shifted from the origin cell to another destination cell. A dissipation term was introduced in the system, leading the convergence of the pressure system to a smoother solution.

Thanks to the collaboration with the BAW institute, the numerical results of the SIFSI model were compared to experimental results. First, the ship-shape definition has been improved using as input a STL file of a reference ship. Then, the setup of the field experiments has been recreated using the provided bathymetric data, the ship configurations and probe position. The comparison focused on the ship-induced waves height, the squat of the vessel, the return-flow speed and the hydrodynamic forces on the hull.

These types of models deliver results in an computationally effective way without sacrificing accuracy, where it matters. They could be of interest to the shipbuilding industry, because they allow to observe the response of the floating body in different conditions, such as in open and confined shallow waters [43]. They allow to perform simulations for multiple hull types and to study their interaction with the boundaries [118]. When they are applied to predict the response in working conditions and with complex bathymetries, such as during manoeuvres [76] in inland waterways or harbours, they can assess the generation of waves and the interplay with other bodies and structures.

1.6 Structure of the thesis

The thesis is organized as follows: in Chapter 2, the equations are written for a one-dimensional horizontal domain, and the components of the method are added progressively in order to ease comprehension and to offer a guideline for readers who would like to implement this method. Then, the scheme is generalized by adding the vertical component, and ultimately the missing transversal component, obtaining a three-dimensional hydrostatic fluid-structure-interaction (FSI) model. In Chapter 3, the model is tested in different scenarios and the obtained numerical results are compared to available exact solutions to verify the water flow and the movement of the floating body, and it is compared to field and laboratory measurements to validate the ship-generated waves, the squat, the return flow and the hydrodynamics forces. In Chapter 4, we summarize and comment on the characteristics of the method, and we outline some possible improvements, which could be the subject of future work.

2 Numerical methods

2.1 Staggered Cartesian mesh

The method presented in this manuscript is a semi-implicit finite volume scheme on a *staggered grid*: in the most general 3D case the physical domain $\Omega \in \mathbb{R}^3$ is discretized by a set of cells $\Omega_{i,j,k} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] \times [z_{k-\frac{1}{2}}, z_{k+\frac{1}{2}}]$ forming a Cartesian grid of mesh spacings $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$, $\Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}$, $\Delta z_k = z_{k+\frac{1}{2}} - z_{k-\frac{1}{2}}$ and barycenter coordinates $x_i = \frac{1}{2}(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}})$, $y_j = \frac{1}{2}(y_{j-\frac{1}{2}} + y_{j+\frac{1}{2}})$, $z_k = \frac{1}{2}(z_{k-\frac{1}{2}} + z_{k+\frac{1}{2}})$, respectively. Throughout this thesis the axes x, y, z and the indices for the variables i, j, k will follow the orientations depicted in Figure 2.1, where north, south, east, west have nothing to do with the actual orientation in the physical space, but they are just labels for the faces to refer to them unequivocally. For the 1D model we consider a domain formed by I_{max} cells along the x axis, for the 2D$_{xz}$ model we have an additional subdivision in up to K_{max} cells along the vertical axis z, and for the 3D we add the horizontal subdivision in $J_{\max}$ cells along the y axis. In each cell some variables are defined in the centre and they represent the average value $a_{i,j,k}$ of a generic quantity $a(x, y, z)$

$$a_{i,j,k} = \frac{1}{|\Omega_{i,j,k}|} \int_{\Omega_{i,j,k}} a(x, y, z)\, dxdydz, \qquad |\Omega_{i,j,k}| = \int_{\Omega_{i,j,k}} dxdydz,$$

where $|\Omega_{i,j,k}|$ is the volume of cell $\Omega_{i,j,k}$. In a staggered grid some variables are not defined in the cell centre, for example the velocities u, v and w, placed at

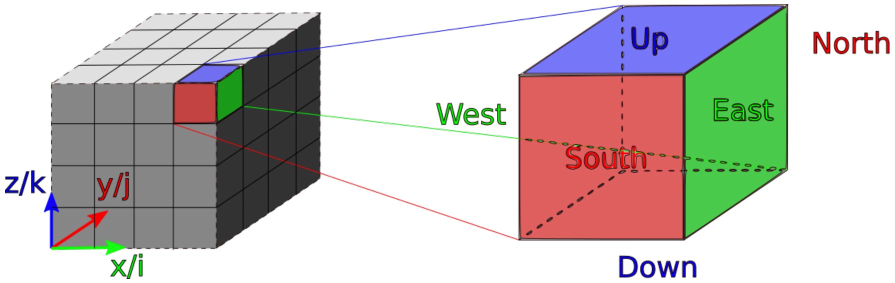


Figure 2.1. Representation of the 3D computational domain and faces orientation

the cell interfaces. Velocity should be interpreted as *average velocity* as

$$u_{i+\frac{1}{2},j,k} = \frac{1}{\Delta x_{i+\frac{1}{2}} \Delta y_j \Delta z_k} \int_{x_i}^{x_{i+1}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} u(x,y,z)\, dxdydz,$$

$$v_{i,j+\frac{1}{2},k} = \frac{1}{\Delta x_i \Delta y_{j+\frac{1}{2}} \Delta z_k} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_j}^{y_{j+1}} \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} v(x,y,z)\, dxdydz,$$

$$w_{i,j,k+\frac{1}{2}} = \frac{1}{\Delta x_i \Delta y_j \Delta z_{k+\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int_{z_k}^{z_{k+1}} w(x,y,z)\, dxdydz,$$

where

$$\Delta x_{i+\frac{1}{2}} = \frac{1}{2}\left(\Delta x_i + \Delta x_{i+1}\right),$$

$$\Delta y_{j+\frac{1}{2}} = \frac{1}{2}\left(\Delta y_j + \Delta y_{j+1}\right),$$

$$\Delta z_{k+\frac{1}{2}} = \frac{1}{2}\left(\Delta z_k + \Delta z_{k+1}\right).$$

2.2 Subgrid

There is an important trade-off for numerical methods between accuracy and the time required for the computations. The introduction of a subgrid by Casulli in [37] is an innovative method to improve computational performance substantially, while still maintaining very high accuracy. The main idea is to maintain the system size relatively small without compromising the accuracy of the computation of the water volume. To do so, two different grids are employed, one with coarse and one with finer resolution. The original mesh is where the free surface elevation $\eta = \eta(x, y, t)$ is defined; the latter (the subgrid) is used to hold the input data of the bottom elevation $b = b(x, y)$ and to compute the total water volume in each computational cell, as well as the face-averaged water depths $H = H(x, y, t)$. This separation allows for a faster algorithm because the operations on the subgrid require a lower computational effort and because the CFL condition acts on the coarser grid; meanwhile, volume accuracy is radically improved thanks to the subgrid, see Fig. 2.2 for a sketch in 1D. In wet regions, the local water depth H, the bottom elevation b and the free surface elevation η are related via the linear relation $\eta = H + b$. In general, including dry areas but in the absence of floating objects, we have the nonlinear relation $H = \max(0, \eta - b) \geq 0$. For the sake of simplicity we present the concept in one space dimension only, and for the moment without

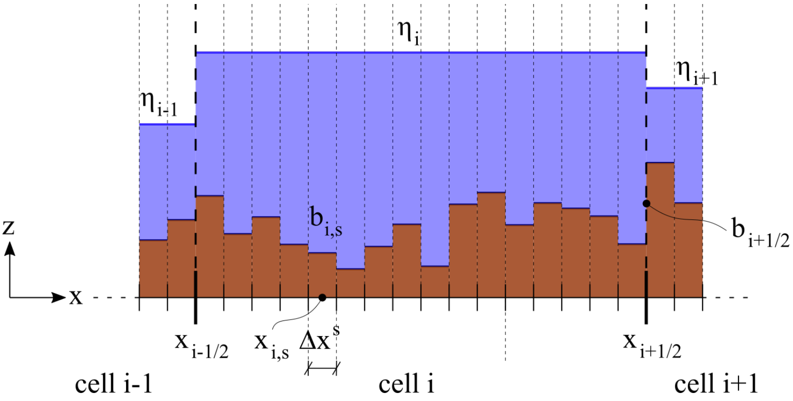


Figure 2.2. Cell divided in subcells

the floating objects, which will be introduced later. We call the subgrid cells size $\Delta x_s = \Delta x / s$, where s is the number of subcells in each cell. Potentially, s could vary spatially, but it is assumed to be constant since there is no need to make distinctions at this point. The average water depth in a cell is computed as a sum of the water depth in each subcell:

$$H(\eta_i) = \frac{1}{\Delta x} \sum_s \max(0, \eta_i - b_{i,s}) \cdot \Delta x_s = \frac{1}{s} \sum_s \max(0, \eta_i - b_{i,s}), \tag{2.1}$$

since Δx_s is assumed to be constant. Since the grid is uniform, this depth function is closely related to the corresponding volume function through $V(\eta_i) = \Delta x \Delta y H(\eta_i)$. The free surface elevation is assumed to be piecewise constant per cell, so the subcell volumes in the summation differ only if the bottom elevation varies spatially at the subgrid level. The derivative of the average depth function is given by

$$H'(\eta_i) = \frac{1}{s} \sum_s H'_s(\eta_i, s), \qquad \text{with} \qquad H'_s(\eta_i, s) = \begin{cases} 1 & \text{if } \eta_i > b_{i,s} \\ 0 & \text{if } \eta_i \leq b_{i,s} \end{cases} \tag{2.2}$$

A visual representation of the improvements on volume computation is provided in Figure 2.3. In this example, a sloped bottom is discretized first on a 2 cells grid, and then on a 2 cells grid linked to a $s = 13$ subgrid. The free surface elevation is always the same. The subgrid substantially improves over the simple piecewise constant approximation of the bottom. Also, in the case without subgrid the cells are regarded either as dry or wet, while in the grid-subgrid case cells can be wet, dry or partially wet.

It is reasonable to spend some words on the similarities and differences between the use of a fine grid without subgrid and a coarse grid connected to an equally

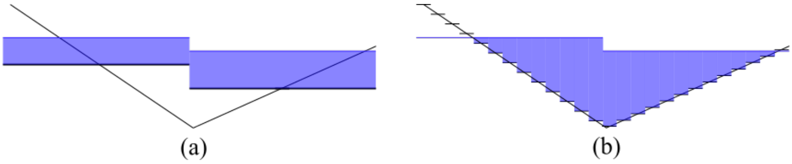


Figure 2.3. Volumes computed with the same number of cells with a simple grid (a) and with the addition of a subgrid (b). Oblique lines are the real bathymetry. The η is the same in both cases but the bathymetry can be more accurate with a subgrid

fine subgrid, see Figure 2.4 for a comparison. In the first case (fine grid without subgrid), each cell has an independently moving free surface, so accuracy is high; free surface waves profiles are more visible and wet/dry areas are more realistic. The price to pay for this is a very high amount of computations, which are time and memory consuming. This is due to the large number of unknowns (free surface elevations) and due to the small time step that is imposed by the CFL stability condition on a fine mesh. In the second case (coarse grid with subgrid) the flow in the domain is represented by a much smaller number of unknowns (free surface elevations), hence free surface wave profiles are less accurately resolved, but the decrease in accuracy is counterbalanced by a remarkable reduction of the unknowns and by a larger time step, since the CFL condition is only based on the mesh spacing of the grid and not on the spacing of the subgrid. At the aid of subgrid results are obtained much faster compared to a fine mesh simulation. A compromise has to be made to obtain

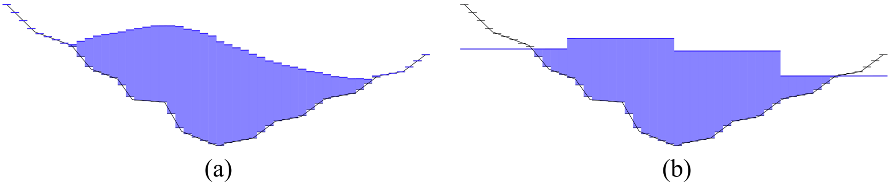


Figure 2.4. Differences between using only a fine grid (a) compared to a coarse grid with a subgrid (b). Δx on the left is equal to Δx_s on the right.

a sufficiently accurate result in a reasonable time; to do so, the subgrid is an invaluable tool. Some small scale information may be lost, but if they are not required there is a lot to be gained in efficiency. The actual number of cells and subcells should take in consideration all these observations. A possible decisional process is the following:

1. Choose the subcells size based on the bottom variability (steep slopes and frequent ups and down require small Δx_s), so that volume is accurately

computed. The total number of subcells in the domain is $N_{subc} = \dfrac{x_E - x_W}{\Delta x_s}$.

2. Choose I_{max} so that the free surface is accurate enough (depends on the purpose of the simulation); to help with the choice, one could simply start with a fairly small I_{max}, run a few time steps of the simulation, observe the results and progressively increase it, if necessary.

3. Calculate $s = \dfrac{N_{subc}}{I_{max}}$, rounded at the highest whole number. I_{max} and s are related: the smaller I_{max}, the bigger s.

2.3 Shape representation of floating objects on the subgrid and mass distribution

The representation of the ship is parametric: the x-y-z coordinates of a finite set of points is expected as input, they are placed in the domain space (according to the centre of mass position and orientations) and used as a blueprint to build the discretized ship, assuming they are connected by straight lines.
An algorithm has to be developed to find the intersections of the input hull with the vertical lines passing through every subcell centre. Those intersection will approximate the hull with piecewise constant values. In 1D/2D$_{xz}$ simulations this reconstruction is exact, i.e. it preserves the volume of ship, where the intersection is approximating only one segment. In the subcells with the input points, there are two segments with different orientations, and the intersection will be on only one segment, so the reconstruction is not exact there; provided a sufficiently fine subgrid, this effect is negligible. In Figure 2.5 is shown an example of interpolation in 1D/2D$_{xz}$; the input ship is defined by the black points, while the reconstructed ship is in red.
In 1D simulations with subgrid along y and in 2D$_{xy}$/3D simulations the idea of reconstruction is the same, but with the additional y dimension there are polygons instead of straight lines. Let us consider an example as shown in Figure 2.6 and follow the process steps. The input vessel is thought as composed of a series of horizontal layers, and the points are their intersections with the ship edges (an edge is highlighted in yellow); in the illustration there are 3 layers, each one with 6 points. Every close couple of points in a layer will form a quadrilateral (one is highlighted in green) with the corresponding couple on a next layer, except for the last one, which will be a generic polygon. At every time step, we are interested in finding the z elevation of the hull in the $(x, y)_{sub}$ subcells centres; Since the coordinates of the ship points $(x, y)_{ship}$ almost

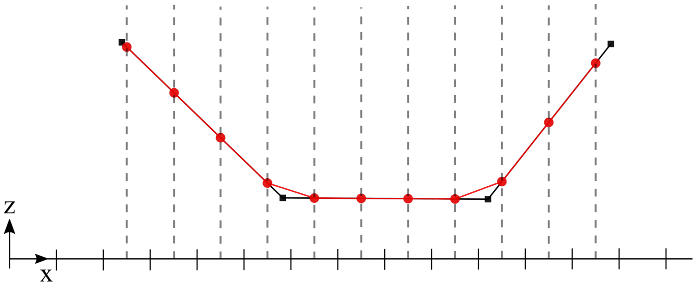


Figure 2.5. Illustration of the ship blueprint in 1D/2D$_{xz}$ and the interpolation on subgrid points

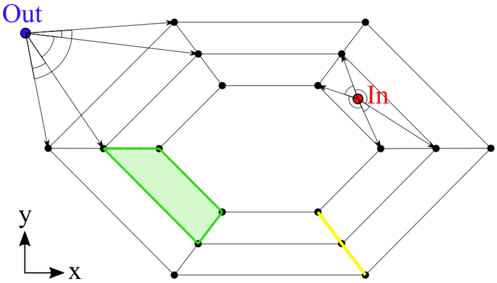


Figure 2.6. Illustration of the ship blueprint in 2D$_{xy}$/3D and the interpolation on subgrid points. The green lines highlight a quadrilateral, the yellow line is an edge.

never coincide with the subcells centres, an interpolation from the blueprint is necessary. A 2D$_{xy}$/3D algorithm carries out the following operations:

1. The portion of the subgrid surrounding the ship footprint (the projection of the ship points in the $x - y$ plane) is cropped out to avoid unnecessary computations where they are surely not needed

2. Each subgrid point p_{sub} (in Figure 2.6, the blue and red points are two examples) is tested to determine if it is inside or outside the footprint. To do so, one can start, for all the vessel polygons, by drawing the vectors connecting their vertices with the point p_{sub} and storing them. Then, the scalar product of all the possible ordered couples of vectors is done to compute the angles between them. If the sum of these angles is $< 2\pi$ the point is out, if it is $= 2\pi$ it is in the polygon. Practically, it is necessary to ease a bit these rules because the angles are computed with a finite precision so there are some errors: p_{sub} is considered inside if the sum of the angles is $\geq 2\pi - \xi$, with ξ a small number (like $2\pi/1000$, it may be adjusted based on the subgrid size)

3. Once the polygon surrounding p_{sub} is determined, it is possible to compute the vector $n = (n_x, n_y, n_z)$ orthogonal to the plane described by two polygon edges. The vertex between the two edges has coordinates $v = (x_v, y_v, z_v)$. The elevation of the hull in that subcell $l_{p_{sub}}$ is then:

$$l_{p_{sub}} = z_v - \frac{1}{n_z}\left[n_x(x_{p_{sub}} - x_v) + n_y(y_{p_{sub}} - y_v)\right] \tag{2.3}$$

With these operations, the input ship is reconstructed where it is needed, in the subcells centres. Interestingly, even when the water flow is 1D it is possible to use the subgrid to have a 3D representation of the ship and improve volume computations.

The mass distribution on the ship determines where the centre of mass is located and so, together with the pressure on the hull, it has a major role in the stability of the vessel. As a general rule, an unstable configuration is one with G high along the vertical and skewed on the port or starboard; once it starts rolling, the torque amplifies the rotational velocity and the ship capsizes. The mass distribution of the floating object is given at the start of the simulation and then kept constant. Thus, in a body-fixed frame of reference also the inertia I is constant. For convenience, in the test cases of Section 3 we used simplifying assumptions on where the mass is located. For applications with real ships, it would be better to remove those assumptions and use the available data on the structure and on the heavy equipment onboard. The following procedure describes how to generate an object with heterogeneous mass distribution:

1. Take the input ship rotated in upright position.

2. Apply the discretization of the hull at subgrid level (as seen in Section 2.3). Now it is possible to imagine the vessel as formed by a sequence of vertical stripes which are bounded below by the hull and above by the deck. Here we define the deck based on the highest elevation reached by the hull.

3. Assign at every stripe a share of the total mass m_s; this controls the horizontal position of the global centre of mass G. If the distribution is known it is possible to define the mass of each stripe manually; if unknown, a simple assumption is to set the masses proportional to the vertical lengths of the stripes.

4. Assign at every stripe the vertical position of its centre of mass; this controls the z position of the global centre of mass. If it is not known,

one could place it always at the same adimensional height: for example if the mass is homogeneously distributed the centre will be at 1/2 of the stripe vertical length.

5. Compute the global centre of mass G and place there the origin of the body-fixed frame of reference $x'z'$.

6. Compute the moment of inertia of the ship I_s using

$$I = \sum_s m_{s,s}\,(x'^2_s + z'^2_s). \tag{2.4}$$

The above procedure allows to have an heterogeneous mass distribution at the finest level of accuracy allowed by the subgrid resolution. It is also extendable to 3D simulations with the due changes (the stripes are now prisms and inertia is a 3x3 tensor).

The fluid-structure interaction problem imposes one more limitation to the volume in each cell, which should be bounded from above at the current elevation of the hull $\boldsymbol{l}$. The depth function is therefore also dependent on the position and orientation of the ship, which moves at each time step:

$$H(\eta_i, x_G, z_G, \varphi) = \frac{1}{s}\sum_s \max\left(0, \min(\eta_i - b_{i,s}, l_{i,s} - b_{i,s})\right) = H_1 - H_2. \tag{2.5}$$

where the Jordan decomposition of H is needed for the nested Newton algorithm of Casulli and Zanolli [35, 36]. An illustration of the relationship between H, H_1 and H_2 is provided in Figure 2.7.

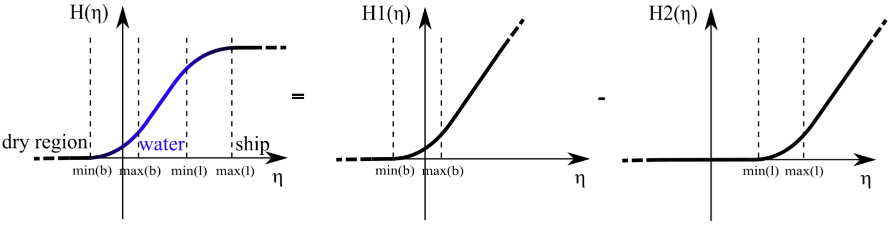


Figure 2.7. Schematic representation of the average depth functions H, H_1 and H_2. Smoothness is a consequence of the subgrid, as each cell can also be partially wet.

Figure 2.8 illustrates how the average water depths are computed on the subgrid in the presence of a floating rigid body. The considerations concerning the use of the subgrid for the vessel reconstruction are analogous to those made before for the bottom resolution on the subgrid for the hydrodynamics. Of course now the total number of subcells has also to be adjusted to discretize with enough accuracy the geometry of the ship, not only the bottom topography. A major difference between the vessel and the bottom is that the former is moving at each time step; the elevation of the hull $\boldsymbol{l}$ is sampled at the intersection of said hull with the vertical passing through the centre of the subcells, so in general the hull is not discretized in the same points. This reconstruction maximises the accuracy of the water volume computation and allows to have the piecewise constant hull $\boldsymbol{l}$ always aligned with the piecewise constant bathymetry $\boldsymbol{b}$. The ship is discretized by a finite number of subcells, which is proportional to the area of the projection of the vessel in the horizontal plane. This may cause some problems when the floating body has almost vertical sides (usually at the external part) because there may not be an adequate number of subcells for them, or none at all (depending on the orientation of the ship and the subcells size); the volume computation would not be significantly affected, but the horizontal forces would be. For this case a different approach to forces computation could be explored, for example sampling them in a constant number of points spread evenly on the hull, regardless of the orientation or the subgrid. In Section 2.16 it is explained how the use of an input triangulated surface overcomes the problem of the vertical sides.

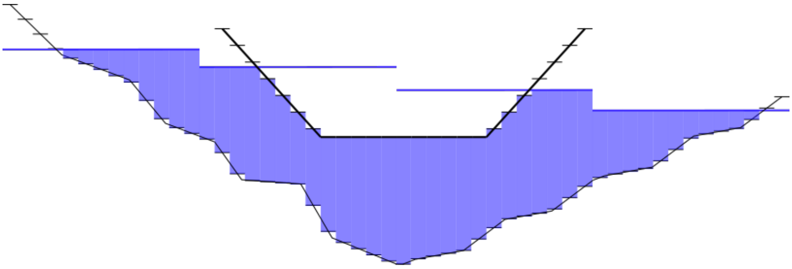


Figure 2.8. Volume computation around a floating ship. In this example the domain is divided in four cells (thus there are four η) and each cell has thirteen subcells.

With the present definition of the algorithm, it is not possible to simulate a completely immersed body since the free surface is defined by a single-valued function, so it is not possible to have water above the object.

2.4 1D model

We first introduce the new method for the simple one-dimensional case, in order to ease notation and to simplify the overall presentation, allowing to focus on the new main ingredients of the approach. Subsequently, we will then show the new method for the general three-dimensional case.

2.4.1 Shallow water equations

The shallow water equations are the starting point for our numerical model. They can be derived from depth-averaging of the incompressible Navier-Stokes equations, which describe the flow of an incompressible fluid with constant density, while conserving mass and linear momentum. Assuming that the flow is *hydrostatic* the equations simplify and become computationally much easier to solve; this assumption is valid when the vertical accelerations and the vertical viscosity are significantly smaller than the gravity acceleration and pressure gradient, which is usually true for geophysical flows. The one-dimensional shallow water equations with bottom friction read

$$\frac{\partial H(\eta)}{\partial t} + \frac{\partial q}{\partial x} = 0, \tag{2.6a}$$

$$\frac{\partial q}{\partial t} + \frac{\partial uq}{\partial x} + gH\frac{\partial \eta}{\partial x} = -\gamma q, \tag{2.6b}$$

where $g = 9.81\, m/s^2$ is the gravity constant, $u = u(x,t)$ is the average velocity, $H = H(x,t)$ is the water depth, $b = b(x)$ the bottom topography and $q = H \cdot u$ is the discharge. $\eta(x,t) = b(x) + H(x,t) + P_s(x,t)$ is the free surface elevation plus an additional height accounting for the pressure acting on the floating body, which is $P_s(x,t) = 0$ outside the body region. A typical dimensionless number appearing in shallow water flows is the Froude number $\mathrm{Fr} = |u|/\sqrt(gH)$, which allows to distinguish subcritial flows from supercritical ones. In the system above, $\gamma = \gamma(H,u)$ is a nonnegative bottom friction coefficient that can be defined e.g. using the Strickler parameter k_{St} and the hydraulic radius R_h. In the engineering practice, for wide channels where one can assume $R_h \approx H$ one obtains

$$\gamma = \frac{g \cdot |q|}{k_{St}^2 \cdot R_h^{\frac{1}{3}} \cdot H^2} \approx \frac{g \cdot |q|}{k_{St}^2 \cdot H^{\frac{7}{3}}} \tag{2.7}$$

If the roughness is known from a Manning coefficient n_M or a Chézy coefficient χ_C, the formula (2.7) is still valid, since the following relations hold:

$$k_{St} = \frac{1}{n_M} = \chi_C R_h^{-\frac{1}{6}} \tag{2.8}$$

Typical values for n_M in an open channel can be found in the work of Chow [112].

2.4.2 Dynamics of floating rigid bodies

Before coupling the floating rigid bodies (ships) with the flow dynamics, it is useful to specify how their kinematics and dynamics is handled, as well as their shape and mass distribution. A ship is treated as a moving rigid body with total mass m_s. It is convenient to define a body-fixed frame of reference $x'z'$, with the origin in its centre of mass $\boldsymbol{G} = (x_G, z_G)$; x' lies on the longitudinal axis of the vessel and is responsible for surge motion, while z' ($\perp x'$) points up towards the sky and is responsible for heave motion. Only one point is needed to track the ship position, all the others can be derived starting from the known one; this special point is set to be the centre of mass. In 1D and 2D$_{xz}$ simulations the vessel has three degrees of freedom, two for translation along the horizontal and vertical axes x and z, one for the rotation around the y axis. The orientation of the object in the space is represented by the angle φ, defined as the angle between axes z and z'.

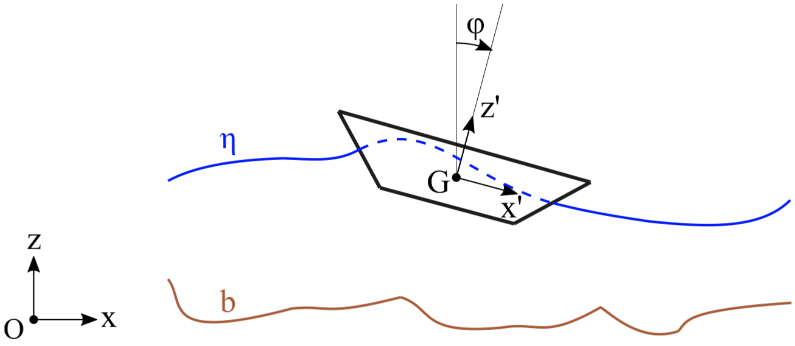


Figure 2.9. General 1D/2D$_{xz}$ scheme of the ship motion problem

The equations that govern the movement are derived from the conservation of linear and angular momentum:

$$m_s \frac{du_G}{dt} = \sum F_{x,s}, \tag{2.9a}$$

$$m_s \frac{dw_G}{dt} = -m_s g + \sum F_{z,s}, \tag{2.9b}$$

$$\frac{d}{dt} I_s \dot{\varphi} = \tau, \tag{2.9c}$$

and from the definitions of the linear and angular velocities

$$\begin{aligned}\frac{dx_G}{dt} &= u_G, &\text{(2.10a)}\\ \frac{dz_G}{dt} &= w_G, &\text{(2.10b)}\\ \frac{d\varphi}{dt} &= \dot{\varphi}. &\text{(2.10c)}\end{aligned}$$

In the equations of motion m_s is the ship mass and I_s is the ship's moment of inertia relative to G, constant in the body-fixed reference frame and given by (2.4). The forces acting on the vessel are those generated by gravity F_g (always vertical) and the hydrostatic pressure orthogonal to the hull, which creates a horizontal and a vertical component of the force F_x and F_z. The ship will then be able to float and passively being carried by an imbalance in the free surface; also, these forces will create a torque that will make the ship oscillate. At this stage of development of the model, the propulsion force and the steering moment are not considered, but we only consider the following forces:

$$\begin{aligned}F_{x,s} = &\quad \rho g \ \max(0, \eta_s - l_s)\, \Delta x_s \ \tan(\alpha_s),\\ F_{z,s} = &\quad \rho g \ \max(0, \eta_s - l_s)\, \Delta x_s,\\ \tau = &\quad \sum_s \left[F_{x,s}(l_s - z_G) + F_{z,s}(x_G - x_s)\right].\end{aligned} \tag{2.11}$$

where ρ is the water density, l_s is the hull elevation at the subcell s centre, α_s is the smallest angle between the horizontal and a vector orthogonal to the hull in that subcell, x_s is the x coordinate of the subcell s centre.

2.4.3 Semi-implicit discretization

The equations of motion of the rigid body (2.9)-(2.10) provide the accelerations that are necessary to compute changes in the ship position, orientation and velocities. At the beginning of each time step, they are updated using an explicit second order accurate Taylor method:

$$x_G^{n+1} = x_G^n + u_G^n \cdot \Delta t + \frac{1}{2}\frac{\sum_s F_{x,s}^n}{m_s} \cdot \Delta t^2, \tag{2.12a}$$

$$z_G^{n+1} = z_G^n + w_G^n \cdot \Delta t + \frac{1}{2}\frac{\sum_s F_{z,s}^n - F_g}{m_s} \cdot \Delta t^2, \tag{2.12b}$$

$$\varphi^{n+1} = \varphi^n + \dot{\varphi}^n \Delta t + \frac{1}{2}\frac{\tau^n}{I}\Delta t^2, \tag{2.12c}$$

$$u_G^{n+1} = u_G^n + \frac{\sum_s F_{x,s}^n}{m_s} \cdot \Delta t, \tag{2.12d}$$

$$w_G^{n+1} = w_G^n + \frac{\sum_s F_{z,s}^n - F_g}{m_s} \cdot \Delta t, \tag{2.12e}$$

$$\dot{\varphi}^{n+1} = \dot{\varphi}^n + \frac{\tau^n}{I}\Delta t. \tag{2.12f}$$

With the notation $f = q \cdot u = H \cdot u^2$ and $\boldsymbol{G} = (x_G, z_G)$, a semi-implicit discretization of the 1D shallow water system (2.6) can be written as:

$$H_i(\eta_i^{n+1}, \boldsymbol{G}^{n+1}, \varphi^{n+1}) = H(\eta_i^n, \boldsymbol{G}^n, \varphi^n) - \frac{\Delta t}{\Delta x}\left(q_{i+\frac{1}{2}}^{n+1} - q_{i-\frac{1}{2}}^{n+1}\right) \quad , \tag{2.13a}$$

$$q_{i+\frac{1}{2}}^{n+1} = q_{i+\frac{1}{2}}^{*} - \frac{\Delta t}{\Delta x} g H_{i+\frac{1}{2}}^n \left(\eta_{i+1}^{n+1} - \eta_i^{n+1}\right) - \Delta \gamma_{i+\frac{1}{2}}^n q_{i+\frac{1}{2}}^{n+1} \quad . \tag{2.13b}$$

The fluxes in the mass conservation equation, the pressure gradient in the momentum equation and the bottom friction term are discretized implicitly, while the nonlinear convective terms are discretized in an explicit manner. In order to avoid nonlinearities in the implicit part, the water depth at the cell interface $H_{i+\frac{1}{2}}^n$ is taken explicitly, as well as the bottom friction coefficient $\gamma_{i+\frac{1}{2}}^n$. For subcritical flows (Fr ≤ 1), $H_{i+\frac{1}{2}}^n$ is taken equal to the average of the water depths H_i^n and H_{i+1}^n, while for supercritical flows (Fr > 1) we use the upwind water depth, hence

$$H_{i+\frac{1}{2}}^n = \begin{cases} \dfrac{H_i^n + H_{i+1}^n}{2} & \text{if Fr} \leq 1, \\ H_i^n & \text{if Fr} > 1 \text{ and } q_{i+\frac{1}{2}} \geq 0, \\ H_{i+1}^n & \text{if Fr} > 1 \text{ and } q_{i+\frac{1}{2}} < 0. \end{cases} \tag{2.14}$$

The term $q_{i+\frac{1}{2}}^*$ contains the explicit discretization of the nonlinear convective terms:

$$q_{i+\frac{1}{2}}^* = q_{i+\frac{1}{2}}^n - \frac{\Delta t}{\Delta x}\left(f_{i+1} - f_i\right) \tag{2.15}$$

The fluxes in equation (2.15) are approximated by any stable numerical flux. Here three possible choices are presented: the momentum-conservative Stelling&Duinmeijer flux f^{SD} [109], the shock-capturing Rusanov flux f^R[101] and the energy-conservative Ducros flux f^D [49]:

$$f_i^{SD} = \frac{q^n_{i+\frac{1}{2}} + q^n_{i-\frac{1}{2}}}{2} u^{upwind}_{i+1}, \qquad \text{with } u_i^{upwind} = \begin{cases} u^n_{i-\frac{1}{2}} & \text{if } \dfrac{q^n_{i+\frac{1}{2}} + q^n_{i-\frac{1}{2}}}{2} > 0 \\ u^n_{i+\frac{1}{2}} & \text{if } \dfrac{q^n_{i+\frac{1}{2}} + q^n_{i-\frac{1}{2}}}{2} \leq 0 \end{cases} . \tag{2.16}$$

$$f_i^R = \frac{1}{2}\left[(qu)^n_{i+\frac{1}{2}} + (qu)^n_{i-\frac{1}{2}}\right] - \frac{1}{2} s_i^{max}\left(q^n_{i+\frac{1}{2}} - q^n_{i-\frac{1}{2}}\right). \tag{2.17}$$

$$f_i^D = \frac{1}{4}\left(q^n_{i+\frac{1}{2}} + q^n_{i-\frac{1}{2}}\right) \cdot \left(u^n_{i+\frac{1}{2}} + u^n_{i-\frac{1}{2}}\right) - \frac{1}{2} s_i^{max}\left(q^n_{i+\frac{1}{2}} - q^n_{i-\frac{1}{2}}\right), \tag{2.18}$$

with the wave speed estimate

$$s_i^{max} = 2 \cdot \max\left(\left|u^n_{i+\frac{1}{2}}\right|, \left|u^n_{i-\frac{1}{2}}\right|\right) + c. \tag{2.19}$$

The product $2u$ in equation (2.19) is the system eigenvalue emerging from a characteristic analysis [24]. The term c is a small constant, for example 0.1, whose purpose is to always force the wave speed to be non-zero even for $u = 0$, so the unstable central flux is avoided. A simple average of f^R and f^{SD} is employed. The time step size Δt is adjusted at every time iteration and it is based on the Courant-Friedrichs-Lewy (CFL) condition to assure stability [44]:

$$\Delta t = \min\left(\Delta t_{max}\, ,\, \frac{\text{CFL} \cdot \Delta x}{2|u|_{max} + c}\right) \qquad \text{with} \qquad \text{CFL} = 0.9. \tag{2.20}$$

A small constant has been introduced in the denominator to avoid division by 0 in the case of still water. A maximum for the time step size can also be set if accuracy requires it. The discretized system (2.13) can be reduced by eliminating the unknown variables q in the mass equation through substitution from the momentum equation:

$$\begin{aligned} H_i(\eta_i^{n+1}, \boldsymbol{G}^{n+1}, \varphi^{n+1}) - \frac{\Delta t^2}{\Delta x^2} \Bigg[& \frac{g H^n_{i+\frac{1}{2}}}{1 + \Delta t \, \gamma^n_{i+\frac{1}{2}}} \left(\eta^{n+1}_{i+1} - \eta^{n+1}_i \right) \\ & - \frac{g H^n_{i-\frac{1}{2}}}{1 + \Delta t \, \gamma^n_{i-\frac{1}{2}}} \left(\eta^{n+1}_{i} - \eta^{n+1}_{i-1} \right) \Bigg] \\ = H_i(\eta_i^{n}, \boldsymbol{G}^{n}, \varphi^{n}) - \frac{\Delta t}{\Delta x} & \left(\frac{q^*_{i+\frac{1}{2}}}{1 + \Delta t \, \gamma^n_{i+\frac{1}{2}}} - \frac{q^*_{i-\frac{1}{2}}}{1 + \Delta t \, \gamma^n_{i-\frac{1}{2}}} \right). \end{aligned} \tag{2.21}$$

The resulting mildly nonlinear system, which now contains the free surface at the new time as the only unknown can be easily solved at the aid of a (nested) Newton method, see [21, 22, 35, 36].

In the final mildly nonlinear pressure system (2.21) the location of the ship at the old and new time levels affects the calculation of the water volume in each cell, and the water exerts pressure forces on the hull which are taken into account in the discretized ODE system of the rigid body dynamics (2.12), so the fluid and the structure are coupled with each other in both directions.

2.5 2Dxz model

The model described in Section 2.4 is applicable in 1D, and for some simplified cases a 1D approach may be sufficient. However, many real applications require the extension to higher dimensions and more details about the velocities. In this section and in the subsequent Section 2.7, the 2D$_{xz}$ and 3D models are presented, together with the description of their most relevant characteristics. Attention will be directed to what is new compared to the 1D case.

The 2D$_{xz}$ model has one horizontal and one vertical dimension, so the domain is a vertical section discretized on a grid of $I_{max} \cdot K_{max}$ cells, whose size is $\Delta x \cdot \Delta z_k$ each. The grid is staggered, so the vertical velocity w is defined at the interface with the cell above or below. The free surface elevation $\eta(t, x)$ and the bathymetry $b(x)$ can vary spatially only moving in the horizontal direction, so they are the same along a vertical column of the grid. All the variables represent lateral averages. In the mass conservation equation the difference of fluxes has to take into account all the different horizontal velocities along the vertical, so on each interface the flux is computed with an integral. The momentum equation has a new advective term and the viscosity along the vertical is added; the horizontal viscosity is neglected in this Section because, with the assumption of shallow water, it is usually fair to say that $(\nu u_z)_z \gg (\nu u_x)_x$.

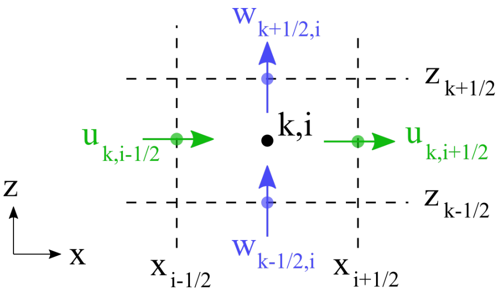


Figure 2.10. Grid in the 2D$_{xz}$ model

The horizontal viscosity is introduced later, in Section 2.14. With the vertical velocity as new unknown, the system needs one more equation, stating that the divergence of the flow velocity is zero.

$$\begin{aligned} \frac{\partial}{\partial t}\left[H(\eta)\right]+\frac{\partial}{\partial x}\left[\int_b^\eta u(x,z,t)dz\right] &= 0, && (2.22a)\\ \frac{\partial u}{\partial t}+\frac{\partial\left(uu\right)}{\partial x}+\frac{\partial\left(uw\right)}{\partial z}+g\frac{\partial\eta}{\partial x} &= \frac{\partial}{\partial z}\left(\nu\frac{\partial u}{\partial z}\right), && (2.22b)\\ \frac{\partial u}{\partial x}+\frac{\partial w}{\partial z} &= 0. && (2.22c) \end{aligned}$$

The momentum equation in system (2.22) is written before integration along the vertical, therefore the unknown is the primitive variable u. Only after the integration $\int u(x,z)dz = q$ the conserved variable appears, see also [33]. In this paper we make use of z-layers, hence the integral in the first equation of (2.22) is discretized as follows:

$$\int_b^\eta u(x,z)dz \approx \sum_k q_{i+\frac{1}{2},k} = \sum_k \Delta z_{i+\frac{1}{2},k} u_{i+\frac{1}{2},k}, \qquad k = 1, 2, ..., K_{max}. \tag{2.23}$$

The vertical mesh spacing depends on the local bottom and on the free surface elevation. More specifically we choose $\Delta z_k = \Delta z$ for all cells that are completely included between b and η, while the length of cells adjacent to the bottom and the free surface are adjusted, so that

$$\sum_k \Delta z_{i+\frac{1}{2},k} = H_{i+\frac{1}{2}} = \max(0, \min(\eta_{i+\frac{1}{2}} - b_{i+\frac{1}{2}}, l_{i+\frac{1}{2}} - b_{i+\frac{1}{2}})). \tag{2.24}$$

The advective term $\partial(qu)/\partial x$ is discretized in the same fashion as described in Section 2.4.3. Instead, for the advective term with mixed velocities we apply a central scheme. Particular care is needed because the product is defined at

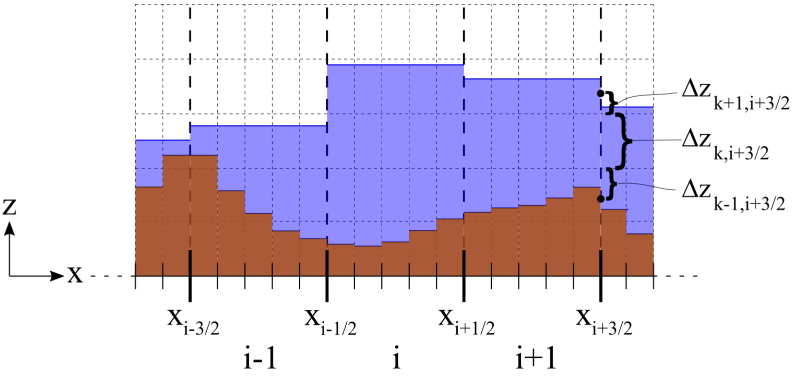


Figure 2.11. General 2D$_{xz}$ scheme, showing the Δz_k at the interface $i+\frac{3}{2}$

the corners of the cell, but on the staggered grid none of the factors are there; some sort of average is needed. The discretization is done using the arithmetic mean of the velocity and linear momentum at time n, as written in equation (2.25) and illustrated in Figure 2.12:

$$\begin{aligned}\frac{\partial\left(qw\right)}{\partial z}\approx&\frac{1}{\Delta z}\left(\frac{q^n_{i+\frac{1}{2},k+1}+q^n_{i+\frac{1}{2},k}}{2}\right)\left(\frac{w^n_{i+1,k+\frac{1}{2}}+w^n_{i,k+\frac{1}{2}}}{2}\right)\\&-\frac{1}{\Delta z}\left(\frac{q^n_{i+\frac{1}{2},k}+q^n_{i+\frac{1}{2},k-1}}{2}\right)\left(\frac{w^n_{i+1,k-\frac{1}{2}}+w^n_{i,k-\frac{1}{2}}}{2}\right).\end{aligned}\tag{2.25}$$

Because the advective term (2.25) is explicit, it is added to the definition of q^*.

The introduction of the vertical viscosity requires boundary conditions due to stresses applied on the boundaries, e.g. roughness or wind stress. Indeed, at the vertical water boundaries the product $\nu\partial u/\partial z$ is substituted by the boundary conditions:

$$\begin{cases}\nu\dfrac{\partial u}{\partial z}=\gamma_T\left(u^a-u^s\right) & \text{at the free surface}\\ \nu\dfrac{\partial u}{\partial z}=\gamma_B u^b & \text{at the bottom}\end{cases}\tag{2.26}$$

with the wind and bottom coefficients

$$\begin{aligned}\gamma_T&=C_d\frac{\rho_a}{\rho}|u^a|,\\ \gamma_B&=\frac{g|u^b|}{k^2_{St}R_h^{1/3}}.\end{aligned}\tag{2.27}$$

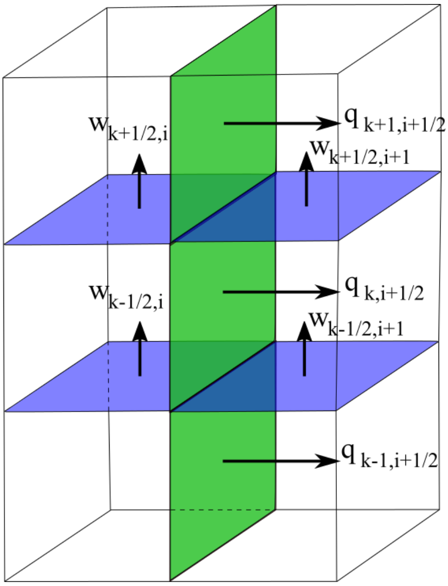


Figure 2.12. Illustration of terms in the mixed velocities product on a staggered grid

where ν is the kinematic viscosity coefficient, C_d is a drag coefficient, $\rho_a = 1.25$ kg/m^3 is the air density, u^a is the horizontal component of the wind speed, u^s and u^b are the horizontal component of the water velocity at the surface and at the bottom, respectively. Viscosity is treated as an implicit term:

$$\begin{aligned}\frac{\partial}{\partial z}\left(\Delta z \nu \frac{\partial u}{\partial z}\right) \approx \frac{1}{\Delta z^n_{i+\frac{1}{2},k}} \Bigg[& \nu^{n+1}_{i+\frac{1}{2},k+\frac{1}{2}}\left(u^{n+1}_{i+\frac{1}{2},k+1} - u^{n+1}_{i+\frac{1}{2},k}\right) \\ & -\nu^{n+1}_{i+\frac{1}{2},k-\frac{1}{2}}\left(u^{n+1}_{i+\frac{1}{2},k} - u^{n+1}_{i+\frac{1}{2},k-1}\right)\Bigg].\end{aligned} \tag{2.28}$$

Due to the viscosity, each momentum equation requires the neighbour velocities, so it not possible to simply do a substitution in the mass equation as was done in the 1D model. The matrix form of system (2.22) (the incompressibility equation is not needed at the moment) is:

$$\begin{aligned}H(\eta_i^{n+1}, \boldsymbol{G}^{n+1}, \varphi^{n+1}) =& H(\eta_i^n, \boldsymbol{G}^n, \varphi^n) \\ & -\frac{\Delta t}{\Delta x}\left[\left(\boldsymbol{\Delta Z}_{i+\frac{1}{2}}\right)^T \boldsymbol{U}^{n+1}_{i+\frac{1}{2}} - \left(\boldsymbol{\Delta Z}_{i+\frac{1}{2}}\right)^T \boldsymbol{U}^{n+1}_{i-\frac{1}{2}}\right],\end{aligned} \tag{2.29a}$$

$$\boldsymbol{A}^n_{i+\frac{1}{2}} \boldsymbol{U}^{n+1}_{i+\frac{1}{2}} = \boldsymbol{E}^n_{i+\frac{1}{2}} - g\frac{\Delta t}{\Delta x}(\eta^{n+1}_{i+1} - \eta^{n+1}_i)\boldsymbol{\Delta Z}^2_{i+\frac{1}{2}}, \tag{2.29b}$$

where $\boldsymbol{\Delta Z}^2_{i+\frac{1}{2}}$ is the vector with the squared wetted distances along the vertical at time n, $\boldsymbol{U}^{n+1}_{i+\frac{1}{2}}$ is for the unknown horizontal velocities, $\boldsymbol{A}^n_{i+\frac{1}{2}}$ is a square symmetric tridiagonal matrix, $\boldsymbol{E}^n_{i+\frac{1}{2}}$ is the vector with all the explicit terms. The equations are computed only in the non-dry cells of the domain column (to avoid unnecessary calculations), so on the vertical cells $k = m_{i+\frac{1}{2}}, m_{i+\frac{1}{2}}+1, ..., M^n_{i+\frac{1}{2}}$. Start index $m_{i+\frac{1}{2}}$ is only spatially variable according to the bottom elevation, while end index $M^n_{i+\frac{1}{2}}$ is spatially variable and time dependent according to the free surface current position and the bathymetry.

$$\boldsymbol{\Delta Z}^2 = \begin{pmatrix} \Delta z^2_M \\ \Delta z^2_{M-1} \\ \vdots \\ \Delta z^2_m \end{pmatrix}, \qquad \boldsymbol{U} = \begin{pmatrix} u_M \\ u_{M-1} \\ \vdots \\ u_m \end{pmatrix}, \qquad \boldsymbol{E}^n = \begin{pmatrix} \Delta z_M q^*_M + \Delta t \gamma_T \Delta z_M u_a \\ \Delta z_{M-1} q^*_{M-1} \\ \vdots \\ \Delta z_m q^*_m \end{pmatrix} \tag{2.30}$$

$$\boldsymbol{A} = \begin{pmatrix} d_M + a_{M-\frac{1}{2}} + \gamma_T \Delta t \Delta z_M & -a_{M-\frac{1}{2}} & 0 & ... & 0 \\ -a_{M-\frac{1}{2}} & d_{M-1} + a_{M-\frac{1}{2}} + a_{M-\frac{3}{2}} & -a_{M-\frac{3}{2}} & & \\ \vdots & \vdots & \vdots & & \\ 0 & 0 & & -a_{m+\frac{1}{2}} & d_m + a_{m+\frac{1}{2}} + \gamma_B \Delta t \Delta z_m \end{pmatrix} \tag{2.31}$$

with $d_k = \Delta z^2_k$ and $a_{k\pm\frac{1}{2}} = \Delta t\, \nu_{k\pm\frac{1}{2}}$, $k = m, m+1, ..., M$. The substitution of the momentum equation in the mass equation and use of the abbreviation $H^n_i = H(\eta^n_i, x^n_G, z^n_G, \varphi^n)$ lead to:

$$\begin{aligned} H^{n+1}_i - g\frac{\Delta t^2}{\Delta x^2}\Bigg\{ & \left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z^2}\right)^n_{i+\frac{1}{2}} \left[(\eta)^{n+1}_{i+1} - (\eta)^{n+1}_i\right] \\ & - \left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z^2}\right)^n_{i-\frac{1}{2}} \left[(\eta)^{n+1}_i - (\eta)^{n+1}_{i-1}\right] \Bigg\} = r^n_i, \end{aligned} \tag{2.32}$$

with

$$r^n_i = H^n_i - \frac{\Delta t}{\Delta x}\left[\left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E}\right)^n_{i+\frac{1}{2}} - \left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E}\right)^n_{i-\frac{1}{2}}\right]. \tag{2.33}$$

It is important to notice that it is never necessary to compute the inverse $\boldsymbol{A}^{-1}$, because the products $\boldsymbol{A}^{-1}\boldsymbol{\Delta Z}^2$ and $\boldsymbol{A}^{-1}\boldsymbol{E}$ can be obtained solving the systems $\boldsymbol{AB} = \boldsymbol{\Delta Z}^2$ and $\boldsymbol{AC} = \boldsymbol{E}$, using the Thomas algorithm for tridiagonal systems.

Starting from equation (2.32) it is possible to solve the system for $\boldsymbol{\eta}^{n+1}$ with the nested Newton method of Casulli and Zanolli. The size of this system is determined by the horizontal resolution, because the vertical one intervenes only during the assembly of its coefficients; this feature is very advantageous for numerical efficiency, also in the fully 3D case for predominantly horizontal flows, so it is possible to easily increase the vertical resolution at reasonable computational costs. With the solution $\boldsymbol{\eta}^{n+1}$, the velocities are computed from the solution of the I_{max} linear systems formed by the momentum equations written in matrix form, making use of the already known products $\boldsymbol{A}^{-1}\boldsymbol{\Delta Z}^2$ and $\boldsymbol{A}^{-1}\boldsymbol{E}$. The vertical velocities are obtained *a posteriori* from the incompressibility equation. Starting from the bottom with $w^{n+1}_{m-\frac{1}{2},i} = 0$, the others are found one after the other:

$$w^{n+1}_{i,k+\frac{1}{2}} = w^{n+1}_{i,k-\frac{1}{2}} - \frac{\Delta z^n_{i+\frac{1}{2},k} u^{n+1}_{i+\frac{1}{2},k} - \Delta z^n_{i-\frac{1}{2},k} u^{n+1}_{i-\frac{1}{2},k}}{\Delta x} \tag{2.34}$$

with $k = m_i, m_i + 1, ..., M_i^n - 1$.
An interesting aspect of this formulation of the algorithm is that for $K_{max} = 1$ it automatically reduces to the $1D$ model previously described.

2.6 Ship dynamics in three space dimensions

When the grid or the subgrid expands in both the x and y horizontal directions, it is possible to place in the domain a three-dimensional floating object with six degrees of freedom, so the equations for the ship dynamics written in Section 2.4.2 need to be extended.
First, let us define the vectors:

$$\boldsymbol{G} = \begin{pmatrix} x_G \\ y_G \\ z_G \end{pmatrix}, \quad \boldsymbol{u}_G = \begin{pmatrix} u_G \\ v_G \\ w_G \end{pmatrix}, \quad \boldsymbol{\varphi} = \begin{pmatrix} \varphi_x \\ \varphi_y \\ \varphi_z \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix}, \quad \boldsymbol{\tau} = \begin{pmatrix} \tau_x \\ \tau_y \\ \tau_z \end{pmatrix}$$

$\boldsymbol{G}$ is the position of the centre of mass relative to the inertial frame of reference. The others are relative to the fixed-body frame of reference: $\boldsymbol{u}_G$ holds the ship velocities so that u_G is the surge motion, v_G is the sway motion, w_G is the heave motion; they change when subject to the force $\boldsymbol{F}$. The orientation $\boldsymbol{\varphi}$ is relative to the $x'y'z'$ axes and change when the body is subject to the torque $\boldsymbol{\tau}$, causing roll, pitch and yaw motion. The system of equation that describes the translation and rotation of the vessel is the following:

$$m_s \dot{\boldsymbol{u}}_G^{n+1} + \dot{\boldsymbol{\varphi}}^n \times (m_s \boldsymbol{u}_G^n) = \boldsymbol{F}^n. \tag{2.35a}$$

$$\boldsymbol{I}_s \ddot{\boldsymbol{\varphi}}^{n+1} + \dot{\boldsymbol{\varphi}}^n \times (\boldsymbol{I}_s \dot{\boldsymbol{\varphi}}^n) = \boldsymbol{\tau}^n. \tag{2.35b}$$

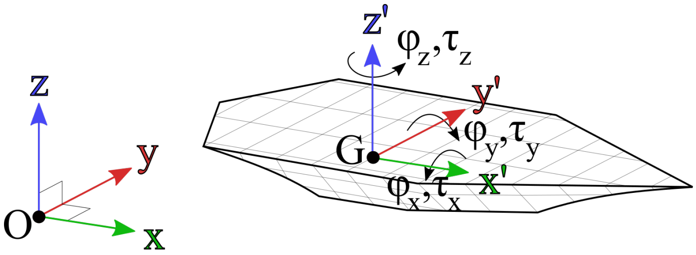


Figure 2.13. General 2D$_{xy}$/3D scheme of the ship motion problem

Here, $\dot{\boldsymbol{u}}$ is the linear acceleration, $\dot{\boldsymbol{\varphi}}$ is the angular velocity and $\ddot{\boldsymbol{\varphi}}$ is the angular acceleration. $\boldsymbol{I}_s$ is the inertia tensor computed in the body-fixed frame of reference (so it is constant during the simulation):

$$\boldsymbol{I}_s = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix},$$

whose components are the moments and products of inertia:

$$\begin{aligned} I_x &= \sum_s m_{s,s}(y'^2_s + z'^2_s) \qquad & I_{xy} &= I_{yx} = \sum_s m_{s,s}\, x'_s\, y'_s \\ I_y &= \sum_s m_{s,s}(x'^2_s + z'^2_s) \qquad & I_{xz} &= I_{zx} = \sum_s m_{s,s}\, x'_s\, z'_s \\ I_z &= \sum_s m_{s,s}(x'^2_s + y'^2_s) \qquad & I_{yz} &= I_{zy} = \sum_s m_{s,s}\, y'_s\, z'_s \end{aligned} \tag{2.36}$$

In analogy with Section 2.4.2, the force $\boldsymbol{F}$ is generated by gravity and the hydrostatic pressure acting on the hull. The pressure force is always orthogonal to the hull, so a normal vector is computed in each subcell relevant for the ship; to do so, it is possible to leverage the parametric representation of ship (see Figure 2.6): two vectors are drawn from the subcell centre to the vertices of the polygon surrounding the centre. Further on, we apply a normalized orthogonal vector $\hat{n}_s = \{\hat{n}_{x,s}, \hat{n}_{y,s}, \hat{n}_{z,s}\}$ resulting from the cross product of these two vectors. The forces are computed in a frame of reference parallel to the inertial one but centred in $\boldsymbol{G}$ (we mark them with a tilde):

$$\begin{aligned} \tilde{F}_x &= \sum_s \tilde{F}_{x,s} & \text{with} \qquad \tilde{F}_{x,s} &= \rho g \,\max(0, \eta_s - l_s)\, \frac{\Delta x_s \Delta y_s}{||\hat{n}_{z,s}||}\, \hat{n}_{x,s} \\ \tilde{F}_y &= \sum_s \tilde{F}_{y,s} & \text{with} \qquad \tilde{F}_{y,s} &= \rho g \,\max(0, \eta_s - l_s)\, \frac{\Delta x_s \Delta y_s}{||\hat{n}_{z,s}||}\, \hat{n}_{y,s} \\ \tilde{F}_z &= \sum_s \tilde{F}_{z,s} - mg & \text{with} \qquad \tilde{F}_{z,s} &= \rho g \,\max(0, \eta_s - l_s)\, \frac{\Delta x_s \Delta y_s}{||\hat{n}_{z,s}||}\, \hat{n}_{z,s} \end{aligned} \tag{2.37}$$

The fraction $\frac{\Delta x_s \Delta y_s}{||\hat{n}_{z,s}||}$ is the approximation of the hull area included in the subcell: the subcell area $\Delta x_s \Delta y_s$ is increased though a division with the inclination of the hull with respect to the horizontal plane, represented by the magnitude of $\hat{n}_{z,s}$, which is $0 < ||\hat{n}_{z,s}|| \leq 1$ ($= 1$ when the piece of the vessel hull in s is horizontal). The torques are the products of these forces on the hull with the distances to the axes (which are centred in $\boldsymbol{G}$), following the right hand rule:

$$\begin{aligned}
\tilde{\tau}_x &= \sum_s \left[\tilde{F}_{y,s}(z_G - l_s) + \tilde{F}_{z,s}(y_s - y_G)\right], \\
\tilde{\tau}_y &= \sum_s \left[\tilde{F}_{z,s}(x_G - x_s) + \tilde{F}_{x,s}(l_s - z_G)\right], \\
\tilde{\tau}_z &= \sum_s \left[\tilde{F}_{x,s}(y_G - y_s) + \tilde{F}_{y,s}(x_s - x_G)\right].
\end{aligned} \tag{2.38}$$

In order to relate the forces and torques to the body-fixed frame of reference the rotation matrix $\boldsymbol{R}$ is applied:

$$\boldsymbol{R} = \begin{bmatrix} cos(\varphi_z) & -sin(\varphi_z) & 0 \\ sin(\varphi_z) & cos(\varphi_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\varphi_y) & 0 & sin(\varphi_y) \\ 0 & 1 & 0 \\ -sin(\varphi_y) & 0 & cos(\varphi_y) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\varphi_x) & -sin(\varphi_x) \\ 0 & sin(\varphi_x) & cos(\varphi_x) \end{bmatrix}. \tag{2.39}$$

So that:

$$\begin{aligned}
\boldsymbol{F} &= \boldsymbol{R}^T \tilde{\boldsymbol{F}} \\
\boldsymbol{\tau} &= \boldsymbol{R}^T \tilde{\boldsymbol{\tau}}
\end{aligned} \tag{2.40}$$

It is now possible to solve system (2.35). In explicit form:

$$\begin{cases}
\dot{u}^{n+1} & -v^n \dot{\varphi}_z^n + w^n \dot{\varphi}_y^n = \dfrac{F_x^n}{m} \\
\dot{v}^{n+1} & -w^n \dot{\varphi}_x^n + u^n \dot{\varphi}_z^n = \dfrac{F_y^n}{m} \\
\dot{w}^{n+1} & -u^n \dot{\varphi}_y^n + v^n \dot{\varphi}_x^n = \dfrac{F_z^n}{m} \\
I_x \ddot{\varphi}_x^{n+1} & +(I_z - I_y)\dot{\varphi}_y^n \dot{\varphi}_z^n - (\ddot{\varphi}_z^{n+1} + \dot{\varphi}_x^n \dot{\varphi}_y^n) I_{xz} \\
 & +(\dot{\varphi}_y^2 - \dot{\varphi}_z^2)^n I_{yz} + (\dot{\varphi}_x^n \dot{\varphi}_z^n - \ddot{\varphi}_y^{n+1}) I_{xy} = \tau_x^n \\
I_y \ddot{\varphi}_y^{n+1} & +(I_x - I_z)\dot{\varphi}_z^n \dot{\varphi}_x^n - (\ddot{\varphi}_x^{n+1} + \dot{\varphi}_y^n \dot{\varphi}_z^n) I_{yx} \\
 & +(\dot{\varphi}_z^2 - \dot{\varphi}_x^2)^n I_{zx} + (\dot{\varphi}_y^n \dot{\varphi}_x^n - \ddot{\varphi}_z^{n+1}) I_{yz} = \tau_y^n \\
I_z \ddot{\varphi}_z^{n+1} & +(I_y - I_x)\dot{\varphi}_x^n \dot{\varphi}_y^n - (\ddot{\varphi}_y^{n+1} + \dot{\varphi}_z^n \dot{\varphi}_x^n) I_{zy} \\
 & +(\dot{\varphi}_x^2 - \dot{\varphi}_y^2)^n I_{xy} + (\dot{\varphi}_z^n \dot{\varphi}_y^n - \ddot{\varphi}_x^{n+1}) I_{zx} = \tau_z^n .
\end{cases} \tag{2.41}$$

The equations for the conservation of the angular momentum are coupled, so the matrix form is solved to get the unknown $\ddot{\boldsymbol{\varphi}}^{n+1}$:

$$\begin{pmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{pmatrix} \begin{pmatrix} \ddot{\varphi}_x^{n+1} \\ \ddot{\varphi}_y^{n+1} \\ \ddot{\varphi}_z^{n+1} \end{pmatrix} = \begin{pmatrix} \tau_x^n - (I_z - I_y)\dot{\varphi}_y^n \dot{\varphi}_z^n + \dot{\varphi}_x^n \dot{\varphi}_y^n I_{xz} \\ \qquad - (\dot{\varphi}_y^2 - \dot{\varphi}_z^2)^n I_{yz} - \dot{\varphi}_x^n \dot{\varphi}_z^n I_{xy} \\ \\ \tau_y^n - (I_x - I_z)\dot{\varphi}_z^n \dot{\varphi}_x^n + \dot{\varphi}_y^n \dot{\varphi}_z^n I_{yx} \\ \qquad - (\dot{\varphi}_z^2 - \dot{\varphi}_x^2)^n I_{zx} - \dot{\varphi}_y^n \dot{\varphi}_x^n I_{yz} \\ \\ \tau_z^n - (I_y - I_x)\dot{\varphi}_x^n \dot{\varphi}_y^n + \dot{\varphi}_z^n \dot{\varphi}_x^n I_{zy} \\ \qquad - (\dot{\varphi}_x^2 - \dot{\varphi}_y^2)^n I_{xy} - \dot{\varphi}_z^n \dot{\varphi}_y^n I_{zx} \end{pmatrix} \tag{2.42}$$

Once the solution for $\dot{\boldsymbol{u}}^{n+1}$ and $\ddot{\boldsymbol{\varphi}}^{n+1}$ is found, the position, orientation and the velocities are updated with a second order Taylor method:

$$\begin{aligned} \boldsymbol{u}^{n+1} &= \boldsymbol{u}^n + \Delta t\, \dot{\boldsymbol{u}}^{n+1} \\ \boldsymbol{R}^T \boldsymbol{G}^{n+1} &= \boldsymbol{R}^T \boldsymbol{G}^n + \Delta t\, \boldsymbol{u}^n + \frac{\Delta t^2}{2} \dot{\boldsymbol{u}}^{n+1} \\ \dot{\boldsymbol{\varphi}}^{n+1} &= \dot{\boldsymbol{\varphi}}^n + \Delta t\, \ddot{\boldsymbol{\varphi}}^{n+1} \\ \boldsymbol{\varphi}^{n+1} &= \boldsymbol{\varphi}^n + \Delta t\, \dot{\boldsymbol{\varphi}}^n + \frac{\Delta t^2}{2} \ddot{\boldsymbol{\varphi}}^{n+1} \end{aligned} \tag{2.43}$$

Figure 2.14 shows a typical output of the ship dynamics based on the model presented in this section in combination with the 3D hydrodynamics shown in the subsequent section. The figure also highlights the freedom in the definition of the shape of the hull, i.e. it is possible to simulate the motion of reasonably complex objects.

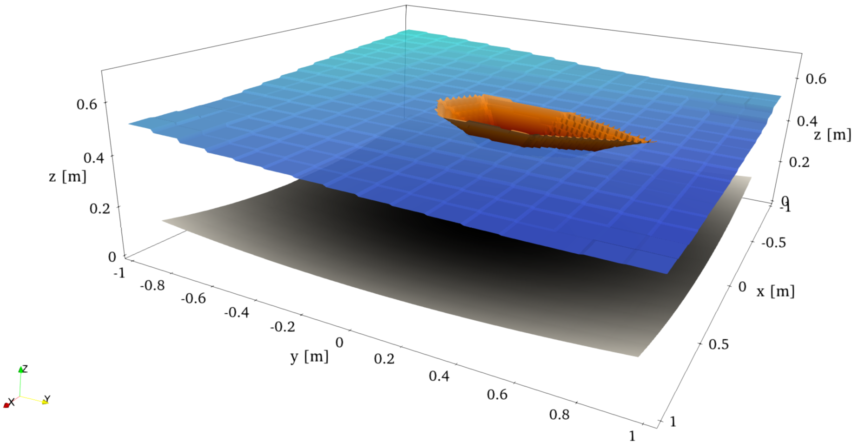


Figure 2.14. Representation of a ship floating on water in a 3D simulation.

2.7 3D model

The 3D model is a direct extension of the concepts explained in Section 2.5. The mass equation has two differences of fluxes for the x and y directions, and each flux is an integral over the vertical of the linear momentum. The momentum equations have all the advective terms, the pressure gradients and the viscosity along the vertical. As explained for the 2D$_{xz}$ model, the system is initially written in continuous form with primitive variables. The discretized form involving the fluxes q_x and q_y is obtained after integration of the momentum equations along the vertical.

$$\frac{\partial}{\partial t}\left[H(\eta)\right] + \frac{\partial}{\partial x}\left[\int_b^\eta u(x,y,z,t)dz\right] + \frac{\partial}{\partial y}\left[\int_b^\eta v(x,y,z,t)dz\right] = 0, \tag{2.44a}$$

$$\frac{\partial u}{\partial t} + \frac{\partial\,(uu)}{\partial x} + \frac{\partial\,(uv)}{\partial y} + \frac{\partial\,(uw)}{\partial z} + g\frac{\partial \eta}{\partial x} = +\frac{\partial}{\partial z}\left(\nu\frac{\partial u}{\partial z}\right), \tag{2.44b}$$

$$\frac{\partial v}{\partial t} + \frac{\partial\,(vu)}{\partial x} + \frac{\partial\,(vv)}{\partial y} + \frac{\partial\,(vw)}{\partial z} + g\frac{\partial \eta}{\partial y} = +\frac{\partial}{\partial z}\left(\nu\frac{\partial v}{\partial z}\right), \tag{2.44c}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \tag{2.44d}$$

The boundary conditions for the wind stress and the bottom friction act in the cells with a dry/wet or wet/air interface and read:

$$\begin{cases} \nu\dfrac{\partial u}{\partial z} = \gamma_T\left(u^a - u^s\right) & \text{at the free surface} \\ \nu\dfrac{\partial v}{\partial z} = \gamma_T\left(v^a - v^s\right) & \text{at the free surface} \\ \nu\dfrac{\partial u}{\partial z} = \gamma_B u^b & \text{at the bottom} \\ \nu\dfrac{\partial v}{\partial z} = \gamma_B v^b & \text{at the bottom} \end{cases} \tag{2.45}$$

with γ_T and γ_B defined as

$$\gamma_T = C_d\frac{\rho_a}{\rho}\sqrt{(u^a)^2 + (v^a)^2}, \qquad \gamma_B = \frac{g\sqrt{(u^b)^2 + (v^b)^2}}{k_{St}^2 R_h^{1/3}}. \tag{2.46}$$

The integrals are approximated similarly to equation (2.23), but, due to the use of a staggered grid, in 2D$_{xy}$/3D they are calculated at every interface along x and y (in total $(I_{max}+1)J_{max} + (J_{max}+1)I_{max}$ times):

$$\int_b^\eta u dz \approx \sum_k \Delta z_{i+\frac{1}{2},j,k} u_{i+\frac{1}{2},j,k}, \qquad \int_b^\eta v dz \approx \sum_k \Delta z_{i+\frac{1}{2},j,k} v_{i,j+\frac{1}{2},k}, \tag{2.47}$$

with $k = 1, 2, ..., K_{max}$, and the vertical spacings $\Delta z_{i+\frac{1}{2},j,k} = \Delta z$ and $\Delta z_{i,j+\frac{1}{2},k} = \Delta z$. The first and last layers are adjusted so that

$$\begin{aligned} \sum_k \Delta z_{i+\frac{1}{2},j,k} &= \max(0, \min(\eta_{i+\frac{1}{2},j} - b_{i+\frac{1}{2},j}, l_{i+\frac{1}{2},j} - b_{i+\frac{1}{2},j})), \\ \sum_k \Delta z_{i,j+\frac{1}{2},k} &= \max(0, \min(\eta_{i,j+\frac{1}{2}} - b_{i,j+\frac{1}{2}}, l_{i,j+\frac{1}{2}} - b_{i,j+\frac{1}{2}})). \end{aligned} \tag{2.48}$$

Consequently, it is possible to identify the partially wet cells along the vertical, marked with the start and end indeces $m_{i+\frac{1}{2},j}$, $m_{i,j+\frac{1}{2}}$, $M^n_{i+\frac{1}{2},j}$, $M^n_{i,j+\frac{1}{2}}$. They will impose the dimension of the arrays along the vertical.

The advective terms $\partial(q_x u)/\partial x$ and $\partial(q_y v)/\partial y$ are discretized with the numerical fluxes presented in Section 2.4.3. The advective term $\partial(q_x w)/\partial z$ is the same as in equations (2.25), with the addition of the missing j index. The new mixed terms $\partial(q_x v)/\partial y$, $\partial(q_y u)/\partial x$ and $\partial(q_y w)/\partial z$ follow again the central scheme:

$$\begin{aligned}\frac{\partial\left(q_x v\right)}{\partial y} \approx & \frac{1}{\Delta y}\left(\frac{\left(q_x\right)_{i+\frac{1}{2},j+1,k}^n+\left(q_x\right)_{i+\frac{1}{2},j,k}^n}{2}\right)\left(\frac{v_{i+1,j+\frac{1}{2},k}^n+v_{i,j+\frac{1}{2},k}^n}{2}\right) \\ & -\frac{1}{\Delta y}\left(\frac{\left(q_x\right)_{i+\frac{1}{2},j,k}^n+\left(q_x\right)_{i+\frac{1}{2},j-1,k}^n}{2}\right)\left(\frac{v_{i+1,j-\frac{1}{2},k}^n+v_{i,j-\frac{1}{2},k}^n}{2}\right) \quad ,\end{aligned} \tag{2.49}$$

$$\begin{aligned}\frac{\partial\left(q_y u\right)}{\partial x} \approx & \frac{1}{\Delta x}\left(\frac{\left(q_y\right)_{i+1,j+\frac{1}{2},k}^n+\left(q_y\right)_{i,j+\frac{1}{2},k}^n}{2}\right)\left(\frac{u_{i+\frac{1}{2},j+1,k}^n+u_{i+\frac{1}{2},j,k}^n}{2}\right) \\ & -\frac{1}{\Delta x}\left(\frac{\left(q_y\right)_{i,j+\frac{1}{2},k}^n+\left(q_y\right)_{i-1,j+\frac{1}{2},k}^n}{2}\right)\left(\frac{u_{i-\frac{1}{2},j+1,k}^n+u_{i-\frac{1}{2},j,k}^n}{2}\right) \quad ,\end{aligned} \tag{2.50}$$

$$\begin{aligned}\frac{\partial\left(q_y w\right)}{\partial z} \approx & \frac{1}{\Delta z_y}\left(\frac{\left(q_y\right)_{i,j+\frac{1}{2},k+1}^n+\left(q_y\right)_{i,j+\frac{1}{2},k}^n}{2}\right)\left(\frac{w_{i,j+1,k+\frac{1}{2}}^n+w_{i,j,k+\frac{1}{2}}^n}{2}\right) \\ & -\frac{1}{\Delta z_y}\left(\frac{\left(q_y\right)_{i,j+\frac{1}{2},k}^n+\left(q_y\right)_{i,j+\frac{1}{2},k-1}^n}{2}\right)\left(\frac{w_{i,j+1,k-\frac{1}{2}}^n+w_{i,j,k-\frac{1}{2}}^n}{2}\right) \quad .\end{aligned} \tag{2.51}$$

The time step limitation is set according to a CFL-type condition as follows:

$$\Delta t=\min\left(\Delta t_{max}\ ,\ \frac{\mathrm{CFL}}{2\left(\frac{|u|_{max}}{\Delta x}+\frac{|v|_{max}}{\Delta y}\right)+c}\right) \qquad \text{with CFL} = 0.9 \tag{2.52}$$

The viscous term along x is discretized exactly like in equation (2.28), but adding a j index everywhere; in the y direction:

$$\begin{aligned}\frac{\partial}{\partial z}\left(\Delta z \nu \frac{\partial v}{\partial z}\right) \approx \frac{1}{\Delta z_{i,j+\frac{1}{2},k}^n}\Bigg[& \nu_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1}\left(v_{i,j+\frac{1}{2},k+1}^{n+1}-v_{i,j+\frac{1}{2},k}^{n+1}\right) \\ & -\nu_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+1}\left(v_{i,j+\frac{1}{2},k}^{n+1}-v_{i,j+\frac{1}{2},k-1}^{n+1}\right)\Bigg].\end{aligned} \tag{2.53}$$

Defining the discrete volume in a cell as $V_{i,j}^n = \Delta x \Delta y H_{i,j}^n$ and using the shorthand notation $V_{i,j}^n = V(\eta_{i,j}^n, \mathbf{G}^n, \boldsymbol{\varphi}^n)$ the discretized form of system (2.44) is:

$$
\begin{aligned}
V_{i,j}^{n+1} = V_{i,j}^{n} - \Delta y \Delta t \left[\left(\boldsymbol{\Delta Z}^T \boldsymbol{U} \right)_{i+\frac{1}{2},j}^{n+1} - \left(\boldsymbol{\Delta Z}^T \boldsymbol{U} \right)_{i-\frac{1}{2},j}^{n+1} \right] \\
-\Delta x \Delta t \left[\left(\boldsymbol{\Delta Z}^T \boldsymbol{V} \right)_{i,j+\frac{1}{2}}^{n+1} - \left(\boldsymbol{\Delta Z}^T \boldsymbol{V} \right)_{i,j-\frac{1}{2}}^{n+1} \right],
\end{aligned} \tag{2.54a}
$$

$$
\boldsymbol{A}_{i+\frac{1}{2},j}^{n} \boldsymbol{U}_{i+\frac{1}{2},j}^{n+1} = \boldsymbol{E}_{i+\frac{1}{2},j}^{n} - g \frac{\Delta t}{\Delta x} (\eta_{i+1,j}^{n+1} - \eta_{i,j}^{n+1}) \boldsymbol{\Delta Z}_{i+\frac{1}{2},j}^{2}, \tag{2.54b}
$$

$$
\boldsymbol{A}_{i,j+\frac{1}{2}}^{n} \boldsymbol{V}_{i,j+\frac{1}{2}}^{n+1} = \boldsymbol{E}_{i,j+\frac{1}{2}}^{n} - g \frac{\Delta t}{\Delta y} (\eta_{i,j+1}^{n+1} - \eta_{i,j}^{n+1}) \boldsymbol{\Delta Z}_{i,j+\frac{1}{2}}^{2}. \tag{2.54c}
$$

For what concerns the momentum along x, the general definitions of the vectors are the same of (2.30) and (2.31), but now they must be computed for every j; for the vectors along y one simply substitutes all the q_x and u with q_y and v.

The substitution of the momentum equations in the mass equation leads to:

$$
\begin{aligned}
V_{i,j}^{n+1} - g \Delta y \frac{\Delta t^2}{\Delta x} \Bigg[\left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2 \right)_{i+\frac{1}{2},j}^{n} \left(\eta_{i+1,j}^{n+1} - \eta_{i,j}^{n+1} \right) \\
- \left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2 \right)_{i-\frac{1}{2},j}^{n} \left(\eta_{i,j}^{n+1} - \eta_{i-1,j}^{n+1} \right) \Bigg] \\
-g \Delta x \frac{\Delta t^2}{\Delta y} \Bigg[\left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2 \right)_{i,j+\frac{1}{2}}^{n} \left(\eta_{i,j+1}^{n+1} - \eta_{i,j}^{n+1} \right) \\
- \left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2 \right)_{i,j-\frac{1}{2}}^{n} \left(\eta_{i,j}^{n+1} - \eta_{i,j-1}^{n+1} \right) \Bigg] = r_{i,j}^{n},
\end{aligned} \tag{2.55}
$$

with

$$
\begin{aligned}
r_{i,j}^{n} = V_{i,j}^{n} - \Delta y \Delta t \Bigg[\left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E} \right)_{i+\frac{1}{2},j}^{n} - \left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E} \right)_{i-\frac{1}{2},j}^{n} \Bigg] \\
-\Delta x \Delta t \Bigg[\left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E} \right)_{i,j+\frac{1}{2}}^{n} - \left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E} \right)_{i,j-\frac{1}{2}}^{n} \Bigg].
\end{aligned} \tag{2.56}
$$

The resulting mildly nonlinear pressure system is again solved with the nested Newton method detailed in [35, 36]. Again the Thomas algorithm is used for computing all terms of the type $\boldsymbol{A}^{-1}\boldsymbol{\Delta Z}^2$ and $\boldsymbol{A}^{-1}\boldsymbol{E}$. The solution of the five-diagonal system is obtained applying a matrix-free conjugate gradient method; with $\boldsymbol{\eta}^{n+1}$, the horizontal velocities at the new time step are found from the momentum equations. Finally, the vertical velocities are computed from the incompressibility constraint. Starting from the bottom with $w^{n+1}_{i,j,m-\frac{1}{2}} = 0$, the others are obtained by

$$w^{n+1}_{i,j,k+\frac{1}{2}} = w^{n+1}_{i,j,k-\frac{1}{2}} - \frac{\Delta z^n_{i+\frac{1}{2},j,k} u^{n+1}_{i+\frac{1}{2},j,k} - \Delta z^n_{i-\frac{1}{2},j,k} u^{n+1}_{i-\frac{1}{2},j,k}}{\Delta x} - \frac{\Delta z^n_{i,j+\frac{1}{2},k} v^{n+1}_{i,j+\frac{1}{2},k} - \Delta z^n_{i,j-\frac{1}{2},k} v^{n+1}_{i,j-\frac{1}{2},k}}{\Delta y} \quad . \tag{2.57}$$

with $k = m_i, m_i + 1, ..., M^n_i - 1$.
The 3D model automatically reduces to all the previously presented models, setting appropriately $K_{max} = 1$ and/or $J_{max} = 1$.

2.8 Theta method

In this Section, the 3D discretization of Section 2.7 is extended with an application of an implicitness parameter ϑ [32], with the aim of increasing the time accuracy of the model. The method consists in taking a convex combination of explicit and implicit terms. For a generic variable a:

$$a^{n+\vartheta} = (1 - \vartheta)a^n + \vartheta a^{n+1}. \tag{2.58}$$

The implicitness parameter is applied to system (2.54), and in particular to the fluxes in the mass equation and the pressure gradient in the momentum equation, as shown in system (2.59):

$$V^{n+1}_{i,j} = V^n_{i,j} - \Delta y \Delta t \left[\left(\boldsymbol{\Delta Z}^T \boldsymbol{U} \right)^{n+\vartheta}_{i+\frac{1}{2},j} - \left(\boldsymbol{\Delta Z}^T \boldsymbol{U} \right)^{n+\vartheta}_{i-\frac{1}{2},j} \right] - \Delta x \Delta t \left[\left(\boldsymbol{\Delta Z}^T \boldsymbol{V} \right)^{n+\vartheta}_{i,j+\frac{1}{2}} - \left(\boldsymbol{\Delta Z}^T \boldsymbol{V} \right)^{n+\vartheta}_{i,j-\frac{1}{2}} \right], \tag{2.59a}$$

$$\boldsymbol{A}^n_{i+\frac{1}{2},j} \boldsymbol{U}^{n+1}_{i+\frac{1}{2},j} = \boldsymbol{E}^n_{i+\frac{1}{2},j} - g\frac{\Delta t}{\Delta x}(\eta^{n+\vartheta}_{i+1,j} - \eta^{n+\vartheta}_{i,j})\boldsymbol{\Delta Z}^2_{i+\frac{1}{2},j}, \tag{2.59b}$$

$$\boldsymbol{A}^n_{i,j+\frac{1}{2}} \boldsymbol{V}^{n+1}_{i,j+\frac{1}{2}} = \boldsymbol{E}^n_{i,j+\frac{1}{2}} - g\frac{\Delta t}{\Delta y}(\eta^{n+\vartheta}_{i,j+1} - \eta^{n+\vartheta}_{i,j})\boldsymbol{\Delta Z}^2_{i,j+\frac{1}{2}}. \tag{2.59c}$$

For stability, $\vartheta \geq \frac{1}{2}$. When $\vartheta = \frac{1}{2}$, the discretization can be seen as an application of the Crank-Nicolson method [81], which means that for what concerns the pressure gradient, the approximation is second-order accurate in time.
In the momentum equation, the new explicit pressure gradient term is incorporated in the operator of the known terms $\boldsymbol{E}^n$:

$$\boldsymbol{E}^n = \begin{pmatrix} \Delta z_M q_M^* + \Delta t \gamma_T \Delta z_M u_a - c_\vartheta \Delta z_M^2 \\ \Delta z_{M-1} q_{M-1}^* - c_\vartheta \Delta z_{M-1}^2 \\ \vdots \\ \Delta z_m q_m^* - c_\vartheta \Delta z_m^2. \end{pmatrix} \tag{2.60}$$

where $c_\vartheta = (1-\vartheta) g \Delta t (\eta_{i+1,j}^n - \eta_{i,j}^n)/\Delta x$ for the inferfaces along the x axis, while $c_\vartheta = (1-\vartheta) g \Delta t (\eta_{i,j+1}^n - \eta_{i,j}^n)/\Delta y$ for the inferfaces along the y axis. The vectors $\boldsymbol{A}^{-1}\boldsymbol{\Delta Z}^2$ and $\boldsymbol{A}^{-1}\boldsymbol{E}$ can still be computed with the Thomas algorithm. After the substitution of the expression of the unknown momentum into the implicit flux of the mass equation, equation (2.61) is obtained:

$$\begin{aligned} V_{i,j}^{n+1} - g\Delta y \frac{\Delta t^2}{\Delta x} \Bigg\{ & \left(\vartheta \boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2\right)_{i+\frac{1}{2},j}^n \left[(\vartheta\eta)_{i+1,j}^{n+1} - (\vartheta\eta)_{i,j}^{n+1}\right] \\ & - \left(\vartheta \boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2\right)_{i-\frac{1}{2},j}^n \left[(\vartheta\eta)_{i,j}^{n+1} - (\vartheta\eta)_{i-1,j}^{n+1}\right] \Bigg\} \\ -g\Delta x \frac{\Delta t^2}{\Delta y} \Bigg\{ & \left(\vartheta \boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2\right)_{i,j+\frac{1}{2}}^n \left[(\vartheta\eta)_{i,j+1}^{n+1} - (\vartheta\eta)_{i,j}^{n+1}\right] \\ & - \left(\vartheta \boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2\right)_{i,j-\frac{1}{2}}^n \left[(\vartheta\eta)_{i,j}^{n+1} - (\vartheta\eta)_{i,j-1}^{n+1}\right] \Bigg\} = r_{i,j}^n, \end{aligned} \tag{2.61}$$

with

$$
\begin{aligned}
r^n_{i,j} = V^n_{i,j} &- \Delta y \Delta t \Bigg[\left(\vartheta \boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E} \right)^n_{i+\frac{1}{2},j} - \left(\vartheta \boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E} \right)^n_{i-\frac{1}{2},j} \Bigg] \\
&- \Delta x \Delta t \Bigg[\left(\vartheta \boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E} \right)^n_{i,j+\frac{1}{2}} - \left(\vartheta \boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E} \right)^n_{i,j-\frac{1}{2}} \Bigg] \\
&- \Delta y \Delta t \Bigg\{ \Big[(1-\vartheta)\, \boldsymbol{\Delta Z}^T \boldsymbol{U} \Big]^n_{i+\frac{1}{2},j} - \Big[(1-\vartheta)\, \boldsymbol{\Delta Z}^T \boldsymbol{U} \Big]^n_{i-\frac{1}{2},j} \Bigg\} \\
&- \Delta x \Delta t \Bigg\{ \Big[(1-\vartheta)\, \boldsymbol{\Delta Z}^T \boldsymbol{V} \Big]^n_{i,j+\frac{1}{2}} - \Big[(1-\vartheta)\, \boldsymbol{\Delta Z}^T \boldsymbol{V} \Big]^n_{i,j-\frac{1}{2}} \Bigg\} \\
&+ g \Delta y \frac{\Delta t^2}{\Delta x} \Bigg\{ \left(\vartheta \boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2 \right)^n_{i+\frac{1}{2},j} \Big[((1-\vartheta)\,\eta)^n_{i+1,j} - ((1-\vartheta)\,\eta)^n_{i,j} \Big] \\
&\qquad - \left(\vartheta \boldsymbol{\Delta} Z^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2 \right)^n_{i-\frac{1}{2},j} \Big[((1-\vartheta)\,\eta)^n_{i,j} - ((1-\vartheta)\,\eta)^n_{i-1,j} \Big] \Bigg\} \\
&+ g \Delta x \frac{\Delta t^2}{\Delta y} \Bigg\{ \left(\vartheta \boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2 \right)^n_{i,j+\frac{1}{2}} \Big[((1-\vartheta)\,\eta)^n_{i,j+1} - ((1-\vartheta)\,\eta)^n_{i,j} \Big] \\
&\qquad - \left(\vartheta \boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z}^2 \right)^n_{i,j-\frac{1}{2}} \Big[((1-\vartheta)\,\eta)^n_{i,j} - ((1-\vartheta)\,\eta)^n_{i,j-1} \Big] \Bigg\}.
\end{aligned}
\tag{2.62}
$$

It should be noticed how by choosing $\vartheta = 1$, all the explicit terms added in this section disappear and equation (2.61) reverts to equation (2.55).
For the compatibility with the pressurized regions, the implicitness parameter is variable in space: at every interface of the ship region $\vartheta = 1$.

2.9 Preconditioning

For every cycle of the nested-Newton algorithm [36], a linearized system of the form $\boldsymbol{L\eta} = \boldsymbol{r}_L$ is solved by a matrix-free conjugate gradient system. $\boldsymbol{L}$ is a sparse symmetric matrix composed of five diagonals. The main diagonal is composed of elements labeled as L_C, while the four off-diagonals are composed by elements labeled as L_E, L_W, L_N and L_S, where the subscripts stand for east, west, north and south following the convention established in Figure 2.1. Due to the symmetry of $\boldsymbol{L}$, $(L_E)_{i,j} = (L_W)_{i+1,j}$ and $(L_N)_{i,j} = (L_S)_{i,j+1}$, where (i, j) are the indices of the computational cells along the x and y axes, not the row/column indices of the matrix.

The equation that needs to be solved for each cell is:

$$\begin{aligned}(L_C)_{i,j}\eta_{i,j}+&(L_E)_{i,j}\eta_{i+1,j}+(L_W)_{i,j}\eta_{i-1,j}\\ &+(L_N)_{i,j}\eta_{i,j+1}+(L_S)_{i,j}\eta_{i,j-1} &= (r_L)_{i,j}.\end{aligned} \tag{2.63}$$

To reach a faster convergence, the system is preconditioned with a diagonal scaling [65]. After rewriting every unknown term as an auxiliary variable $\tilde{\eta}_{i,j}=\sqrt{(L_C)_{i,j}}\,\eta_{i,j}$ and rearranging the coefficients, equation (2.64) is obtained:

$$\begin{aligned}\tilde{\eta}_{i,j}+&\frac{(L_E)_{i,j}}{\sqrt{(L_C)_{i,j}(L_C)_{i+1,j}}}\tilde{\eta}_{i+1,j}+\frac{(L_W)_{i,j}}{\sqrt{(L_C)_{i,j}(L_C)_{i-1,j}}}\tilde{\eta}_{i-1,j}\\ +&\frac{(L_N)_{i,j}}{\sqrt{(L_C)_{i,j}(L_C)_{i,j+1}}}\tilde{\eta}_{i,j+1}+\frac{(L_S)_{i,j}}{\sqrt{(L_C)_{i,j}(L_C)_{i,j-1}}}\tilde{\eta}_{i,j-1} &= \frac{(r_L)_{i,j}}{\sqrt{(L_C)_{i,j}}}.\end{aligned} \tag{2.64}$$

The preconditioned system can be summarized as $\tilde{\boldsymbol{L}}\tilde{\boldsymbol{\eta}}=\tilde{\boldsymbol{r}}_L$. Matrix $\tilde{\boldsymbol{L}}$ is still symmetric, positive definite and has a lower condition number than $\boldsymbol{L}$, so the system can be solved faster than before by a similar matrix-free conjugate-gradient method. Once convergence is reached, the original unknown is recovered from $\eta_{i,j}=\tilde{\eta}_{i,j}/\sqrt{(L_C)_{i,j}}$.

2.10 MUSCL-Hancock

The discretization of the advective term presented in Section 2.4.3 limits the space and time accuracy of the model to first order. A second-order approximation can be obtained by the application of the MUSCL-Hancock method [80, 116]. The idea is to consider the variables as piece-wise linear, instead of piece-wise constant, and base the computation of the advective terms on the values reconstructed at the cell center and evolved in time. In other terms, instead of computing the $f=q_x u$ terms as $f_{i,j}\left(\{q_x\}^n_{i+\frac{1}{2},j},\{q_x\}^n_{i-\frac{1}{2},j}\right)$, they are computed as:

$$f_{i,j}\left(\{q_x\}^{n+\frac{1}{2},+}_{i,j},\{q_x\}^{n+\frac{1}{2},-}_{i,j}\right), \tag{2.65}$$

with the + and - signs indicating where the reconstruction comes from. For example:

$$(q_x)^{n+\frac{1}{2},+}_{i,j}=(q_x)^n_{i+\frac{1}{2},j}-\frac{1}{2}\Delta\left(q_x\right)^x_{i+\frac{1}{2},j}+\frac{1}{2}\Delta t\partial_t\left(q_x\right)_{i+\frac{1}{2},j}. \tag{2.66}$$

The spatial-reconstruction term $\Delta\left(q_x\right)^x_{i+\frac{1}{2},j}$ is computed from the minmod slope limiter:

$$\Delta\left(q_x\right)^x_{i+\frac{1}{2},j} = \operatorname{minmod}\left(\{q_x\}^n_{i+\frac{3}{2},j} - \{q_x\}^n_{i+\frac{1}{2},j}, \{q_x\}^n_{i+\frac{1}{2},j} - \{q_x\}^n_{i-\frac{1}{2},j}\right), \tag{2.67}$$

$$\operatorname{minmod}\left(a, b\right) = \begin{cases} 0, & \text{if } ab \leq 0 \\ a, & \text{if } |a| < |b| \\ b, & \text{if } |a| \geq |b|. \end{cases} \tag{2.68}$$

The time derivative is obtained directly from the momentum equation $\partial_t\left(q_x\right) = -\partial_x\left(q_x u\right) - \partial_y\left(q_x v\right)$, which is discretized as a central finite difference:

$$\begin{aligned} \partial_t\left(q_x\right)_{i+\frac{1}{2},j} \approx &- \frac{\left[\left(q_x\right)^{n,-}_{i+1,j}\right]^2 / H^n_{i+1,j} - \left[\left(q_x\right)^{n,+}_{i,j}\right]^2 / H^n_{i,j}}{\Delta x} \\ &- \frac{\left[\left(q_x\right)^{n,-}_{i+\frac{1}{2},j+\frac{1}{2}} v^n_{i+\frac{1}{2},j+\frac{1}{2}}\right] - \left[\left(q_y\right)^{n,+}_{i+\frac{1}{2},j-\frac{1}{2}} v^n_{i+\frac{1}{2},j-\frac{1}{2}}\right]}{\Delta y}, \end{aligned} \tag{2.69}$$

where the relation $u = q_x/H$ was used and the variables at the known time are reconstructed using again equation (2.67):

$$\left(q_x\right)^{n,\pm}_{i,j} = \left(q_x\right)^n_{i\pm\frac{1}{2},j} \mp \frac{1}{2}\Delta\left(q_x\right)^x_{i\pm\frac{1}{2},j}, \tag{2.70}$$

$$\left(q_x\right)^{n,\pm}_{i+\frac{1}{2},j\mp\frac{1}{2}} = \left(q_x\right)^n_{i+\frac{1}{2},j} \mp \frac{1}{2}\Delta\left(q_x\right)^y_{i+\frac{1}{2},j}, \tag{2.71}$$

with

$$\Delta\left(q_x\right)^y_{i+\frac{1}{2},j} = \operatorname{minmod}\left(\{q_x\}^n_{i+\frac{1}{2},j+1} - \{q_x\}^n_{i+\frac{1}{2},j}, \{q_x\}^n_{i+\frac{1}{2},j} - \{q_x\}^n_{i+\frac{1}{2},j-1}\right), \tag{2.72}$$

$$v^n_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}\left(v^n_{i,j+\frac{1}{2}} + v^n_{i+1,j+\frac{1}{2}}\right). \tag{2.73}$$

Once the variables are reconstructed in space and evolved in time as in equation (2.66), the advective terms can be computed using the chosen numerical flux as in equation (2.65).

The equations are written here for the advective term in direction x; an analogue procedure can be repeated for the y direction.

2.11 Added-mass instability

One important feature of our model is the coupling of the ship and fluid dynamics, important for the ship buoyancy and the generation of waves. The PDE system needs the upper bound on the water depth, which is computed from the hull position. The ODE system for the ship motion requires the computation of the integral of pressure on the hull, which includes implicitly the added-mass force. This integral is computed explicitly. For a model of this type, Lee [79] shows that when the added mass is greater that the mass of the ship, the solution becomes unconditionally unstable. This usually happens when the ship has a large waterplane area and it floats in shallow waters. We emphasize that this condition is present even if the body is only allowed to heave vertically and it starts from hydrostatic equilibrium.

For our problem, the instability manifests as high-frequency oscillations of η in the ship region. In the central part of the ship η goes below the hull, which means that in a 1D simulation the ship region has pressurized parts separated by a non-pressurized part. An example of this instability is shown in Figure 2.15, where multiple time iterations have been overlapped.

Interestingly, this oscillating force results in a decent prediction of the vertical motion for a sufficiently small timestep. The ship remains close to the expected equilibrium position while vibrating endlessly, as seen in Figure 2.18. For bigger timesteps, the ship bounces on the free-surface.

A solution for this problem is the reconfiguration technique of Lee [79], which consists in the addition of an acceleration term on both sides of the ODE, as seen in equation (2.74). They have the same coefficient m_{psa}, called pseudo-added mass, but the acceleration on one side is treated as implicit, while the other as explicit. Equation (2.75) is the stability condition, which is respected when the pseudo-added mass is close to the added mass. As an estimate for the added mass we use $\tilde{m}_a = \rho \Delta y L_s^3/(12\bar{H})$, which is an approximation of the 1D integral derived by Lannes [78]. $\bar{H}$ is the average water depth in the ship region. Then, by taking $m_{psa} = \tilde{m}_a$, we solve the ODE.

$$(m + m_{psa})\dot{w}_G = mg + \int_{hull} p n_z \, dS + m_{psa}\dot{w}_G \tag{2.74}$$

$$\left| \frac{m_a - m_{psa}}{m + m_{psa}} \right| < 1 \tag{2.75}$$

With the application of the reconfiguration technique, shown in Figures 2.16 and 2.17, the added-mass instablity is kept well under control, the pressure η does not oscillate, and with it the upward force. The elevation of the center of mass

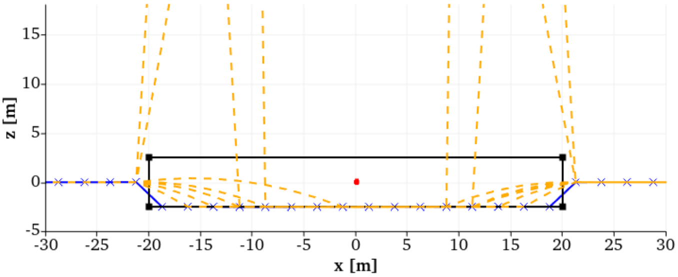


Figure 2.15. Instability due to $m_a > m_s$ for a floating body in its equilibrium position. The yellow-dashed lines are η at different time iterations.

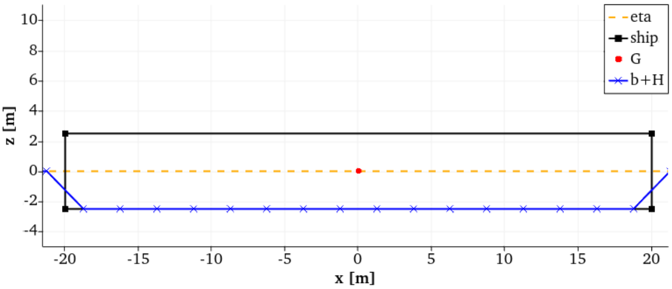


Figure 2.16. Correction of the added-mass instability with the reconfiguration method of Lee [79]. The pressure η in the ship domain is stable and at the static-equilibrium condition it has the expected value.

follows the predicted solution without sudden jumps around the equilibrium position, as seen in Figure 2.19.

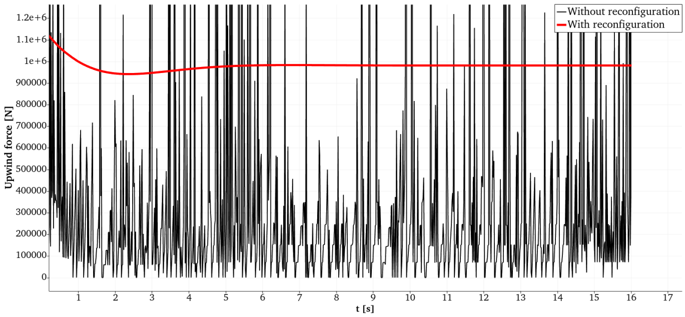
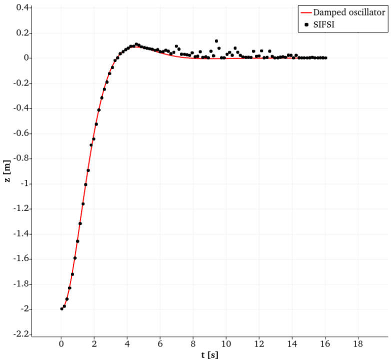


Figure 2.17. Upward force on the hull. The application of the reconfiguraton technique stabilizes the force.



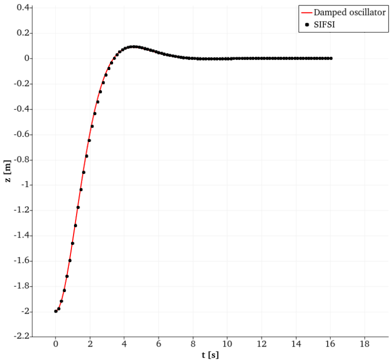


Figure 2.18. Centre of mass vertical position (w/o reconf.)

Figure 2.19. Centre of mass vertical position (with reconf.)

2.12 Non-hydrostatic pressure

In this Section, dispersive effects are introduced in the model through a Boussinesq-type term in the momentum equation.

Weakly nonlinear Boussinesq-type models are based on a couple of assumptions: the first, called the small amplitude hypothesis, assumes that the ratio of the wave amplitude to the water depth $\epsilon_w = A_w/H \ll 1$. The second concerns the wavelength and restricts the application to long waves, i.e. $\sigma_w = H/L_w \ll 1$. These conditions are respected by solitary waves.

The most simple Boussinesq-type model is the linearized Peregrine-Abbott model [1, 57, 72, 93], which introduces an elliptic operator in the momentum equation, as shown for the 1D case in equation (2.76). The mass equation is not modified.

$$\left[1 - \partial_x \left(\frac{H^2}{3}\partial_x \cdot\right)\right] \partial_t(q) + \partial_x(qu) + gH\partial_x\eta = 0. \tag{2.76}$$

The implementation is performed as follows: the first step is solving the hydrostatic system (2.77), without the additional high-order term. The unknowns of this system are called $\boldsymbol{\eta}^{hyd}$ and $\boldsymbol{q}^{hyd}$.

$$\begin{aligned} H_i(\eta_i^{hyd}, x_G^{n+1}, z_G^{n+1}, \varphi^{n+1}) &= H(\eta_i^n, x_G^n, z_G^n, \varphi^n) - \frac{\Delta t}{\Delta x}\left(q_{i+\frac{1}{2}}^{hyd} - q_{i-\frac{1}{2}}^{hyd}\right), \\ q_{i+\frac{1}{2}}^{hyd} &= q_{i+\frac{1}{2}}^{*} - \frac{\Delta t}{\Delta x} g H_{i+\frac{1}{2}}^n \left(\eta_{i+1}^{n+1} - \eta_i^{n+1}\right) - \Delta t \gamma_{i+\frac{1}{2}}^n q_{i+\frac{1}{2}}^{n+1}. \end{aligned} \tag{2.77}$$

The second step of the algorithm is the nonhydrostatic update. The water depth at the next time iteration is taken equal to the hydrostatic water depth, as in equation (2.78). The discharge $\boldsymbol{q}^{n+1}$ is updated through the application of the elliptic operator.

$$H_i(\eta_i^{n+1}, x_G^{n+1}, z_G^{n+1}, \varphi^{n+1}) = H(\eta_i^{hyd}, x_G^{n+1}, z_G^{n+1}, \varphi^{n+1}), \tag{2.78}$$

$$\begin{aligned} q_{i+\frac{1}{2}}^{n+1} - \frac{1}{3\Delta x^2}\left\{\left[(H_{i+1}^n)^2\left(q_{i+\frac{3}{2}}^{n+1} - q_{i+\frac{1}{2}}^{n+1}\right) - (H_i^n)^2\left(q_{i+\frac{1}{2}}^{n+1} - q_{i-\frac{1}{2}}^{n+1}\right)\right]\right\} &= \\ q_{i+\frac{1}{2}}^{hyd} - \frac{1}{3\Delta x^2}\left\{\left[(H_{i+1}^n)^2\left(q_{i+\frac{3}{2}}^{n} - q_{i+\frac{1}{2}}^{n}\right) - (H_i^n)^2\left(q_{i+\frac{1}{2}}^{n} - q_{i-\frac{1}{2}}^{n}\right)\right]\right\} &. \end{aligned} \tag{2.79}$$

Equation (2.79) is a tridiagonal system that can be readily solved by the Thomas algorithm.

2.13 Horizontal motion

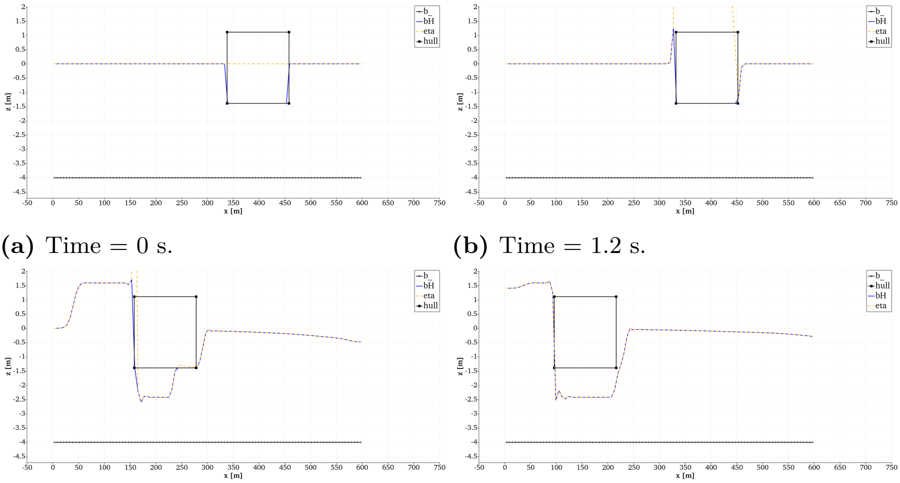
2.13.1 Starting problem

Figure 2.20 shows an example of a configuration which can be hard to solve: a ship with vertical walls moving horizontally at a constant speed in an initially-still water body. Also, the simulation is one-dimensional, which can be thought in the real world as a box-like object in a perfectly fitted canal with negligible lateral friction. This geometry implies that in the orthogonal horizontal direction the hull elevation is constant, so in the y direction one computational cell is enough.

Having vertical walls means that there is a sharp transition between part of

the domain governed by elliptical equations, the ship region, and the part govern by hyperbolic equations, the free-surface region. It also means that the volume of water pushed during the motion V_s will be confined in a small area, corresponding most likely to one computational cell, for a uniform grid.
The constant speed poses a problem especially in the first time iterations, because without a velocity field around the ship, V_s must be displaced by a strong gradient of pressure.
The one dimensional setup forces the existence of only two points of transition between the ship and free-surface regions, so the displaced volume has a very limited set of cells where the volume V_s can be pushed.
In this configuration, strong pressure oscillations forms at the bow when the ship enters a cell that is not pressurized. This oscillation is numerical in nature and it is connected to the localized change of equations from hyperbolic to elliptic and to the new constraints, which are a maximum volume allowed in the cell and a constant discharge in the ship region. The pressure oscillation is responsible for pushing the volume V_s out of the cell and for imposing the compatible gradient of flow velocity.
The pressure gradient can be so strong that in the ship region η goes below the hull, meaning that some cells become free-surface cells. Unfortunately, this effect is usually exacerbated by a time-step refinement because the volume V_s is fixed, so the required $\frac{\partial}{\partial t}\left[H(\eta)\right]$ increases.

In Figure 2.21, a grid convergence study is shown in the same setup of Figure 2.20. It is clear how with a smaller cell size the region of void below the ship shrinks, which is a consequence of smaller pressure oscillations.



(a) Time = 0 s.

(b) Time = 1.2 s.

(c) Time = 36 s.

(d) Time = 48.4 s.

Figure 2.20. Example of the oscillation-driven void generation during the horizontal motion. Setup: $\Delta x = 6$ m, no subgrid, ship at constant speed of 5 m/s, $\Delta t = 0.2$ s.

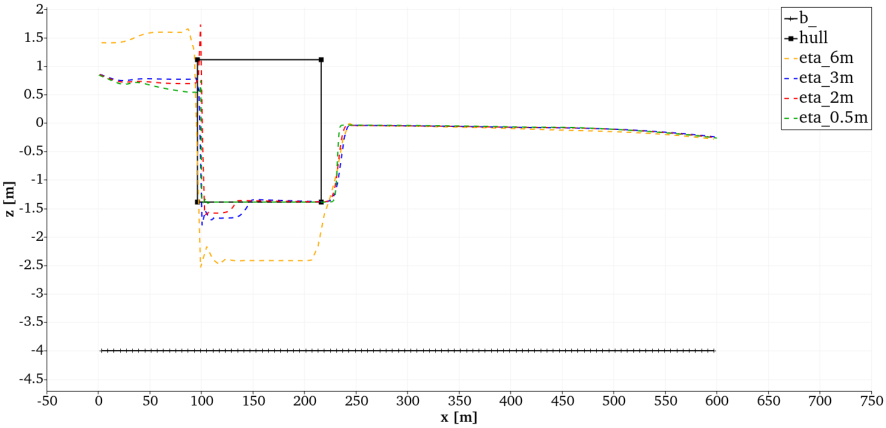


Figure 2.21. Grid convergence of the initial case of Figure 2.20. The pressure oscillations get smaller with a refinement of the computational mesh.

2.13.1.1 The origin of the pressure oscillations

In the ship region the PDE is elliptic, while outside it is hyperbolic. The term that causes this transition is the time derivative $\partial\left[H(\eta)\right]/\partial t$, which becomes a known term when η is above the hull. For example, the special case where it is exactly zero is when the cell is pressurized at the current and next time iteration, and the hull elevation in that cell does not change.
In free-surface cells, system (2.55) has to be solved. On the left side we have the future volume of water in the cell and the terms related to the pressure gradient; on the right side we have the current volume of water and advection. There are two mechanisms for the solution of the system, and they work in the same direction. For example, an increase in the unknown η^{n+1} in a cell increases both the water inside the cell and the pressure gradients. However, if we do a scale analysis of the system of equations we can see that the former mechanism is usually stronger than the other.
We want to rewrite equation (2.55) in a simplified way, so that we can estimate the order of magnitude of each term.
The volume terms are equal to the cell area multiplied by the water height, which incorporates in a single value all the space variability within the cell.

$$V_{i,j}^{n+1} = \Delta x \Delta y (H + \Delta H) \quad \text{and} \quad V_{i,j}^{n} = \Delta x \Delta y H. \tag{2.80}$$

If we neglect viscosity, bottom friction and wind friction effects, which have a secondary influence, the term $\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z^2}$ is equal to the water depth, so:

$$g\Delta y \frac{\Delta t^2}{\Delta x}\left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{\Delta Z^2}\right)_{i+\frac{1}{2},j}^{n}\left(\eta_{i+1,j}^{n+1} - \eta_{i,j}^{n+1}\right) \approx g\Delta y \frac{\Delta t^2}{\Delta x} H \Delta\eta. \tag{2.81}$$

Of the four pressure gradients, here we assume that only one is dominant in terms of order of magnitude, so that we can neglect the other three.
Neglecting again the viscosity, bottom friction and wind friction effects, the convective term $\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E}$ is composed of the old discharge and the advection terms.

$$\Delta y \Delta t \left(\boldsymbol{\Delta Z}^T \boldsymbol{A}^{-1} \boldsymbol{E}\right)_{i+\frac{1}{2},j}^{n} \approx \Delta y \Delta t \left[H u + \frac{\Delta t}{\Delta x}\Delta\left(H u^2\right)\right]. \tag{2.82}$$

Combining 2.80, 2.81 and 2.82, we obtain a simplified form of equation 2.55:

$$\Delta x \Delta y (H + \Delta H) + g\Delta y \frac{\Delta t^2}{\Delta x} H \Delta\eta \approx \Delta x \Delta y H + \Delta y \Delta t H \Delta u + \Delta y \frac{\Delta t^2}{\Delta x}\Delta\left(H u^2\right). \tag{2.83}$$

In non-pressurized cells, the variation in cell volume is unknown because it is proportional to the variation of the free-surface elevation, i.e. $\Delta H = \Delta\eta$:

$$\left(\Delta x\Delta y + g\Delta y\frac{\Delta t^2}{\Delta x}H\right)\Delta\eta \approx \Delta y\Delta tH\Delta u + \Delta y\frac{\Delta t^2}{\Delta x}\Delta\left(Hu^2\right). \tag{2.84}$$

We assume $\Delta x = \Delta y$. For an application at mesoscale, the orders of magnitude that we can expect are $1 < \Delta x < 10$ meters, $10^{-3} < \Delta t < 1$ seconds, $1 < H < 10$ meters, $0 < \Delta u < 1$ meters per second. Thus, the volume coefficient is of order $1 < \Delta x\Delta y < 100$ m^2, while the pressure coefficient is of order $10^{-5} < g\Delta y(\Delta t^2/\Delta x)H < 100$ m^2. Most likely, at the left-hand side the contribution of the water volume term is bigger than the contribution of the pressure gradient term. The advection terms are both of order $0 < \Delta y\Delta tH\Delta u < 100$ m^3 and $0 < \Delta y\left(\Delta t^2/\Delta x\right)\Delta\left(Hu^2\right) < 100$ m^3. In non-pressurized cells, the expected pressure difference is in the order of $0 < \Delta\eta < 1$ meters.

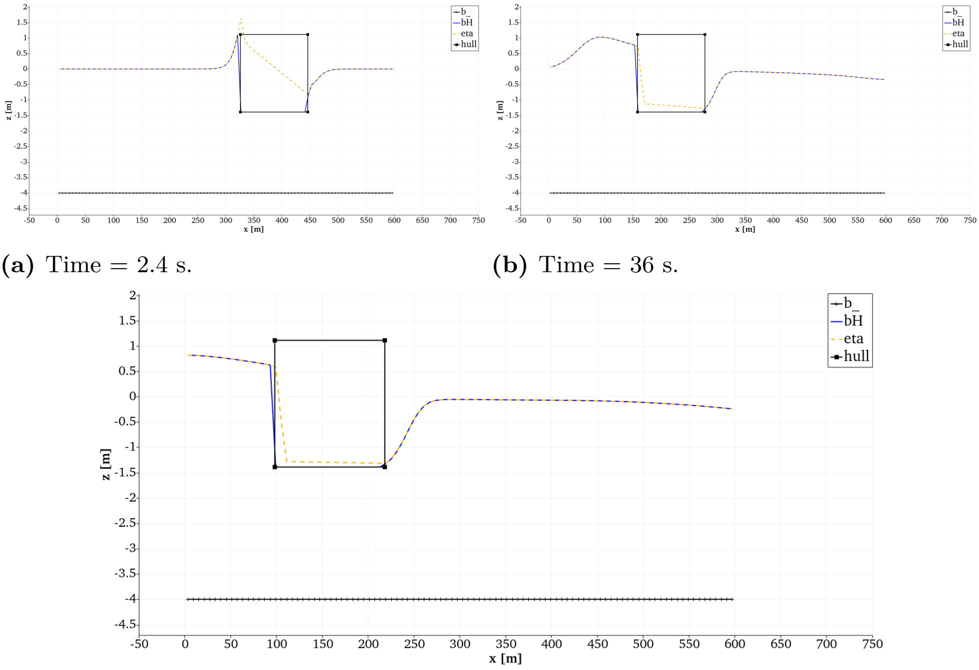
In fully-pressurized cells, at the left-hand side only the pressure gradient is responsible for the solution of the system, since the relationship $\Delta H = \Delta\eta$ is no longer valid. Also, the system has to converge to a configuration that displaces a predetermined amount of water volume $\Delta x\Delta y\Delta H$, with $10^{-2} < \Delta H < 1$ meters. In transitioning cells ΔH is the ship draft, while in already-pressurized cells it is a function of the vertical motion and the ship rotation.
If we repeat the order-of-magnitude analysis for the pressurized-cell equation, we obtain equation (2.85):

$$g\Delta y\frac{\Delta t^2}{\Delta x}H\Delta\eta \approx \Delta x\Delta y\Delta H + \Delta y\Delta tH\Delta u + \Delta y\frac{\Delta t^2}{\Delta x}\Delta\left(Hu^2\right). \tag{2.85}$$

With no relationship between ΔH and Δt, the leading known term in transitioning cells is usually $\Delta x\Delta y\Delta H$, making $\Delta\eta$ blow up easily, with an expected value of $0 < \Delta\eta < 10^7$ meters. Pushing water out of the cell necessitate of strong pressure gradients.
With the establishment of a proportionality between ΔH and Δt through the smooth-movement modification of Section 2.13.3, the ship displaces the water gradually, in smaller chunks.

2.13.1.2 Motion at gridspeed

It is worth noticing that the oscillation effect minimizes when the relationship $u_s\Delta t = k\Delta x$ holds, with $k = 1, 2, 3, \ldots$. In this case, the pressure η always



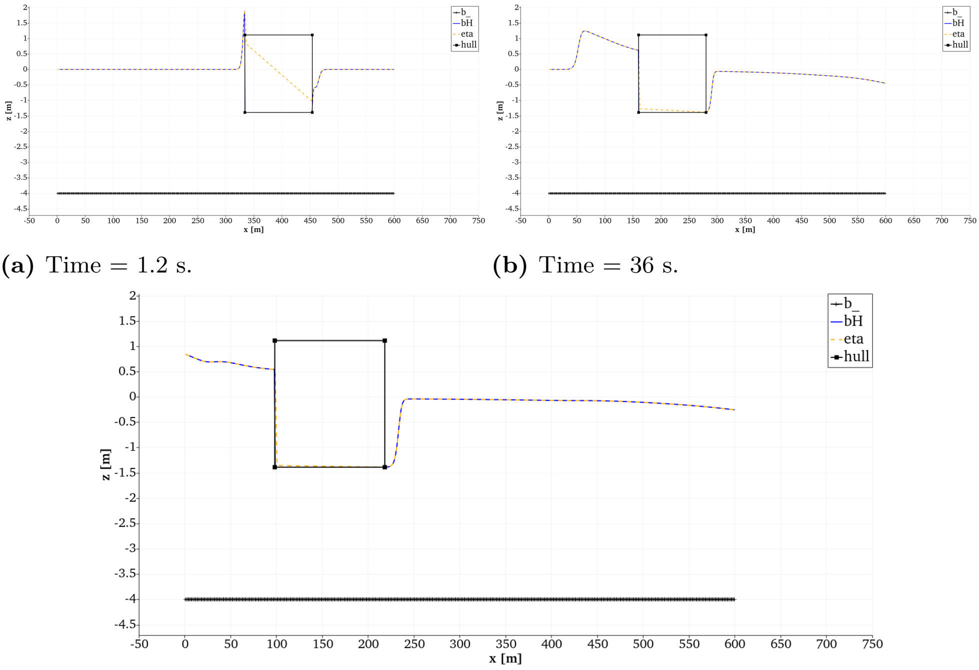
(a) Time = 2.4 s.

(b) Time = 36 s.

(c) Time = 48 s.

Figure 2.22. Horizontal motion at gridspeed, where the oscillation-driven void generation is suppressed. Coarse space-time discretization. Setup: $\Delta x = 6$ m, no subgrid, ship at constant speed of 5 m/s, $\Delta t = 1.2$ s.

has a shape characterized by a sudden drop at the bow, followed by a linear decrease which connects to the free-surface at the stern. The pressure jump at the bow increases in time until it reaches the hull, where it stabilizes. From here on, we take $k = 1$ when we are talking about the gridspeed, so $\Delta x/\Delta t = u_s$. Figures 2.23 and 2.22 illustrate the time evolution when the grid speed is equal to the ship speed. In the former case the relationship holds because the timestep size is bigger than the base case of Figure 2.20, while in the latter it holds because the grid size is smaller; this to show that this behavior is tightly related to the ratio $\Delta x/\Delta t$, and not to the two variables independently.



(a) Time = 1.2 s.

(b) Time = 36 s.

(c) Time = 48.4 s.

Figure 2.23. Horizontal motion at gridspeed, where the oscillation-driven void generation is suppressed. Fine space-time discretization. Setup: $\Delta x = 1$ m, no subgrid, ship at constant speed of 5 m/s, $\Delta t = 0.2$ s.

2.13.2 Galilean invariance setup

For completeness it is worth checking the Galilean invariance of the model, in order to discriminate whether the oscillation effect manifests only because there is a transition between different domain types, or because the domains move relatively to each other. To test this, a 1D case with a ship moving horizontally at a fixed speed is compared to a 1D case where the ship is fixed and the water flows below her with the same relative speed but with opposing direction. At the continuous level, the two setups are equivalent, due to the Galilean invariance of Newtonian mechanics. However, this is not completely true here due to the discretization.

If the pressure oscillation requires a relative motion, it is of interest testing if the waves generated by the ship-water interaction in the moving-ship case converge to the fixed-ship waves when such oscillation is minimized, which happens in the particular case when $u_s = \Delta x / \Delta t$. That would justify the efforts of applying techniques to suppress the oscillations of η in the general case.

Figure 2.24 shows the comparison of η in a case with moving ship in still water and fixed ship in flowing water. The snapshots are coordinated so that the same time has passed when they are in the same position. In this configuration $u_s \neq \Delta x/\Delta t$. We observe that while the moving-ship case has strong pressure oscillations at the front and creates a depression, the fixed-ship case is stable and the pressure converges monotonically to the hull elevation. The stern waves are mostly comparable, while the bow waves differ of the amount of water that came from the ship region. This test case seems to confirm that the oscillation effect disappears for immobile objects, regardless of the grid speed. This would explain why this model was successfully applied for drainage systems [41], where the pressurized regions are fixed pipes.

In Figure 2.25, two cases show that the moving-ship solution converges to its own Galilean-invariant solution when $u_s = \Delta x/\Delta t$. These simulations are two variations of the case shown in Figure 2.24. They both respect the grid-speed relationship, one by increasing Δt, the other by reducing Δx. As said in Section 2.13.1, the gridspeed configuration eliminates the oscillation-driven error. We observe a stable η also in the moving-ship case, and a good match of said η with its counterpart in the fixed configuration.
As we expect due to space-time discretization errors, the waves shape changes with different mesh sizes. In the case where the ship moves with gridspeed on a coarse mesh, where the space-time discretization errors are bigger, the solution differs slightly from the Galilean-invariant counterpart. In the refined motion at gridspeed case, they match perfectly. This is a confirmation that the two frames of reference converge to the same solution, except for the discretization errors. This provides us with a reference solution that we aim to when we study moving objects.

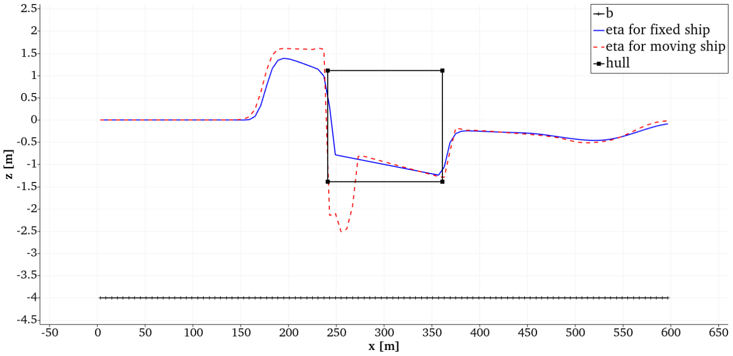


Figure 2.24. Galilean invariance: the red-dashed line refers to a simulation in which the ship is moving left in still water at a steady speed of 5 m/s, while the blue line refers to the case in which the ship is fixed and the water is flowing with an opposite relative speed. In the fixed-ship simulation, the pressure is not oscillating. Setup: $\Delta x = 6$ m, no subgrid, $\Delta t = 0.2$ s.

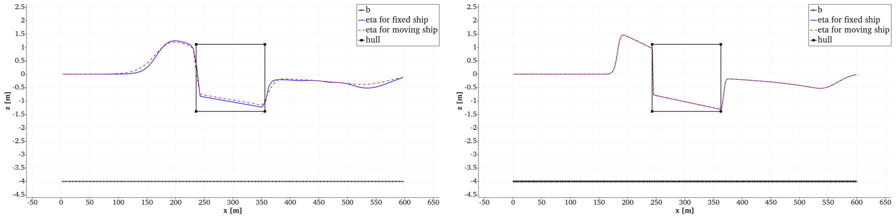


Figure 2.25. Simulations that respect the $u_s = \Delta x/\Delta t$ relationship. Comparison of fixed-ship and moving-ship results. In the left image, $\Delta x = 6$ m, $\Delta t = 1.2$ s. In the right image, $\Delta x = 1$ m, $\Delta t = 0.2$ s .

2.13.3 Smooth ship movement

As mentioned in 2.13.1, the magnitude of the pressure oscillation is proportional to the $\partial\left[H(\eta)\right]/\partial t$ term. Let us analyze what affects this term.

The depth variation ΔH in wet cells outside the ship region is an unknown, since H is proportional to η. Inside the ship region, ΔH depends entirely on the ship shape and its current and next positions. The floating objects we deal with have shapes smooth enough that the variation of the upper bound of H is limited. The position changes based on the ship's linear and angular velocity, acceleration and the timestep; the vertical component of the displacement Δz_v is proportional to Δt and is governed by the floating condition. For our applications, since the timestep is small enough, it is in the order of magnitude of a few millimeters or centimeters. The horizontal component depends on the ship's speed and shape, both input parameters but with the largest capability for creating big ΔH, especially at the transition points and for objects with vertical sides. Indeed, in those cells that are becoming pressurized, ΔH is the difference between the current waterline elevation and the elevation of the entering hull at the subcell centre. This difference is independent from the timestep, except for the minor contribution of Δz_v. It maximizes in the presence of vertical sides and it can be in the order of a few meters.

Since the timestep-independent component of ΔH is much bigger than the timestep-dependent counterpart, the overall effect of a reduction of the discretized timestep Δt is an increase of $\Delta H/\Delta t$. To minimize the oscillation effect, the timestep should be as big as possible. Δt has an upper bound imposed by the mild CFL condition, so this is already the optimal choice, unless we need it smaller to get the physics of the problem correctly.

To get more stable results we need ways of reducing $\Delta H/\Delta t$. The implementation utilized until now has a major disadvantage, which is that it considers the ship as inside of a cell only when its vertical projection includes the cell centre. Once it is inside, the elevation of the hull is computed and used as new upper bound of the water depth in the cell. The transition of the cell to the pressurized state is sudden, concentrated in a single time iteration even when $\Delta t\, u_s < \Delta x$. Consequently, a reduction in Δt always causes an increase in the oscillation.

To improve this implementation, we need to make the $\Delta H/\Delta t$ term appear every time iteration, based on the ship displacement. This split of the displaced water volume in multiple time iterations is beneficial for the stability of η, but it is also more physically correct. We notice that the subgrid was already helpful in smoothing the transition and creating partially-pressurized cells, since the passage of the ship was detected in each subcell. With no subgrid, the oscillations are more pronounced. If the subgrid is uniform, refining the subgrid only to leverage this effect is not recommended, as it would require an excessive use of computational resources outside the ship region. A possible approach could be using a dedicated refined subgrid for the ship region, but we believe a better solution is to compute the exact displaced volume based on the reconstructed ship. To do that, we first have to rethink the ship-reconstruction process by switching the order of operations. Before, we positioned the set of input-ship points in the new position of the domain space, and reconstructed the ship over the subset of subgrid cells inscribed by the ship vertical projection. The time variation of the subgrid-cells subset is responsible for the sudden ship dematerialization and materialization.

We propose instead the following procedure: for each cell, the water volume V_w^n is computed from the known η^n, bottom and hull. The new ship position and rotation is computed from the solution of the discretized ODE system; the input ship is rotated based on the angles at the new time and its center of mass is placed always in the same horizontal position, for example the midpoint of the domain. Then, the reconstruction generates a numerical hull, which is composed of piecewise-constant rectangles. The reconstructed ship is translated horizontally in the correct position, which have been determined from the ODE system solution. For each cell and each nested-Newton iteration, the water volume V_w^{n+1} is computed based on η, the bottom and the new hull.

One important detail is missing, which is how to compute the water volume in each subcell. With this new procedure, the numerical ship will not be aligned with the domain subgrid, in general. The vertical projection of the piecewise-constant hull will divide each subcell in different parts, each one with a different area and hull elevation. The water depth, obtained as the difference of the two functions H_1 and H_2, must consider this additional space variability of the hull inside the subcell. The derivative $\partial H_2/\partial \eta$ will be affected, as well, while $\partial H_1/\partial \eta$ remains equal to (2.2).

$$\begin{aligned} H_{i,j}(\eta_{i,j}, \boldsymbol{G}, \boldsymbol{\varphi}) = H_1 - H_2 =& \frac{1}{s}\sum_s \max\left(0, \eta_{i,j} - b_{i,j,s}\right) \\ & - \frac{1}{\Delta x \Delta y}\sum_s \sum_p \max\left(0, \eta_{i,j} - l_{i,j,s,p}\right) \cdot a_{i,j,s,p} \end{aligned} \tag{2.86}$$

$$\sum_p a_{i,j,s,p} = \Delta x_s \Delta y_s \qquad \forall i, j, s \tag{2.87}$$

$$\left(\frac{\partial H_2}{\partial \eta}\right)_{i,j} = \frac{1}{\Delta x \Delta y}\sum_s \sum_p H'_p \cdot a_{i,j,s,p}, \qquad \text{with} \qquad H'_p = \begin{cases} 1 & \text{if } \eta_{i,j} > l_{i,j,s,p} \\ 0 & \text{if } \eta_{i,j} \leq l_{i,j,s,p} \end{cases} \tag{2.88}$$

With p the number of parts the cell is divided into by the overlapping of the domain and ship grids.

During the reconstruction process, it is fundamental to maintain the same relative distance between the centre of mass and the subgrid at all time iterations, which can be seen as doing the reconstruction on a auxiliary grid where the object is not moving. Indeed, this allows for a volume computation that is independent from the domain grid and subgrid; thus, we can include the displaced-water volume in the mass equation at each time iteration, independent of the displacement.

In absence of rotations, this procedure generates always the same shape, so

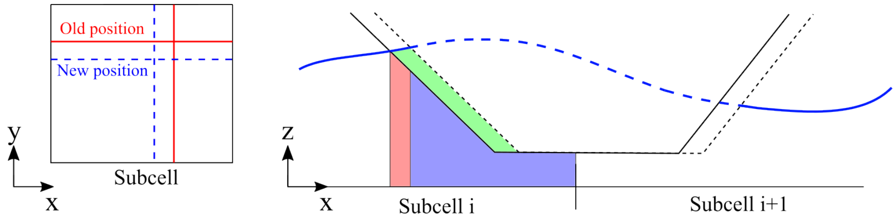


Figure 2.26. Illustration of the smooth-movement feature. On the left, the top view of a subcell and its subdivision in four zones, each with its own area and upper bound (the hull elevation). On the right, a side view of the computation process of the displaced volumes; if the ship is moving leftwards, the green part is the volume that has to be displaced, while the red part is the water that transitions to the pressurized state.

two ships at different time iterations will differ only by a vertical translation, on the auxiliary grid. This observation allows to optimize the implementation for the translation-only case and avoid doing the reconstruction procedure at all time iterations.
Nothing forbids using a ship-reconstruction grid different from the domain grid. For example, we could choose a grid with a smaller size, or a different orientation, or a different shape, but any of this choices would require a higher computational effort and complexity. In our work, we consider as accurate enough a grid that is equivalent to the subgrid, meaning it is uniform and of size $\Delta x_s \Delta y_s$. With this choice, each domain subcell is divided in up to four parts and the area-size computation is easy because the lines are parallel. On another note, let us observe that the depth function H_2 could involve the input shape directly, to maximize the accuracy of the volume computations. However, we believe that for our case the increase in accuracy is not worth the effort of dealing with high-resolution 3D objects, especially when the input shape is complex and/or curvilinear. It is much more convenient to use reconstructed shapes.

With the application of this procedure, the volume displacement caused by the horizontal motion will be proportional to Δt and independent of the grid or subgrid choice. This means that choosing a smaller Δt_{max} will not cause a sharp increase in the pressure oscillation, as would happen otherwise.

2.13.3.1 Smooth movement vs subgrid

It is of some importance to discuss how the use of a subgrid and the smooth-movement algorithm are related and what are their differences. Both methods are a way to compute the volumes and their time variation more accurately.

They both allow the existence of partially wet cells. In our implementation, the subgrid does its job at the wet/dry interface and at the wet/pressurized interface, while the smooth-movement algorithm works only for the wet/pressurized interface.

They can coexist in one simulation, and a particular choice of cell parameters makes their interaction simple and computationally efficient: if the ship domain cells have the same dimensions of the subcells, we are guaranteed that each subcell needs only four couples of area and elevation for the smooth-movement algorithm to work, and figuring out how they overlap does not involve special cases.

The subgrid is present in the entire domain, while the smooth-movement algorithm does computations only on the ship domain. The subgrid is static, it is set up only once at the initial time. The smooth-movement algorithm is dynamic, the set of areas and elevations changes every time the ship moves. The subgrid works in discrete uniform chunks, it can detect the ship only once it goes above the subcell's center. The smooth-movement algorithm detects the motion of the ship at each time iteration, in a more continuous way. This means that when they are used together, the subcell-dimensions' choice is independent from the motion of the ship, as now a fine subgrid is not critical to capture it.

2.13.4 Roof relaxation

In Section 2.13.1.1, it was showed that pressurized cells are more sensible to variations in the water volume or velocity. At the same time, transitioning cells are usually those where the biggest displacement of water is required. Together, these two conditions generate the strong oscillations of η. One possible numerical solution is to use a pseudo-compressibility method, as was done by Godlewski et al. [63]. The depth function in the pressurized region is artificially modified so that the water depth partially grows with the pressure. In our model, it corresponds to multiply H_2 by a relaxation factor $0 \leq (1 - r_r) \leq 1$, where with $r_r = 0$ the relaxation is not applied, while with $r_r = 1$ the ship becomes transparent to the waves. It follows immediately that there is the possibility that the solution is capable of de-balancing one side of the fluid-structure coupling, since in the extreme case where $r_r = 1$ everywhere, the cells would not get pressurized and would not generate ship waves. The cells where the roof relaxation is applied can never become fully-pressurized.

The mechanism of action of this solution is that it reduces the displaced volume within the nested-Newton loops. At each iteration where the pressure η increases, so does the free volume, reducing the volume that has to be pushed

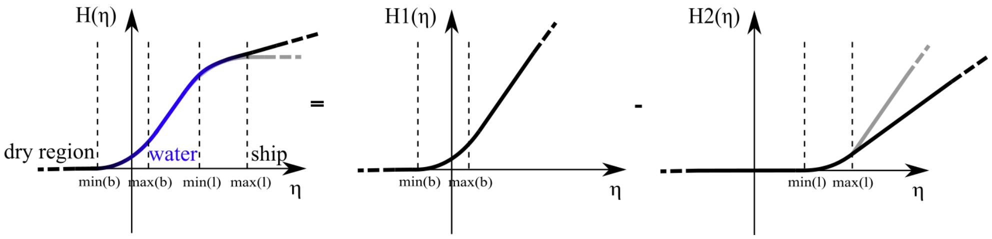


Figure 2.27. Modified volume functions H,H_1 and H_2 due to the roof-relaxation technique. The low-opacity lines are the original volume function of Figure 2.7.

out by the pressure gradient. This is especially true for the cells at the bow. Once the convergence of the system has been reached, this additional volume that has penetrated the ship hull is incorporated in the cell water volume $V_{i,j}^{n+1}$ and depth $H_{i,j}^{n+1}$. We reach the stability with a cost of an error in the ship-generated wave prediction. With the application of the relaxation, the model behaves as if the ship were smaller. The effective size is the one described by the variable $H_{i,j}^{n+1}$ in the ship region, which corresponds to the reconstructed ship only when $r_r = 0$ everywhere.

We decided to apply a relaxation that is variable in space. The coefficient r_r gradually changes from 1 to 0 starting from the waterline and going towards the center of the ship region. In this way, the relaxation acts as a transition buffer to the fully-pressurized state, and during the horizontal motion the request for displacement is spread among more cells, instead of only one. We limited the buffer extension to a fixed number of cells, which has the benefit to use the relaxation where it is needed the most and to reduce the ship-induced wave error. Also, as the grid is refined the relaxation area is smaller and the solution converges towards the one where no-relaxation is applied.

In our tests with different configurations of r_r, we observed that it is important to transition gradually especially in the inner part, where the intensity gets closer to zero. To clarify what we mean by that, let us consider two configurations: in the first one, the intensity is 1 at the waterline and decreases to 0.5 at the end of the buffer zone. We remind that the fully-pressurized region corresponds to the use of $r_r = 0$. In the second configuration, the intensity is 0.5 at the waterline and decreases to 0 at the end of the buffer zone. In the former case, there is a jump of r_r in the inner part, from 0.5 to 0, while in the latter case the same jump is at the waterline, from 1 to 0.5. If we compare the results of these two configurations, the latter will perform better in terms of pressure oscillations.

2.13.5 Smooth ship shape

In the discussion about $\partial\left[H(\eta)\right]/\partial t$, it was noticed that if the floating object has vertical sides, during a horizontal motion the displaced volume spans among the smallest number of cells. This concentrates the destabilizing term in a small area, increasing the pressure gradients. On the other hand, the internal horizontal sections of the object do not contribute to the instability since in their area of competence the volume of water has already been displaced. Considering these premises, another solution consists in smoothing the hull shape. This solution is inspired to the blurring process for image processing.
Before the application of this new algorithm, the input ship goes through the reconstruction process, which provides as output a set of piece-wise constant elements. These elements have different elevations, in general. The biggest intervention should be concentrated near the vertical walls, while long horizontal parts should be left untouched. The term "vertical wall" in this context refers to the case in which one element is below the free surface while its neighbor is above the free surface; since convex shapes are usually adopted, this condition is found at the waterline, where the transition to pressurized state happens. The shape smoothing should reduce the elevation difference between neighboring elements.
The algorithm is the following: a nxn matrix, called filter, is chosen. For each subcell of the original reconstructed shape, the surrounding subset of nxn subcells is taken. An element-wise product between the filter and the subset is done, and the n^2 values are summed. This value, which is a weighted average, is the new elevation of the reconstructed shape.
The dimension n of the filter is an odd number, so that the value that is going to be replaced is at the center. The coefficients of the matrix are chosen so that the subcells close to the center have additional weight; this condition is satisfied by many functions, e.g. the gaussian. In the implementation, the normalized $n°$ row of Pascal's triangle was chosen. The 5x5 filter is computed in equation (2.89) as an example.

$$\frac{1}{256}\begin{bmatrix}1\\4\\6\\4\\1\end{bmatrix} * \begin{bmatrix}1&4&6&4&1\\1&4&6&4&1\\1&4&6&4&1\\1&4&6&4&1\\1&4&6&4&1\end{bmatrix} = \frac{1}{256}\begin{bmatrix}1&4&6&4&1\\4&16&24&16&4\\6&24&36&24&6\\4&16&24&16&4\\1&4&6&4&1\end{bmatrix}. \qquad (2.89)$$

If we are in 1D, the kernel is composed only of the central row of the 2D kernel, renormalized. The reconstructed shape has a buffer to allow the application of the smoothing at the borders. The extension is composed of $\lfloor n/2 \rfloor$ subcells that have the value of the closest subcell. The smoothing can be repeated multiple times, using as input the previous-iteration shape, for example until the rounded sides extend more than one computational cell.

We remark that the smoothing is applied after the ship is rotated based on the roll, sway and yaw angles. With this order of operations, each time iteration the shape smoothing acts on different parts of the ship with different intensity, depending on how the elements align with the vertical.

There is a justification why the smoothing is applied to the reconstructed ship, and not to the input ship or to η_{lid}. The input ship has no relation with the subgrid: the number of elements, their distribution in space and their connectivity are different. Besides the complexity of adapting the subroutine to handle non-uniform input, the smoothing's computational time of a given input would be fixed, which could be unreasonable for coarse grids. Also, choosing the filter's coefficients would not be straightforward. η_{lid} in the ship region is a good candidate when we are not using the smooth-movement algorithm described in Section 2.13.3, since it has the same characteristics of the reconstructed ship. However, the smooth movement is so useful that it is always active, and in this case each subcell of η_{lid} is split in multiple parts with different values and areas. The shape smoothing would become more computationally expensive and the choice of the filter's coefficient not simple, with no particular benefit to the output quality.

We observed that with a smoother ship shape there could be significant reductions in the total computational time, mostly due to the more gradual transition between states and the spreading of displaced volumes among more cells, which generates smaller velocity gradients and a reduction in the maximum absolute flow velocity, making the CFL condition less restrictive.

2.13.5.1 Smooth shape vs roof relaxation

We want to emphasize the differences and similarities between the smooth-shape solution presented in this Section and the roof-relaxation solution presented in Section 2.13.4. They are similar in their action, which is to modify the depth function to get a rounded profile in a non-uniform way. This reduces the difference of water depth between one cell and another and may also reduce the total displaced volume during the horizontal motion. Also, both have been engineered to focus their effect on the parts with strong inclination. However, they operate along clearly discernible different paths. First, the water depth function is different: with the roof relaxation, the ship reconstruction is not altered, so the cell will start to be in the pressurized state at a lower elevation, more coherent with the input ship; after that point, it will always be partially pressurized. With the shape smoothing, there is more volume to be filled with water before it is pressurized; once η reaches the hull, is it fully pressurized. The smooth-shape solution is not able to create a partially-pressurized state by itself, there must be at least one other mechanism active, such as the smooth movement, the subgrid, or the roof relaxation. If the computation of forces is done on the reconstructed ship, in the case of the smooth shape the horizontal components are more accurate; this is because the smoothing decreseas the inclination of the sides, increasing the number of discretization points for those sides in the reconstruction.

2.13.6 Cell merging

In Section 2.13.1.1 we showed that pressurized cells are more prone to instabilities due to the missing or restricted capacity for the water to occupy the cell space. A reduction of $\partial H/\partial \eta$ increases the pressure gradients, and consequently the velocity gradients, required to push out the displaced volume. The main idea, inspired by the work of Seo and Mittal [103], is to create a procedure to transfer the additional displaced-volume source term from the pressutized cells to free-surface cells. This volume of water would have left the ship region anyway, as a result of the solution of the system. If we move part of the source term before the solution algorithm starts, we alter the convergence behavior so that the pressure gradients get smaller. In the destination cell, the water volume has to increase of the transferred amount, but it can do it benefiting from the free-surface cell condition, where $\partial H/\partial \eta = 1$. There, the pressure gradient will be limited and the overall water elevation profile will be smoother. It happens at the cost of the burden of choosing the destination cell for each volume we move; before, this was an automatic result of the system convergence. We chose to displace each volume to the nearest free-surface cell, because it is where we assume it would go. As a side note, if we chose to displace the volume on the opposite side of the ship, where the cell is transitioning in the opposite direction, we would completely lose the coupling, the ship would move without generating waves.

We specify that the partially-pressurized cells are excluded from the pool of destination candidates, even though they are often the closest to the waterline. Some of these cells are the origin of the displaced volume, hence placing the volume there would short-circuit the algorithm. In any case, their reduced $\partial H/\partial \eta$ makes them inefficient in the stabilization and would make the effort futile. The error committed from this exclusion vanishes as the grid is refined. Also, the floating object we simulate have big inertia and the direction changes

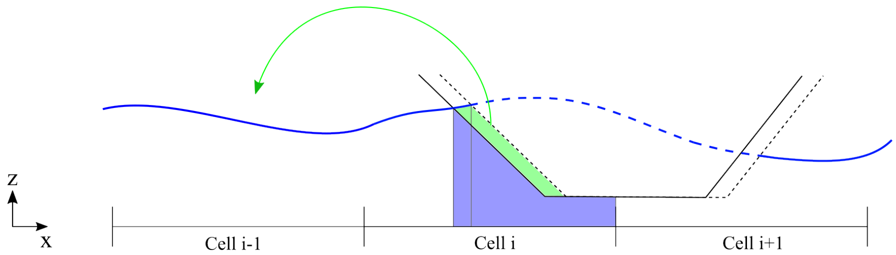


Figure 2.28. Illustration of the cell-merging feature. The green volume is displaced in the closest non-pressurized cell.

slowly; this means that even if we place the volume in the transitioning cell, it is likely that most of this volume will be moved out eventually in a successive time iteration, as the ship proceeds to occupy the cell. This consideration suggests that the error of exclusion is limited.

2.13.7 Rusanov dissipation

We determined that in the ship region, the system is more susceptible to generate strong pressure gradients because the coefficients of the matrix $\boldsymbol{A}$ are small. One tool at our disposal to reduce gradients is the dissipation, which usually is an unwanted numerical error because it levels out waves, but in this case we can use it to reduce the pressure gradients.

A Rusanov-type dissipation term $\boldsymbol{R}_d$ is introduced in the mass equations of the ship region, so that the transition between hyperbolic and elliptic systems is smoother. This term has the same structure of the pressure gradients, so the coefficients of the matrix are altered but the matrix remains symmetric, sparse and positive definite.

$$H\left(\boldsymbol{\eta}^{n+1}, \boldsymbol{G}^{n+1}, \boldsymbol{\varphi}^{n+1}\right) + \left(\boldsymbol{T} + \boldsymbol{R}_d\right)\boldsymbol{\eta}^{n+1} = \boldsymbol{rhs}^n, \tag{2.90}$$

$$\begin{split}(R_d)_{j,i} = 0.5\left(\sqrt{10}^{-\log_{10}(5\Delta t)}\alpha_d\,\Delta t\, s_{max}\right)^2 \Bigg[\frac{\left(\eta_{j,i}^{n+1} - \eta_{j,i-1}^{n+1}\right) - \left(\eta_{j,i+1}^{n+1} - \eta_{j,i}^{n+1}\right)}{\Delta x^2} \\ + \frac{\left(\eta_{j,i}^{n+1} - \eta_{j-1,i}^{n+1}\right) - \left(\eta_{j+1,i}^{n+1} - \eta_{j,i}^{n+1}\right)}{\Delta y^2}\Bigg].\end{split} \tag{2.91}$$

The dissipation requires the definition of a speed s_{max}. The actual speed of the wave in the pressurized region is unknown. In a region with elliptic equations the wave speed would approach infinity, but in this case the ship is moving vertically and horizontally and we do not know how this motion affects the speed. We use s_{max} equal to the wave speed in shallow water, as in equation (2.92):

$$s_{max} = ||\boldsymbol{u}_G|| + \sqrt{gH}. \tag{2.92}$$

The intensity of the dissipation can be calibrated using the coefficient α_d, and the factor $\sqrt{10}^{-\log_{10}(5\Delta t)}$ scales the intensity in a way than keeps the overall effect of α_d similar when the timestep changes.
The dissipation can be applied in any portion of the domain, so nothing forbids to apply it only close to the waterline as we did for the roof relaxation of Section 2.13.4. However, in the implementation we preferred to add the dissipation term in the entire ship domain because it is more efficient in controlling the oscillations. A dissipation only at the transition border would not guarantee the absence of voids below the hull in the middle.
The dissipation should be applied with care because it comes at a cost, which is an absorption of the wave energy and an alteration of the hull pressure. Thus, the ship dynamics is different and the ship-generated waves are smaller.

2.13.8 Combination

The proposed solutions to the horizontal-motion problem can be applied together. Their concurrent application creates some interactions between the solutions, not only because the simulation is evolving in time, but because they modify the domain of application of another solution and they modify the system components. Usually, to obtain a stable η profile not all the features are required, but it is interesting to see how they can work in synergy.
Let us explore a typical time iteration in a simulation in which all the features explained in the Sections from 2.13.3 to 2.13.7 are active, and highlight their interactions. The first action is done by the smooth-movement algorithm: the input ship is brought on a auxiliary grid where it is reconstructed. This numerical hull is then rounded by the smooth-shape algorithm and brought back in the original domain grid. The water depth H is computed using a space-variable depth function, which have been modified by the roof-relaxation method and also by the inexact overlapping of the ship cells with the domain subcells. The forces and torques acting on the ship are computed on the rounded numerical hull and they control the future position and orientation of the ship. With the knowledge of the current and next ship configurations we compute the volume of water that needs to be displaced by the cell-merging method. We compute the coefficients of the matrix, and in the ship region there

are additional terms coming from the Rusanov dissipation. The nested-Newton algorithm starts and the $\partial H/\partial\eta$ are those of the modified depth function.

2.14 Horizontal viscosity

In this Section, we introduce the horizontal viscosity operator in the system. The vertical viscosity is already present since Section 2.5. Contrarily to the vertical viscosity, the horizontal viscosity terms are taken explicitly. This implies that we make the CFL condition more restrictive. However, since the derivatives are taken with respect to the horizontal directions, they should not pose a strong limitation on the timestep size.

$$\Delta t = \min\left(\Delta t_{max}\ ,\ \frac{\mathrm{CFL}}{2\left(\frac{|u|_{max}}{\Delta x}+\frac{|v|_{max}}{\Delta y}+\frac{\nu_{max}}{\Delta x^2}+\frac{\nu_{max}}{\Delta y^2}\right)+c}\right). \tag{2.93}$$

with CFL=0.9 and $c = 0.1$.
The discretization is done as in equations from (2.94) to (2.97):

$$\begin{aligned}\frac{\partial}{\partial x}\left(\nu\Delta z\frac{\partial u}{\partial x}\right) \approx \frac{1}{\Delta x}\Bigg[&\nu^n_{i+1,j,k}\Delta z^n_{i+1,j,k}\left(\frac{u^n_{i+\frac{3}{2},j,k}-u^n_{i+\frac{1}{2},j,k}}{\Delta x}\right)\\ &-\nu^n_{i,j,k}\Delta z^n_{i,j,k}\left(\frac{u^n_{i+\frac{1}{2},j,k}-u^n_{i-\frac{1}{2},j,k}}{\Delta x}\right)\Bigg],\end{aligned} \tag{2.94}$$

$$\begin{aligned}\frac{\partial}{\partial y}\left(\nu\Delta z\frac{\partial v}{\partial y}\right) \approx \frac{1}{\Delta y}\Bigg[&\nu^n_{i,j+1,k}\Delta z^n_{i,j+1,k}\left(\frac{v^n_{i,j+\frac{3}{2},k}-v^n_{i,j+\frac{1}{2},k}}{\Delta y}\right)\\ &-\nu^n_{i,j,k}\Delta z^n_{i,j,k}\left(\frac{v^n_{i,j+\frac{1}{2},k}-v^n_{i,j-\frac{1}{2},k}}{\Delta y}\right)\Bigg],\end{aligned} \tag{2.95}$$

$$\begin{aligned}\frac{\partial}{\partial y}\left(\nu\Delta z\frac{\partial u}{\partial y}\right) \approx \frac{1}{\Delta y}\Bigg[&\nu^n_{i+\frac{1}{2},j+\frac{1}{2},k}\Delta z^n_{i+\frac{1}{2},j+\frac{1}{2},k}\left(\frac{u^n_{i+\frac{1}{2},j+1,k}-u^n_{i+\frac{1}{2},j,k}}{\Delta y}\right)\\ &-\nu^n_{i+\frac{1}{2},j-\frac{1}{2},k}\Delta z^n_{i+\frac{1}{2},j-\frac{1}{2},k}\left(\frac{u^n_{i+\frac{1}{2},j,k}-u^n_{i+\frac{1}{2},j-1,k}}{\Delta y}\right)\Bigg],\end{aligned} \tag{2.96}$$

$$\frac{\partial}{\partial x}\left(\nu \Delta z \frac{\partial v}{\partial x}\right) \approx \frac{1}{\Delta x}\Bigg[\nu^n_{i+\frac{1}{2},j+\frac{1}{2},k}\Delta z^n_{i+\frac{1}{2},j+\frac{1}{2},k}\left(\frac{v^n_{i+1,j+\frac{1}{2},k}-v^n_{i,j+\frac{1}{2},k}}{\Delta x}\right) \\ -\nu^n_{i-\frac{1}{2},j+\frac{1}{2},k}\Delta z^n_{i-\frac{1}{2},j+\frac{1}{2},k}\left(\frac{v^n_{i,j+\frac{1}{2},k}-v^n_{i-1,j+\frac{1}{2},k}}{\Delta x}\right)\Bigg]. \tag{2.97}$$

These explicit terms are added to $\boldsymbol{E}^n$ of equation (2.30).
At the boundaries, the velocities on the mixed derivatives $\partial u/\partial y$ and $\partial v/\partial x$ needs a definition because they are out of the domain. The choice can vary for different boundary stretches, each velocity depends on the problem: if we have a moving wall then we impose the wall's speed; if the wall is not moving, it is always zero; if there is an open boundary, we take it equal to the closest inner velocity, e.g. $u^n_{i+\frac{1}{2},J_{max}+1,k} = u^n_{i+\frac{1}{2},J_{max},k}$.
At the walls, we may want to impose the slip condition, which influences how much exchange of kinetic energy there is due to the friction between the wall and the water. If we want to apply the slip condition, i.e. there is no friction at the wall, the boundary velocity must be defined as the closest inner velocity, regardless of the wall speed.
Inside the domain there may be channels or canals where we want to apply a wall condition. This is possible by imposing an elevation of the bathymetry in the cell above any expected free-surface level. In this way we create a dry cell, where the velocity is zero by definition, which is like a vertical wall interacting with the water in neighboring cells through the horizontal viscosity operator and according to our choice of slip or no-slip condition.

2.15 Non-reflecting boundary conditions and sponge layer

Especially when working with two or three-dimensional domains, it is necessary to restrict the area of the simulation to the minimum in order to keep the total computational time within an acceptable range. On the other hand, when a wave propagates in the domain and reaches the boundaries of the numerical domain, the signal reflects back if the boundary condition is not chosen appropriately. This reflected wave keeps traveling within the domain and affects the solution wherever it passes; this can heavily disturb the measurements at the gauges and can induce heaving if the wave reaches a ship. To limit this effect, a couple of solutions were implemented.

The first solution is to apply a Dirichlet boundary condition that is informed of the incoming wave characteristics and lets it exit without causing reflection. Choosing a perfectly non-reflecting boundary condition is a complex task, which is why many different solutions have been developed that are simpler but partially reflect the wave. A partial reflection can be acceptable if the error it produces is of the order of the discretization error. Additional information on the non-reflecting boundary conditions (NRBC) can be found in the works of Givoli [61] and Durran [50]. What has been implemented is a Higdon NRBC of order one [62], which corresponds to the Sommerfeld radiation condition. An example of computation of a NRBC at the west border is provided in equation (2.98):

$$H^{n+1}_{\frac{1}{2},j} = H^{n}_{\frac{1}{2},j} - \frac{\Delta t}{\Delta x} C_1 \left(H^{n}_{\frac{1}{2},j} - H^{n}_{1+\frac{1}{2},j} \right), \tag{2.98}$$

where C_1 is the wave speed estimate, taken as $\sqrt{gH_{max}}$. This condition is based on a transport equation.

The second solution is to dissipate the incoming wave before it reaches the boundary. This is done by creating a sponge layer, a portion of the domain close to the boundaries where a relaxation scheme is applied [14, 28]. A relaxation term is introduced in the momentum equation, as shown in equation (2.99):

$$q^{n+1}_{i+\frac{1}{2},j} = q^{*}_{i+\frac{1}{2},j} - \frac{\Delta t}{\Delta x} g H^{n}_{i+\frac{1}{2},j} \left(\eta^{n+1}_{i+1,j} - \eta^{n+1}_{i,j} \right) + \Delta t \beta_{i+\frac{1}{2},j} q^{n+1}_{i+\frac{1}{2},j}, \tag{2.99}$$

where β is the sponge-layer function that defines the space-variable intensity of the relaxation. Indeed, this intensity must increase from the inner part to the outer part of the sponge layer, otherwise the incoming wave would reflect when entering the sponge layer. The sponge-layer function is defined as $\beta = \omega_w B_{sl}$, where ω_w is the wave frequency and B_{sl} is the shape function. Different forms of the shape function exist in the literature; a trigonometric shape function has been implemented, as shown in equation (2.100):

$$B_{sl} = B^{max}_{sl} \sin^{n_B} \left(\frac{\pi}{2} \frac{x - x_{sl}}{x_{BC} - x_{sl}} \right) \mathcal{H} \left(x - x_{sl} \right), \tag{2.100}$$

where B^{max}_{sl} is a chosen maximum value of the shape function, $n_B > 0$, x_{sl} is the coordinate where the sponge layer starts, x_{BC} is the coordinate of the boundary and $\mathcal{H}$ is the Heaviside function that makes the sponge layer ineffective outside of its region of application.

2.16 STL

In Section 2.3, the procedure to compute a reconstructed ship is explained; the result is a piece-wise constant surface that defines the elevation of the hull. In Section 2.6, it is explained how the reaction forces are computed as the Riemann-sum over the reconstructed hull of the pressure-area product, and how the information of the inclination is used to account for the three-dimensionality of the surface.
As has been pointed out in Section 2.3, the reconstruction procedure is based on the intersection of the blueprint ship with the verticals that pass through the subcells centers. This makes the algorithm exact when it approximates horizontal surfaces, while it misses vertical surfaces; the inclined surfaces are in between in terms of accuracy, the points that approximate a given area decrease in proportion to the inclination with respect to the horizontal plane. As a consequence, the horizontal forces are computed in very few points, not enough to provide accurate values.
A solution to this problem is the use of a surface that has elements on the vertical sides. Thanks to the collaboration with the BAW, a triangulated surface of a model ship is interfaced with the model. This surface is read from a file in the STL format, in which a series of triangles are defined by the coordinates of the vertices and by the components of the normal vector. This input ship, called Referenzschiff, can be visualized in Figure 2.29, while its real-world equivalent is the model ship shown in Figure 2.30. The STL input ship is used directly in substitution to the reconstructed ship for the forces computation, where each triangle is represented by its barycenter. Thanks to the STL input ship, the precision of the horizontal forces and the torques has drastically improved, while for the vertical force only a marginal difference was observed.

The reconstructed ship is also a key element in the fluid-structure interaction, because each tile of the surface sets the upper bound of the depth function H in its cell. This implies that the reconstructed ship determines the volumes of water displaced during its motion. Considering that the reconstruction process is more accurate for the ship horizontal components by construction, it was decided to not upgrade a volume computation done directly on the STL input ship. As shown by the verification of the heaving motion and by the limited improvement to the vertical force, the volume described by the reconstructed ship seems precise enough for the intended application. Also, doing computations on the reconstructed ship is quicker than doing them on the STL input ship, simply because there are less elements.

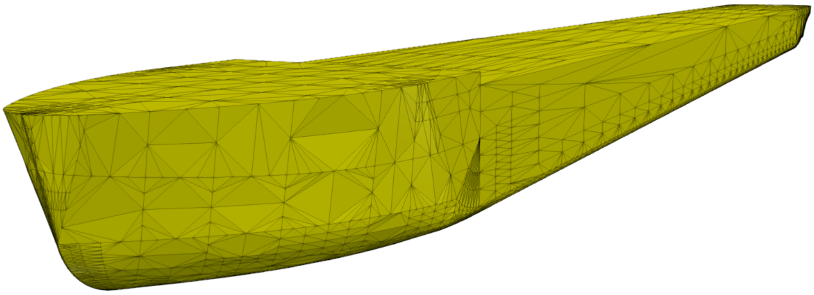
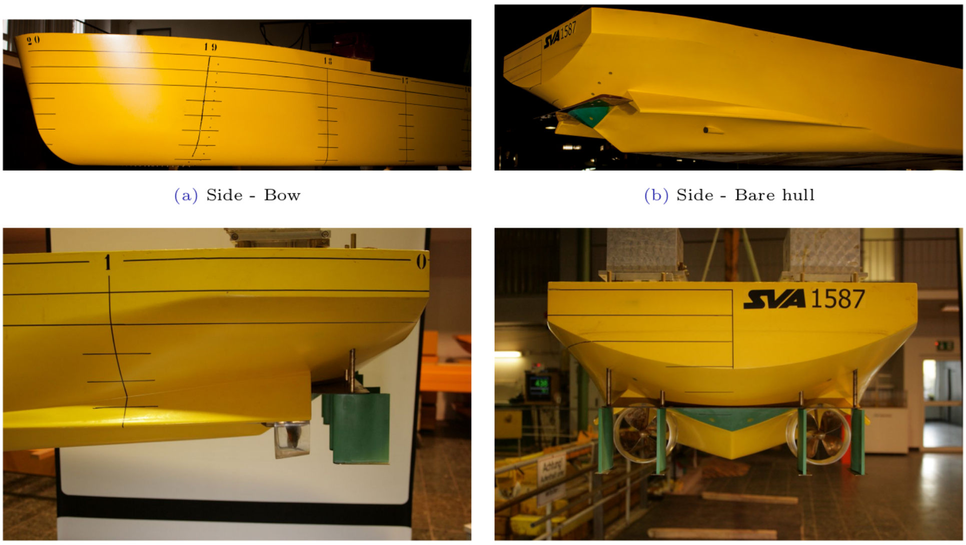


Figure 2.29. View of the Referenzschiff, composed of 18918 triangles.

The decision of maintaining the reconstructed ship requires a procedure to create it starting from the STL input ship. In particular, the algorithm expects a set of connected layers and edges as explained in Section 2.3. The procedure consists of the following steps:

- For each of the triangles of the STL input ship, the barycenter is determined.
- The barycenters are divided based on their elevation in a number of groups equal to the number of layers of the reconstructed ship.
- For each group of point, only the most external points are kept. The selection is done by a convex hull algorithm, specifically the Graham scan algorithm [66].
- A point in the center of the ship is selected. This point is computed as the barycenter of all the input points.
- The central point is the origin for the lines that will define the edges of the reconstructed ship. The lines are drawn by dividing the round angle in a number of parts defined by the number of edges. The choice of each angle is influenced by the general dimensions of the ship, so that even for an elongated object like a ship the edges are uniformly spread along the ship hull.
- The intersections of these lines with each convex hull are the points in the edge/layer format.



(a) Side - Bow

(b) Side - Bare hull

(c) Side - Appended hull

(d) Back - Appended hull

Figure 2.30. This model ship represents a typical inland ship and is the ship that is approximated in Figure 2.29. It is used in laboratory experiments to collect data for validation purposes. (Source: Lahbib Zentari, SHINING 2022).

2.17 Parallelization

The application of a Cartesian grid where different dynamics happen at different scales has a disadvantage, which is that the simulation of small objects in a large domain requires many computational cells. This translates in a large system that needs to be solved each time iteration, which in turn increases the computational time. In order to improve the performance, the processing power has been increased by using multiple core processors working in coordination. To do that, the domain has first to be decomposed in different parts and each part is assigned to a process. Every process has complete knowledge of the variables inside its subdomain; only a few variables are known globally. Since each subdomain needs boundary conditions at their borders, neighbor subdomains exchange the values of the variables at the borders.

In the implementation, the domain is splitted horizontally in rectangles composed of the same number of cells, in order to balance the computational load; the algorithm is also informed of the ratio between the number of cells in the two directions, so that the perimeter of each subdomain is minimized and with it the number of variable in the communication halo. The communication is

handled by the Message Passing Interface (MPI) communication protocol.
The parallelization is most effective when the number of cells is high. Indeed, if each processor is still adequately challenged by its share of computations even when the computational load is divided among many processors, the additional communication cost remains a small percentage of the total computational time.
Some words have to be spent for the floating object. To simplify the implementation, the entire ship region in known by all the processes. Since its size relative to the domain is usually small, the amount of data to be exchanged is not excessive. Each processor does the same ship reconstruction, forces computation and evolution of the position. This is inefficient in principle, but it is not a huge concern because these operations are generally quick to be performed. Also, since at different points of the algorithm the processes have to synchronize, doing the ship-related computations only in some of the subdomains would unbalance the computational load, leaving the other processes to wait.

To provide a quantitative measure of the efficiency of the model and its implementation, the total computational time is reported in Table 2.1 for a test case run in different configurations. This test case, which is labeled C-V1 in Section 3.4, involves a ship towed in a canal with a maximum speed of 2.384 m/s. The table is divided in three main parts: a base case "b" with no subgrid and running in serial mode, a case "s" with a subgrid and a case "p" running in parallel. For each part, four simulations with increasing number of computational cells are run.
The domain is 1600 m in the x direction and 60 m in the y direction. The end time of the simulations is 650 s. The timestep is 0.1 s, which is always below the CFL condition. Generation of outputs has been suppressed for these simulations. The CPU running the tests is a 12th Gen Intel(R) Core(TM) i7-12700H 2.70 GHz.
In Table 2.1, N_p is the number of processes running in parallel. NNo_{avg} and NNi_{avg} are the average number of iterations of the outer and inner loops during the application of the nested Newton algorithm; they are connected to the transition to pressurized and dry cells, respectively. CG_{avg} is the average number of iterations of the conjugate gradient algorithm.
A series of considerations emerges from the observation of these results:

- The parameter that governs the computational time is the number of cells, not only because it corresponds to the number of unknowns, but also because NNo_{avg} and CG_{avg} grow with it, increasing the computational

	$I_{max}J_{max}$	Subgrid	N_p	Total time [s]	NNo_{avg}	NNi_{avg}	CG_{avg}
b1	400x30	1x1	1	250	3.081	1.114	32.476
b2	800x30	1x1	1	521	3.973	1.150	46.284
b3	1600x30	1x1	1	1310	4.363	1.157	76.491
b4	1600x60	1x1	1	6400	4.540	1.207	89.995
s1	400x30	4x4	1	581	4.335	1.111	33.841
s2	800x30	4x4	1	1617	4.725	1.132	46.964
s3	1600x30	4x4	1	3182	5.024	1.181	73.690
s4	1600x60	4x4	1	7769	5.312	1.171	93.666
p1	400x30	1x1	6	234	3.081	1.113	32.478
p2	800x30	1x1	6	401	3.973	1.149	46.294
p3	1600x30	1x1	6	845	4.363	1.157	76.494
p4	1600x60	1x1	6	1469	4.540	1.206	89.998

Table 2.1. Measurements of the computational time, the number of iterations for the outer and inner loops in the nested Newton algorithm and the number of iterations in the conjugate gradient algorithm. The number of cells, subcells or processes vary in each line of the table.

cost of each time iteration.

- The subgrid, which increases the number of subcells by a factor of 16, causes a smaller increase in the time, usually by a factor between 2 and 3.
- Running the script in parallel is more efficient with a high number of cells. For coarser grids, the additional costs of communication between the processors become significant.
- NNo_{avg}, NNi_{avg} and CG_{avg} are the same in the serial and parallel executions "b" and "p", as they should be.

2.18 Frictional resistance

In this Section we introduce the frictional resistance, or skin friction, that is generated at the interface between the hull and the water due to the relative motion of one over the other.

The lack of frictional resistance in the model could cause the underestimation of the waves, squat and return current of a ship moving in a confined space.

Indeed, the skin friction induces a stress on the flow directed as the ship speed, and the current below the ship modifies accordingly; at steady state, this means that the return current at the ship's sides must get stronger. A stronger current could increase the drawdown and with that the squat.
Different methods have been developed to model the skin friction [12, 71]. We choose to adopt the method proposed by the *International Towing Tank Conference*, or ITTC [94, 96]. The frictional resistance is thought as a function of the water density ρ_w, the wetted-ship surface S, the ship velocity relative to the water u_r and a resistance coefficient C_T.

$$R_T = \frac{1}{2}\,\rho_w\,u_r^2 \int_S C_T\,dA \tag{2.101}$$

The integral is discretized on the STL when available, otherwise on the reconstructed ship.
The total resistance coefficient C_T is equal to

$$C_T = C_F + \Delta C_F + C_A + C_R \tag{2.102}$$

where the residual resistance coefficient C_R is takes as zero, in absence of data from a resistance test.
The ITTC57 Model-Ship Correlation Line C_F is:

$$C_F = \frac{0.075}{(\log_{10} Re_x - 2)^2} \tag{2.103}$$

where Re_x is the Reynolds number based on the distance x_{bl} from the point where the boundary layer starts to form. Such point is taken as the first ship point encountered on the relative-velocity axis, front side.

$$Re_x = \frac{u_r x_{bl}}{\nu} \tag{2.104}$$

ΔC_F is called the roughness allowance and it considers the additional resistance due to the hull roughness:

$$\Delta C_F = 0.044 \left[\left(\frac{k_S}{L_s} \right)^{1/3} - 10 \cdot Re_x^{-1/3} \right] + 0.000125 \tag{2.105}$$

where L_s is the ship length. In absence of case-specific data, we use the ITTC-recommended value of $k_S = 150 \cdot 10^{-6}$ for the roughness of the hull surface. The effects of surface roughness in shallow waters was recently investigated by Song et al. [105]; they concluded that sinkage is independent of the roughness condition.

The correlation allowance C_A is the coefficient that corrects for the difference between the model and the full-scale results:

$$C_A = (5.68 - 0.6 \log Re_x) \cdot 10^{-3} \tag{2.106}$$

The skin friction works against the ship motion so it has to be included in the force $\boldsymbol{F}$ used in the ODE that determines the next position of the center of mass.

The same skin friction is also transferring energy between the ship and the water, so the water flow is also affected. We then add the local wall shear stress τ_w^* to the momentum equation, and we discretize it as an explicit term.

$$\tau_w = \frac{1}{2}\, \rho_w\, u_r^2\, C_T \qquad \tau_w^* = \frac{\tau_w}{\rho_w} \tag{2.107}$$

Since we are using the equations with the momentum as the unknown, the skin-friction term has to be multiplied by a length. This length cannot be the whole water column H because we expect that after a certain value of H the skin friction should be constant and it should affect only the superficial part of the water body. For these reasons, we choose the depth Δz^*, which has the property that $\int_b^l \Delta z^* = \delta_{99}$, where b represents the bathymetry, l the ship hull and $\delta_{99} = 0.37 \cdot x \cdot Re_x^{-1/5}$ is 99% the turbulent boundary layer thickness [102]. In multi-layer simulations, the length Δz^* is split among the vertical k-layers in the range $[l - \delta_{99}, l]$, which are the layers close to the ship hull; outside this range, $\Delta z_k^* = 0$. The skin-friction term updates the vector of the known terms, as shown in equation (2.108).

$$\boldsymbol{E}^n = \begin{pmatrix} \Delta z_M q_M^* + \Delta t \gamma_T \Delta z_M u_a + \Delta t \Delta z_M^* \gamma_{sf} \tau_w^* \\ \Delta z_{M-1} q_{M-1}^* + \Delta t \Delta z_{M-1}^* \gamma_{sf} \tau_w^* \\ \vdots \\ \Delta z_m q_m^* + \Delta t \Delta z_m^* \gamma_{sf} \tau_w^* \end{pmatrix}. \tag{2.108}$$

The skin-friction term is modulated by a calibration factor γ_{sf}, which allows to modify the friction to better match some reference data.

3 Numerical results

All the following results are obtained applying the implementation of the 3D model in form of a Fortran code and the results are visualised using ParaView [3, 4].

3.1 Water flow

In this Section, we verify the model features and capabilities applying it to a selection of academic standard test cases. At the beginning, we verify the model applied for flows without floating objects and compare the results to available exact solutions.

Each test case is useful to highlight and verify specific features of the model:

- Dam-break problems over wet and dry bed (3.1.1): shock capturing, wetting and drying, importance of the momentum conservative discretization.
- Uniform flow (3.1.2): Dirichlet and Neumann boundary conditions, bottom friction.
- Oscillations in a lake (3.1.3): wetting and drying, non-trivial bottom with subgrid, numerical dissipation.

Throughout this section we make use of the SI system of units.

3.1.1 Dam-break problems over wet and dry bed

The first validation is a classical test problem in CFD, namely the so-called Riemann problem or dambreak problem for shallow water flows. We consider two cases here, one dambreak over wet bed and another one over dry bed. In both cases the bottom is flat ($b = 0$ m). In order to properly validate the 2Dxz model, we carry out each simulation with a single layer $\Delta z > H$ and with a number of z-layers of maximum size of $\Delta z_k = 0.1$ m, apart from the uppermost layer, whose size is dynamically adjusted in order to fit the free surface. Since the vertical viscosity coefficient is set to $\nu = 0$ m^2/s, the vertical velocity profile is constant and therefore the 1D model and the 2Dxz model must produce the same results.

Dambreak over wet bed The initial condition of the first Riemann problem consists in still water over flat bottom, with left water depth $H_L = 2$ m and right water depth $H_R = 1$ m. A constant time step of $\Delta t = 1 \cdot 10^{-3}$ s is used on a uniform Cartesian mesh composed of 100 cells in x direction. No-flux boundary conditions are applied on the left and right boundaries. In the left panel of Figure 3.1 the obtained numerical solution is compared against the exact solution provided by Stoker [110], and also available in the textbook of Toro [115]. The numerical results obtained with the mass and momentum conservative semi-implicit scheme agree well with the reference solution, capturing both the wave heights and the wave propagation speeds in a correct manner. We also observe that the solution obtained with the 1D model and the 2Dxz model coincide, as expected.

Dam-break over dry bed In the initial condition of the second Riemann problem the downstream side is dry. The comparison is made with Ritter's solution [100], which is formulated for a frictionless wide horizontal channel. Its biggest merit is simplicity, so it is a quick test to see if the wet/dry transition is treated appropriately. However, it is not in good agreement with experimental data because it does not consider friction, turbulence, vertical acceleration and non-hydrostatic pressure, see [30]. The exact solution of the Riemann problem can be found in [100], or in the textbook [115]. In the numerical simulation the boundary conditions are no-flux, the Strickler coefficient is set to $k_{St} = 100$ m$^{1/3}$/s so that bottom friction can be essentially neglected, the initial location of the dam is in $x_0 = 0$ m, the initial water heights on the left and on the right of the dam are $H_L = 1$ m and $H_R = 0$ m, respectively, the initial velocity is zero in the entire domain, a constant time step of $\Delta t = 1 \cdot 10^{-3}$ s is used and the number of cells is set to $I_{max} = 100$. The results are shown in the right panel of Figure 3.1. There is a good agreement between the numerical results and the analytical solution. We highlight that the numerical solution is a rarefaction wave thanks to the conservative discretization of the mass and momentum conservation equations, see also [77, 109] for the importance of momentum conservation in rapidly varied flows. For an alternative nonconservative discretization of the shallow water equations, unphysical solutions are obtained. Again, the 1D model and the 2Dxz model agree perfectly well with each other, as expected.

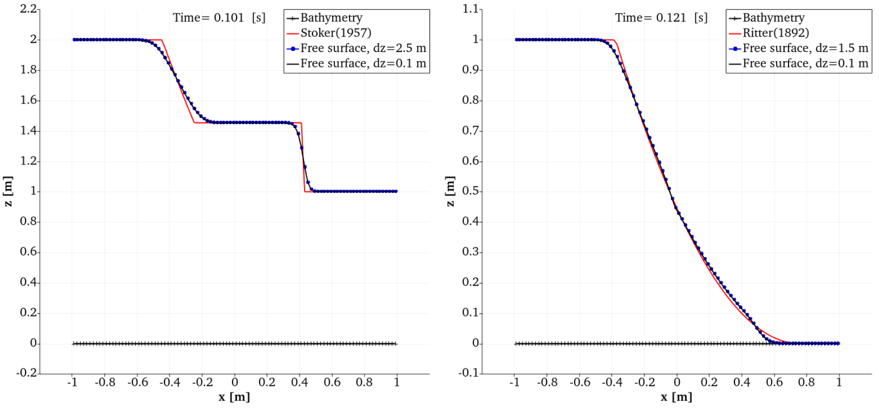


Figure 3.1. Two dam break problems. Dambreak over wet bed (left): Comparison of the new momentum conservative semi-implicit scheme with the analytical solution of Stoker(1957); comparison between the 2Dxz model with maximum layer size $\Delta z = 0.1$ m and the 1D model with only one layer of maximum size $\Delta z = 2.5$ m. Dambreak over dry bed (right): Comparison of the new momentum conservative semi-implicit scheme with the analytical solution of Ritter(1892); comparison between the 2Dxz model of maximum layer size $\Delta z = 0.1$ m and the 1D model with only one layer of maximum size $\Delta z = 1.5$ m.

3.1.2 Uniform flow

With the introduction of bottom friction in the model, it is possible to compute a numerical test case of the uniform flow in an open channel, proving the ability of the model to maintain it. In a uniform flow, both time derivative and the advection term vanish, so there exists a characteristic equilibrium between the pressure gradient and the friction term. The conditions for this flow to realize are that the inflow discharge remains constant in time and that the shape of the channel does not change in the direction of the flow. These are essentially never met in a real-world scenario; nonetheless, this flow condition is of great interest as a reference point and also because it can give insights about a river free-surface elevation as a function of flow velocity and average bottom slope. The equation that governs the flow is the Gauckler-Manning-Strickler formula [59, 87]:

$$u = k_{St}\, R_h^{\frac{2}{3}}\, s^{\frac{1}{2}} \tag{3.1}$$

where s is the constant bottom slope $s = -b_x$. Under the assumption of wide and rectangular channel, an explicit formula for the uniform water depth H_u

can be derived:

$$H_u = \left(\frac{q}{k_{St}\sqrt{s}} \right)^{\frac{3}{5}}. \tag{3.2}$$

Another parameter of interest is the critical water depth H_c, the depth of the transcritical flow at a given discharge:

$$H_c = \sqrt[3]{\frac{q^2}{g}} \tag{3.3}$$

With the background theory set in place, a simulation of a subcritical flow is presented. The initial conditions are a constant flux and a water height lower than the uniform one. On the west side, the Neumann boundary condition is constant in time and it is the initial water discharge q_0, on the east side the Dirichlet boundary condition imposes the water depth H_u. The parameters of the chosen computational test problem are $q_0 = 4.42$ m^2/s for the inflow discharge, $H_0 = 1.27$ m the initial water height, $H_c = 1.26$ m the critical depth, $H_u = 1.4$ m the uniform flow depth, $k_{St} = 40$ m$^{1/3}$/s the Strickler coefficient, $s = 4 \cdot 10^{-3}$ the constant bottom slope and $I_{max} = 100$ the number of cells. The results are shown in Figure 3.2, at time $t = 30$ when the ascending wave is in the middle of the domain. At this instant the transition to uniform flow is in progress, which makes visible that it is a smooth transition. On the east side of the wave, the water depth and velocity are approaching the expected values and the flow is successfully maintained without generating instabilities.

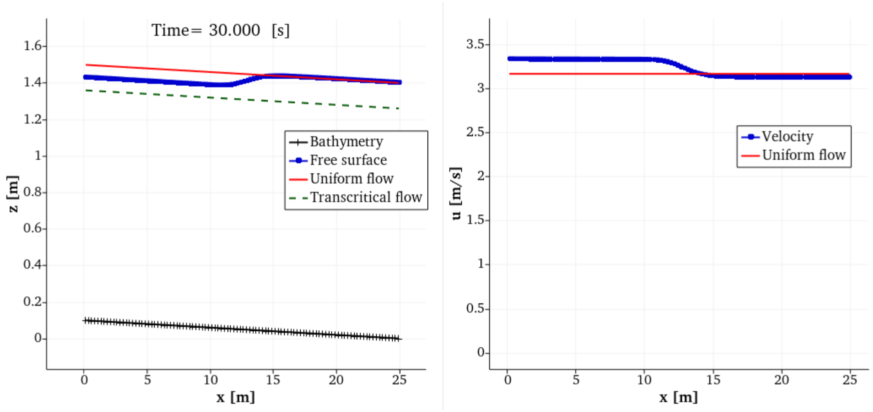


Figure 3.2. Numerical simulation of the uniform flow, starting from non-uniform flow conditions. At this timestep, the transition from initial to uniform conditions is still in progress, with a wave ascending from east to west

3.1.3 Oscillations in a lake

The model is also tested in a domain with a non trivial bathymetry. Specifically, the bottom is shaped as a parabola

$$b(x) = H_0 \left[\frac{1}{a^2} \left(x - \frac{L}{2} \right)^2 - 1 \right], \tag{3.4}$$

where H_0 is the water depth at vertex coordinate, $L = x_E - x_W$ is the domain width, and a is a parameter for the shape of the parabola. At each time the water depth and velocity are:

$$H(t,x) = \begin{cases} -H_0 \left\{ \left[\frac{1}{a} \left(x - \frac{L}{2} \right) + \frac{\tilde{B}}{\sqrt{2gH_0}} \cos \left(\frac{\sqrt{2gH_0}}{a} t \right) \right]^2 - 1 \right\} & \text{for } x_1(t) \le x \le x_2(t) \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

$$u(t,x) = \begin{cases} B \sin \frac{\sqrt{2gH_0}}{a} t & \text{for } x_1(t) \le x \le x_2(t) \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

with $\tilde{B} = \sqrt{2gH_0}/(2a)$. This particular configuration is intended to reproduce a case for which Thacker found an analytical solution for a frictionless bottom [113] in 2D, while the solution for the simplified 1D case was first provided by Delestre et al. in [46]. The positions of the wet/dry interfaces are denoted by $x_1(t)$ and $x_2(t)$ and read:

$$x_1(t) = -\frac{1}{2} \cos \left(\frac{\sqrt{2gH_0}}{a} t \right) - a + \frac{L}{2}, \qquad x_2(t) = -\frac{1}{2} \cos \left(\frac{\sqrt{2gH_0}}{a} t \right) + a + \frac{L}{2}. \tag{3.7}$$

For our simulations, we use the following setup. The water depth at the parabola vertex is set to $H_0 = 0.5$ m, the initial velocity is set to zero throughout the computational domain, the Strickler coefficient is chosen as $k_{St} = 20000$ m$^{1/3}$/s so that friction can be neglected, the maximum time step is $\Delta t_{max} = 1 \cdot 10^{-3}$ s, the computational domain is $\Omega = [0, 4]$ m, discretized with $I_{max} = 100$ uniform cells and the parabola shape parameter is set to $a = 1$ m. Two times after approximately five oscillations is shown in Figure 3.3. The numerical model and the analytical solution are in good agreement, but the former does not reach the same elevation at x_1 and x_2, due to numerical dissipation.

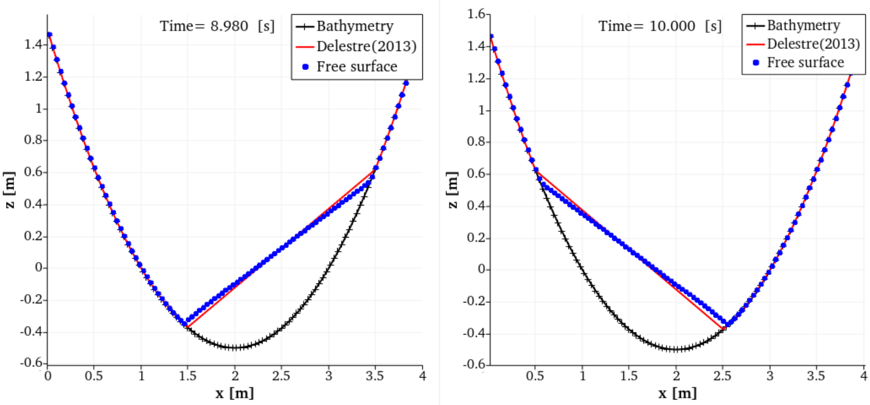


Figure 3.3. Flow of water in a parabolic lake. The cells get wet and dry cyclically. Comparison with Delestre et al. solution [46].

3.1.4 Nonhydrostatic pressure

In this Section, the Boussinesq-type dispersive term introduced in Section 2.12 is applied.

3.1.4.1 Dambreak

The dambreak-over-wet-bed test case of Section 3.1.1 is repeated. Compared to said previous simulation, the initial heights are lowered to $H_L = 0.02$ m and $H_R = 0.01$ m and the grid has been refined, so to make the effect of dispersion more evident. The reference solution is again the analytical solution of the hydrostatic case.
Figure 3.4 shows the results of the application of the nonhydrostatic-pressure term. Dispersion effects are visible at the front of the shock wave as oscillations of the free surface.

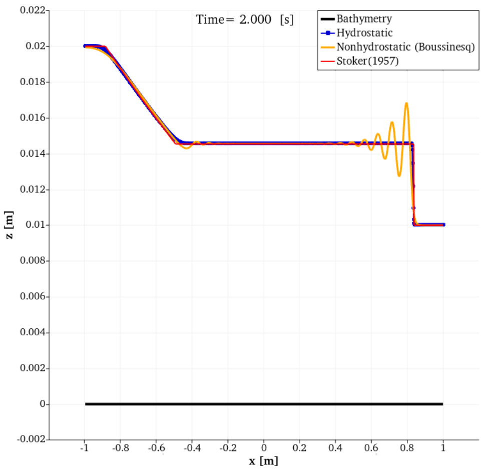


Figure 3.4. Application of the Boussinesq-type dispersive term in the dambreak case. Setup: $\Delta x = 0.002$ m, $\Delta t_{max} = 0.001$ s.

3.2 Vertical motion

In the following, we study the response of a ship in various simple fluid-structure interaction situations concentrating on the vertical degree of freedom (heaving). For a basic consistency-check of the method with elementary physics, the main requirement is the straightforward test that when the ship is in static equilibrium in a still water body, its position in time should not change, which checks if the method is well-balanced [11, 29]. Having checked this seemingly trivial property, we proceeded to verify the heaving motion with a floating body set in an unbalanced position. The use of a simplified hull geometry allows to obtain an analytical solution of the damped-harmonic oscillator equation for the body motion, which is compared to the results obtained with the numerical method presented in this thesis.

Test cases in this section:

- Return to equilibrium in deep water (3.2.1): heaving response and waves generation.
- Return to equilibrium in shallow water. (3.2.2): effects of the water depth on the heaving response and the waves generation.
- Heaving due to incoming waves (3.2.3): response to external regular wave field.

Throughout this section we make use of the SI system of units.

3.2.1 Return to equilibrium in deep water

The aim of this test case is to verify the response of a floating body, which can for example represent a passive ship with simplified geometry, which is moved out from its equilibrium floating position at the free surface of a deep water body. The oscillation pattern of the ship should be consistent with the corresponding one in the reference solution, which is the well-known case of a damped harmonic oscillator. At the initial time, the vessel with mass m_s is out of equilibrium in still water. The forces generated by gravity and hydrostatic pressure move the vessel along the vertical direction, radiating waves. The shape of the body representing a ship is a prism with rectangular base $L_s \Delta y$. The total ship height is H_s, and the height below the centre of mass G is h_s. We choose $h_s = m_s/(\rho L_s \Delta y)$, so that G will tend to the still water level (SWL). The waves would reflect at the side boundary of the water body and return to the ship, altering its oscillation pattern. To avoid this disturbance, the domain must be larger than $c_w \cdot t_{end} + L_s$, where c_w is the wave speed, t_{end} the simulation end time, and L_s the width of the ship. In this test case, we choose a large water depth H so that the effects of the added mass are reduced. It is important to stress that although we are in deep water, the model behaves equivalently to shallow water conditions. This means that the radiated waves will travel at the same speed $c_w = \sqrt{gH}$. We focus on the movement in the vertical direction only (one degree of freedom, heaving) and we neglect viscosity. The reference solution of the damped-harmonic-oscillator originates from the balance of the forces in (3.8):

$$(m_s + a_s)\ddot{z}_G + b_s \dot{z}_G + c_s z_G = 0 \tag{3.8}$$

where

$$b_s = \frac{2\rho g c_g}{\omega^2} \left(\frac{\xi_w}{z_0}\right)^2 \Delta y \qquad c_s = \rho g A_s \tag{3.9}$$

$$\omega = \frac{2\pi}{T} \qquad T = 2\pi \sqrt{\frac{m_s + a_s}{\rho g A_s}}. \tag{3.10}$$

z_G is the elevation of the ship centre of mass G, $\dot{z}_G$ is the vertical velocity of G, $\ddot{z}_G$ is its vertical acceleration. a_s is the added mass and in this case it is positive. We compute it with the equation derived by Lannes [78]:

$$a_s = \rho\, dy \int_{x_-}^{x_+} \frac{(x - \hat{x})^2}{H}\, dx, \qquad \text{with} \quad \hat{x} = \frac{1}{\int_{x_-}^{x_+} \frac{1}{H}\, dx} \int_{x_-}^{x_+} \frac{x}{H}\, dx \tag{3.11}$$

where x_- and x_+ are the x coordinates of the two points P_- and P_+ where the free surface η intersects the ship, whose coordinates are (x_-, z_-) and (x_+, z_+). We also have $x_+ - x_- = L_s$, so that for this test case $a_s = \rho \Delta y L_s^3/(12H)$. c_s is the spring coefficient, derived from a combination of the Archimedes' principle and Hooke's law. The waterplane area $A_s = L_s \Delta y$ is constant; b_s is the damping coefficient for the heave motion; its expression in (3.9) is derived in linear wave theory considering the transfer of energy from the ship to the waves [69]; ω is the undamped natural frequency; T is the ship oscillation period, which in this test case is taken equal to the wave period. In shallow water the group velocity c_g is constant and equal to the wave speed $c_g = c_w = \sqrt{gH}$, and thus the wavelength is $L_w = T\sqrt{gH}$; z_0 is the ship oscillation amplitude and it corresponds to the initial displacement, while ξ_w is the wave amplitude and usually it is not known a priori. We observe that during a half-period the volume of water displaced by the ship V_s must be equal to the volume of the radiated wave V_w , so for this specific case we can derive an expression of the amplitude ξ_w:

$$(2z_0)\frac{L_s}{2}\Delta y = V_s = V_w = \frac{\xi_w L_w \Delta y}{\pi} \qquad \Longrightarrow \qquad \xi_w = \frac{\pi z_0 L_s}{L_w} \tag{3.12}$$

We substitute c_g, ω and ξ_w in (3.9) and obtain the damping coefficient b_s expressed in terms of known quantities:

$$b_s = \frac{\rho g L_s^2 \Delta y}{2\sqrt{gH}}. \tag{3.13}$$

We define $\sigma = \dfrac{b_s}{2(m_s + a_s)}$ and we rewrite equation (3.8) in a canonical form:

$$\ddot{z}_G + 2\sigma \dot{z}_G + \omega^2 z_G = 0. \tag{3.14}$$

The solution of (3.14) is

$$z_G(t) = z_0 \mathrm{e}^{-\sigma t}\left[\cos(\omega_z t) + \frac{\sigma}{\omega_z}\sin(\omega_z t)\right], \tag{3.15}$$

with $w_z = \sqrt{w^2 - \sigma^2}$. The velocity is then:

$$\dot{z}_G(t) = -z_0 \mathrm{e}^{-\sigma t}\left(\omega_z + \frac{\sigma^2}{\omega_z}\right)\sin(\omega_z t). \tag{3.16}$$

The simulation parameters are given in Table 3.1. The initial configuration can be seen in Figure 3.5 and the results are in Figures 3.6 and 3.7. The oscillation

Symbol	Definition	Value	Unit
η_0	Initial free surface elevation (SWL)	0	m
u	Initial horizontal velocity	0	m/s
b	Bottom	-1000	m
k_{St}	Strickler coefficient	100	$\mathrm{m}^{\frac{1}{3}}/\mathrm{s}$
s	Subcells in each cell	4	-
ϑ	Implicitness factor	1	-
Δt	Time step size (constant)	$1 \cdot 10^{-3}$	s
Δx	Grid size along x	2.5	m
z_0	Initial displacement of the ship	-2	m
m_s	Ship mass	100000	kg
L_s	Ship length	20	m
Δy	Ship width	1	m
H_s	Total ship height	10	m

Table 3.1. Data for the return to equilibrium in deep water test case.

pattern is in good agreement with the reference solution.

We want to provide an interpretation of how the fluid-structure coupling dampens the oscillation implicitly. The dampening is the result of a restoring force that is consistently smaller than what would be required to keep the amplitude of oscillation constant. Such force is a function of the equilibrium position z_{eq} of the ship and the instantaneous value of η at the waterline, the point of intersection of the ship and the waves. For the damped-harmonic-oscillator reference solution, z_{eq} is assumed at the SWL, and thus the instant force is simply proportional to $z_{eq} - z_G(t)$. What makes the oscillator "damped" is the addition to the ODE of the radiation resistance force $b_s\dot{z}_G$, and the additional inertia of the motion in water is introduced by means of the added mass a_s. On the contrary, with a fluid-structure coupling the radiated waves move the position of the waterline, meaning that z_{eq} is time dependent and specifically it oscillates together with η, so the net vertical force is given by (2.37). When the ship is accelerating upwards, η at the waterline decreases and the pressure distribution is concave up. The vertical force is lower than it would be if the equilibrium position was fixed at the SWL, so the oscillation is damped. Of course, when the ship is accelerating downwards the opposite is true: η at the waterline increases and its concavity is down, but the effect is again a dampening of the oscillation. In this specific test case there is vertical symmetry, so when the acceleration of the ship is zero the pressure η inside the ship is almost horizontal everywhere. It follows that in this precise moment the

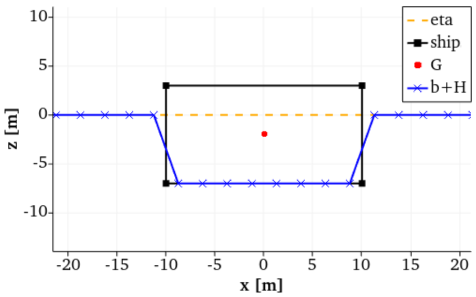


Figure 3.5. Initial configuration of the return to equilibrium test case. The figure shows only the area near the ship, even though the domain is larger and higher

outgoing waves are at their minimum or maximum elevation, called trough and crest. For our specific choice of h_s, when the acceleration is zero it also means that $z_G = \eta$. We can relate the interpretation above to the results in Figure 3.7. We can see that while $z_G < \eta$, which is when the ship is accelerating upwards, η in the middle of the ship is lower than the η at the waterline, which indicates that η is concave up. When $z_G \approx \eta$, the distribution of η is, in first approximation, horizontal. When $z_G > \eta$, η is concave down. We can also see that for the first half of the oscillation period, $\eta <$SWL, which means that the upward force on the hull is always smaller than it would be without the fluid-structure interaction, and this causes the dampening. In the second half oscillation period, $\eta >$SWL, the upward force on the hull is always bigger that it would be without the FSI, so the downward descent is slowed down.
From the time evolution of η in Figure 3.7 we can also get information on the generation of radiated waves, because η has the meaning of piezometric head in the ship region, and of free-surface outside the ship region. The waves in this test case start from the symmetry point, the centre of the ship, and radiate outwards. We know the value of the wave amplitude from equation (3.12), and we can expect that it will be damped of a factor $\mathrm{e}^{-\sigma t}$, as the ship oscillation amplitude. The envelope of η at the ship centre is computed from equation (3.17):

$$\Xi_w(t) = \xi_w \mathrm{e}^{-\sigma t} \tag{3.17}$$

The value predicted by equation (3.17) corresponds to the numerical wave trough and crest. We also observe that the curves of η in the middle of the ship and η at the waterline are almost overlapping; this happens because the wave speed c_w is high, so the time delay $(L_s/2)/c_w$ between the two is small.

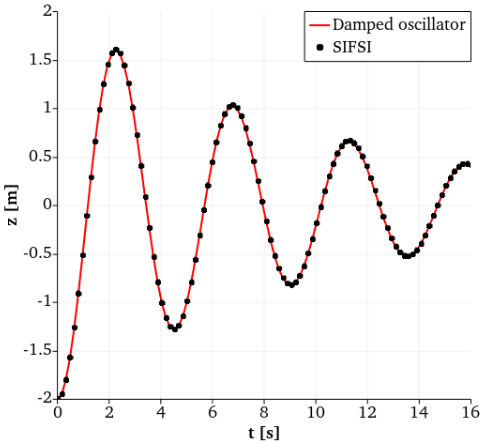


Figure 3.6. Time evolution of the ship elevation in the test case of return to equilibrium in deep water. The elevation of the centre of mass z_G is compared to the expected value from the damped-oscillator solution

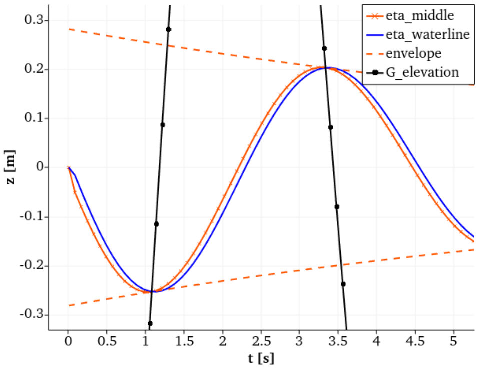


Figure 3.7. Time evolution of η in the middle of the ship and at the waterline, in the test case of return to equilibrium in deep water. The z_G curve is shown only in the range of interest of the vertical axis, where we want to highlight the points of intersection with the η curves; the full curve is in Figure 3.6. The dashed lines are the envelope of the wave amplitude

3.2.2 Return to equilibrium in shallow water

In this test case, we want to look at the effects of a smaller water depth H on the oscillation pattern. The reference solution is formally equal to the solution in Section 3.2.1. With a smaller water depth, the solid boundary is closer to the ship, so we expect an increased contribution of the added mass [20]. The effect of a_s (it's value is always positive), looking at the definitions of T and σ in Section 3.2.1, is to increase the period of oscillation and reduce the damping coefficient. However, this does not necessarily mean that the overall damping is smaller, because from equation (3.13) we see that b_s increases as H decreases. In fact, generally the damping is stronger in shallow water [74]. The parameters for this test case are the same as in the previous one, Table 3.1, with the only exception of setting the bottom at $b = -20$ m. The time evolution of z_G is shown in Figure 3.8. In shallow water the oscillation is more damped than it is in deep water, and we have a good agreement with the analytical solution again. Figure 3.9 displays the time evolution of η in the ship region. Most of the considerations done for the deep water test case in Section 3.2.1 are still true, but there are some important differences. First, the amplitude is bigger, which is the reason for the increased dampening. Second, the wave speed is lower, which means that the peaks of the curves of η are shifted in time. The consequence of this time delay is that the concavity of η is much more pronounced. Also, in this time period the ship is heaving and its motion seems to deform the outgoing wave, reducing its peaks; now, the envelope of equation (3.17) predicts well only η in the middle of the ship.
An interesting observation can be done if we separate the dampening into two components, one being the variation of η at the waterline and the second one being the concavity of η. The former is equivalent to assuming that η is a straight line connecting the intersection points P_- and P_+, the latter is the deviation from this straight line. In this test case, thanks to the symmetry of the problem, we can separate the two components also in the numerical simulation: when computing the vertical force, we assume η constant inside the ship and equal to the value at the waterline. If we do that, we are removing the component due to the concavity of η. What we observe from the results in Figure 3.10 is that this numerical solution matches the reference solution with $a_s = 0$, meaning that the period of oscillation is smaller and the oscillation is more damped. This experiment would suggest that the "linear" component of η has the equivalent effect of the damping coefficient b_s in the ODE, while the curvature of η has the equivalent effect of the added mass a_s. We can do one more observation on the concavity of η, which is related to its steepness. From the momentum equation (2.13) we derive the expression (3.18) for the

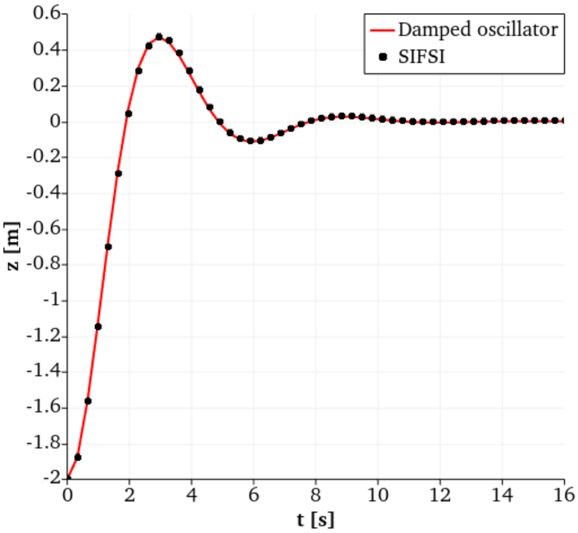


Figure 3.8. Time evolution of the ship elevation in the test case of return to equilibrium in shallow water. The elevation of the centre of mass z_G is compared to the expected value from the damped-oscillator solution

gradient of pressure.

$$\frac{\Delta\eta}{\Delta x} = \frac{q^* - q^{n+1}}{gH\Delta t}. \tag{3.18}$$

The gradient depends on the space and time flux derivatives and it is inversely proportional to the water depth. It follows that in shallow water the curvature of η is more pronounced, which is what we observed from the numerical simulations.

3.2.3 Heaving due to incoming waves

In this test case, we analyse the response of a trapezoidal ship to the influence of regular incoming waves produced by a wavemaker. The sinusoidal waves are generated at the west boundary of the water body containing the ship, forming a numerical wave tank. At the initial time, the ship floats in its hydrostatic equilibrium position close to the west boundary of the tank. The ship has a flat bottom of length L_{bottom} and oblique sides inclined of an angle α_s from the vertical. Due to the shape of the ship, the immersed volume is not directly proportional to the vertical displacement. We choose the position of the east boundary to make sure the reflected waves do not reach back the ship during the simulation time. The water depth is the same used in the

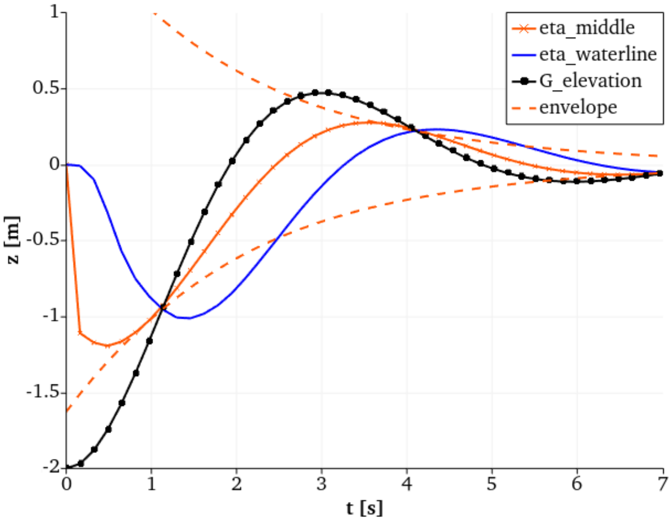


Figure 3.9. Time evolution of η in the middle of the ship and at the waterline, in the test case of return to equilibrium in shallow water. The dashed lines are the envelope of the wave amplitude

return-to-equilibrium in shallow-water test case, in Section 3.2.2. Again, we concentrate on the ship movement with one degree of freedom, namely on the vertical heaving. The physical reference case is a forced and damped oscillator, described by the second order ODE (3.19), which is a generalization of equation (3.8). ODE (3.19) allows for both rectangular and trapezoidal shapes of the ship, an arbitrary initial still water level, an arbitrary relative position of the centre of mass and a delayed external forcing by regular waves. We apply the so-called small-body approximation [92], which states that the length of the floating body is much smaller than the wave length ($L_{wl} \ll T_w \cdot c_w$). The solution is found numerically using a fourth-order Runge-Kutta scheme.

$$\begin{cases} (m_s + a_s)\ddot{z}_G + b_s\dot{z}_G + c_s z_G - \epsilon z_G^2 = c_s\Big(\eta_0 + \dfrac{H_w}{2}\sin(\omega_w(t-t_0)) + h_s\Big) \\ \qquad\qquad\qquad\qquad\qquad\qquad -\epsilon\Big(\eta_0 + \dfrac{H_w}{2}\sin(\omega_w(t-t_0)) + h_s\Big)^2 - m_s g \\ \dot{z}_G(t = t_0) = 0 \\ z_G(t = t_0) = \eta_0 - D_{eq} + h_s \end{cases} \tag{3.19}$$

where m_s is the ship mass, η_0 is the free-surface elevation at the start of simulation, H_w is the incoming wave height, $\omega_w = 2\pi/T_w$ is the wave frequency, h_s is the vertical distance between the centre of mass G and the bottom of

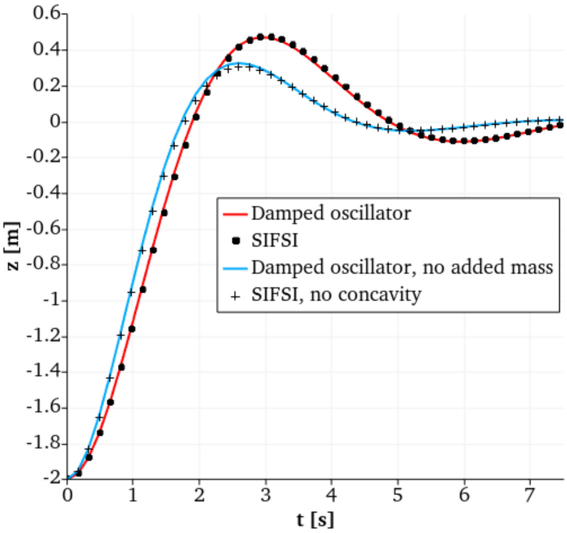


Figure 3.10. Effect of the concavity of η on the ship motion. If only the linear component of η is used to compute the vertical force, the numerical results agree with the reference solution with $a_s = 0$

the ship. The added mass a_s and the damping coefficient b_s are defined as in Section 3.2.1. For b_s, we use the water depth defined by the current free-surface elevation near the ship, and as length the length at the water level $L_{wl} = L_{bottom} + 2\tan(\alpha_s)D_{eq}$, where the draft at equilibrium D_{eq} is the solution of:

$$\tan(\alpha_s)D_{eq}^2 + L_{bottom}D_{eq} - \frac{m_s}{\rho\Delta y} = 0 \tag{3.20}$$

The terms c_s and ϵ are:

$$c_s = \rho g\Delta y \left[L_{bottom} + 2\tan(\alpha_s)\left(\eta_0 + \frac{H_w}{2}\sin(\omega_w(t-t_0)) + h_s\right)\right] \tag{3.21}$$

$$\epsilon = \rho g\Delta y \tan(\alpha_s) \tag{3.22}$$

The reference solution is delayed in time because the incoming waves generated at the west boundary reach the ship side at

$$t_0 = \frac{1}{c_w}\left[x_G - \left(\frac{L_{bottom}}{2} + \tan(\alpha_s)D_{eq}\right) - x_W\right].$$

When the waves pass by the ship position, the instantaneous equilibrium position changes in time and a restoring force acts on the hull, but the ship stays outside the position of equilibrium. As a result, the wave elevation and the ship response are shifted in time of t_{ws}. This delay in time can be obtained, with accuracy sufficient for our purposes, from the analytical solution of an equivalent problem with a rectangular ship:

$$t_{ws} = \frac{1}{\omega_w} \arctan\left(\frac{b_s \omega_w}{(m_s + a_s)\omega_w^2 - 0.5\rho g \Delta y (L_{wl} + L_{bottom})} \right) \tag{3.23}$$

It is possible to show, by substituting the value of b_s in (3.23) and assuming ω_w and $\tan(\alpha_s)D_{eq}$ are small, that this time delay is approximately the time necessary for the wave to travel half of the ship length at the waterline:

$$t_{ws} \approx -\frac{1}{c_w}\left(\frac{L_{bottom}}{2} + \tan(\alpha_s)D_{eq}\right) \tag{3.24}$$

To check the presence of the time delay t_{ws}, we will plot the motion of a ship always in equilibrium with the current wave elevation at the west ship side, which is equivalent to a simple harmonic oscillator with a heave amplitude and period equal to the ones of the incoming wave:

$$z_G(t) = z_G(0) + \frac{H_w}{2}\sin(\omega_w(t - t_0)) \qquad \text{for } t \geq t_0 \tag{3.25}$$

The data for the simulation are summarized in Table 3.2, while a view of the initial conditions is shown in Figure 3.11. The numerical results are compared to the reference solutions in Figure 3.12. The numerical and reference solution agree fairly well with each other. This is true starting from the first oscillation, when the ship has to adjust to a sinusoidal motion from the initial equilibrium conditions. We observe that there is the time delay between the wave passage and the ship response, which for this test case was predicted as $t_{ws} = 1.33$ s. We notice that the oscillation pattern has the same amplitude of the incoming

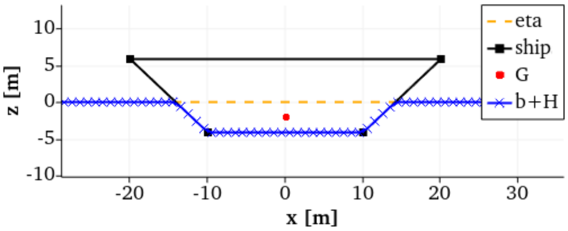


Figure 3.11. Initial configuration of the heaving due to incoming waves test case. The figure shows only the area near the ship, even though the domain is larger and higher

Symbol	Definition	Value	Unit
η_0	Initial free surface elevation (SWL)	0	m
u	Initial horizontal velocity	0	m/s
b	Bottom	-20	m
k_{St}	Strickler coefficient	100	$\text{m}^{\frac{1}{3}}/\text{s}$
s	Subcells in each cell	4	-
ϑ	Implicitness factor	1	-
Δt	Time step size (constant)	0.01	s
Δx	Grid size along x	1	m
z_0	Initial vertical displacement of the ship	0	m
m_s	Ship mass	100000	kg
L_{bottom}	Ship length at bottom	20	m
L_{top}	Ship length at top	40	m
Δy	Ship width	1	m
H_s	Total ship height	10	m
H_w	Wave height	1	m
T_w	Wave period	24	s

Table 3.2. Data for the heaving due to incoming waves test case.

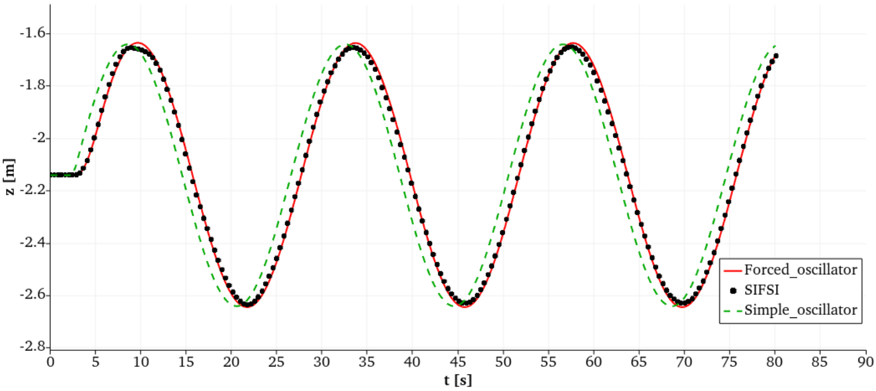


Figure 3.12. Time evolution of the ship elevation in a test case with incoming waves in shallow water. The elevation of the centre of mass z_G is compared to the expected value from the forced and damped oscillator solution. In addition, the simple oscillator solution shows the time delay between the wave elevation and the ship response

waves; the natural oscillation of the ship, whose time period is computed from equation (3.10) and is equal to $T_s = 5.7$ s, is damped. There are small errors at the highest and lowest elevation points. Namely, the oscillation amplitude is slightly smaller compared to the reference solution. The probable cause is that the problem solved with the new model is more general than the reference problem. Some additional physical and numerical processes deform the waves in the simulation: first, the waves that hit the ship are partially reflected back, so less pressure acts on the hull; second, even though the small-body assumption holds, in the ship region we never have uniform values of the piezometric head, which means its average value is never as high as the value in the reference problem; third, while the waves are travelling from the west boundary to the ship, they are affected by numerical dissipation and their peaks are damped. These three effects, which would slightly reduce the oscillation amplitude, are not included in the reference ODE (3.19).

3.3 Horizontal motion

In this Section, we show the application of the ideas introduced in Section 2.13 and their effect on the pressure oscillation.

We start from the problem of Section 2.13.1 and we activate one of the algorithms, one at the time. This helps to recognize and understand in isolation the mechanisms of action of the algorithms and have a qualitative measure of their consequences. We remind here the main test parameters, valid for the Sections from 3.3.1 to 3.3.6: grid size of $\Delta x = 6$ m, timestep size of $\Delta t = 0.2$ s, ship speed of $u_s = 5$ m/s moving left, the subgrid is equivalent to the grid. At the boundaries, non-reflecting Dirichlet boundary conditions are applied. The ship is in hydrostatic equilibrium at the start of the simulation. The only allowed motion is horizontal, i.e. no vertical translation or rotations.

For an easier interpretation of the numerical results, every test case is compared with the same two results. The first reference case is the original test of Section 2.13.1, which is a generic setup with no modifications. The second reference is the particular case of Section 2.13.2, where the relationship $u_s = \Delta x/\Delta t$ holds by choosing $\Delta t = 1.2$ s. These two test cases manifest the biggest and smallest η oscillations; they will be referenced later and in the figures/methods as "void" and "gridspeed" cases, respectively.

3.3.1 Smooth ship movement

Figure 3.13 shows the results of the application of the smooth-ship-movement algorithm of Section 2.13.3. Given our test parameters, the ship enters a new cell every 6 time iterations. At first approximation, we can say that the volume

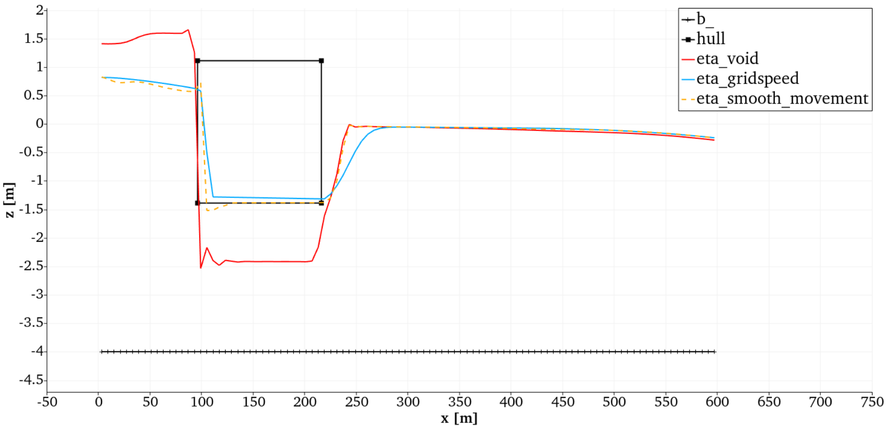


Figure 3.13. Application of the smooth-movement algorithm. In this setup, the ship moves in a new cell every six time iterations. Setup: $\Delta x = 6$ m, $\Delta t = 0.2$ s.

displaced from each cell is divided in six parts and at every time iteration one part is pushed out of the ship region. In the absence of this algorithm, we had the six parts all pushed out in one time iteration, while during the other five iterations the volume of water in the cell remained constant.
The amplitude of the pressure oscillation is smaller. The bow wave amplitude decreases. The generation of the void below the ship is greatly reduced and remains constant in time.
Given these results, the smooth-ship-movement algorithm seems worth keeping always active.

In Figure 3.14, the results from the application of the smooth-movement algorithm are compared with the results of a subgrid convergence study. The grid remains always the same. It can be seen how as the subgrid is refined, the results converge towards those of the smooth-movement simulation. Indeed, with a fine subgrid the displaced volume computation gets more precise.

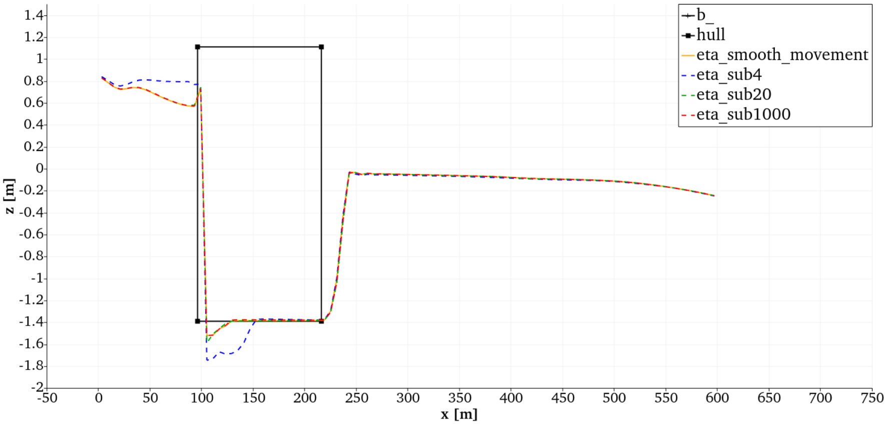


Figure 3.14. Convergence of the subgrid solution to the smooth-movement solution. The subgrid solution is shown for the choice of 4, 20 and 1000 subcells per cell.

3.3.2 Roof relaxation

Figure 3.15 shows the results of the application of the roof-relaxation algorithm of Section 2.13.4. Here, the water is allowed to penetrate the hull in proportion to the pressure and the relaxation's intensity. We chose to apply the relaxation to the first 5 cells near the waterline, with space-varying intensity. At the waterline it is the strongest and it decays going inside. Since the ship region is composed of 20 cells, there are 10 cells in the middle that are unaffected by the relaxation and consequently are fully-pressurized.

We notice that the roof relaxation is effective in the stabilization of the pressure and is able to prevent the growth of the void below the hull.

Due to the mechanism of action of the relaxation, the position of the b+H line is peculiar. The water entered the ship volume especially at the bow, where the pressure is high. At the stern, the occupied volume is much lower; nonetheless, it is enough to keep the water close to the hull, which is the behavior seen at grid convergence.

In Figure 3.16, a grid convergence study is performed with the roof-relaxation algorithm active. It can be seen in particular how the region interested by the relaxation shrinks with the cell size.

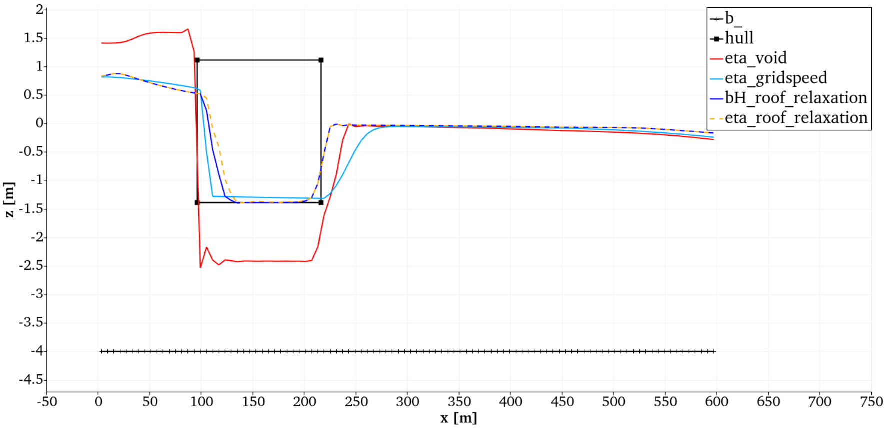


Figure 3.15. Application of the roof relaxation.

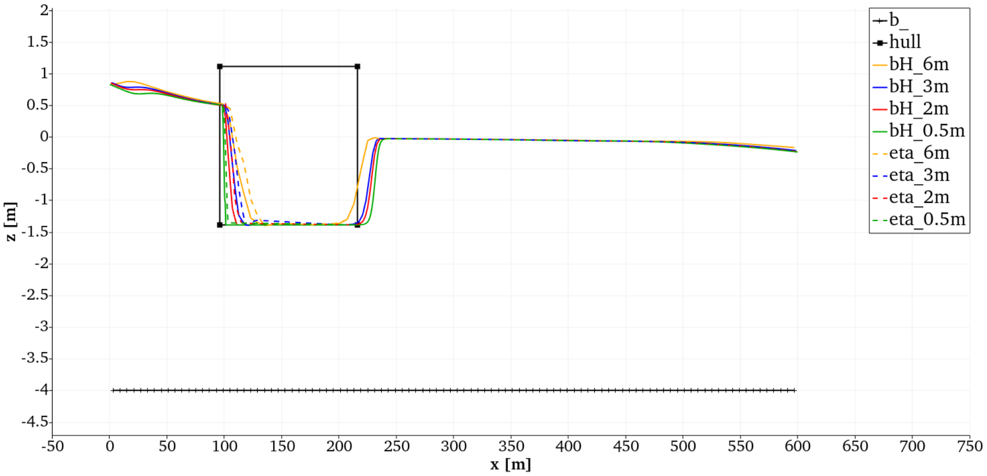


Figure 3.16. Grid convergence of the roof-relaxation method. The solid lines are the free-surface elevations; in the ship region they can penetrate the hull because the relaxation alters the depth function. Setup: variable Δx, $\Delta t = 0.2$ s.

3.3.3 Smooth ship shape

Figure 3.17 shows the results of the application of the modification of the input hull, as explained in Section 2.13.5. The hull's shape is obtained after 5 smoothing iterations with a filter of size 5. The profile becomes more rounded and the total jump of the original vertical wall is split in many parts.
The alteration of the shape impacts the equilibrium position of the ship, which would be lower, so it is is impossible to keep both the immersed volume and the initial draft equal to the ones of the other test cases. Here we are not interested in the floating dynamics, so we choose to maintain the same initial draft, which means the volume displaced during the horizontal motion is almost the same. The results suggests a good ability of this solution to reduce the pressure peaks at the bow. Contrarily to the other test cases, the free-surface behind the ship is undulated; every small jump at the bow generates a small pressure peak and this pattern emerges. Although smoothing the shape is beneficial to the suppression of the oscillations, it seems insufficient if applied by itself, because a portion of the solution converges differently than what we expected. Considering that it also changes the floating dynamics of the ship, we suggest to not use this smoothing as first choice.
One may consider the case in which this rounded hull is not the product of a smoothing but is the input hull. In this case, the application of the other techniques is enough to suppress these shape-induced undulations. To test that, we applied the smooth-shape and the smooth-movement algorithms together. The results can be seen in Figure 3.18.

In Figure 3.19, a grid convergence study is done with the smooth-shape algorithm. As the cell size decreases, the smoothed shape gets closer and closer to the original shape. Also, the unusual oscillations seen for the coarser grid disappear in the other cases.

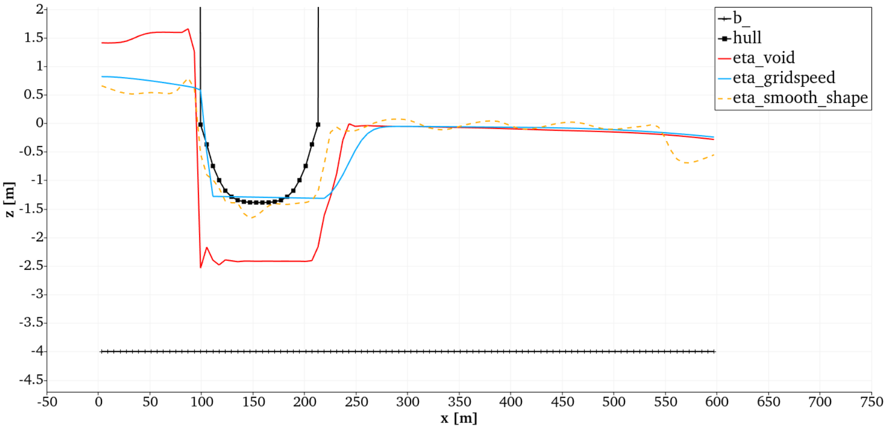


Figure 3.17. Application of the smooth-shape algorithm. The input shape is rectangular. Setup: $\Delta x = 6$ m, $\Delta t = 0.2$ s.

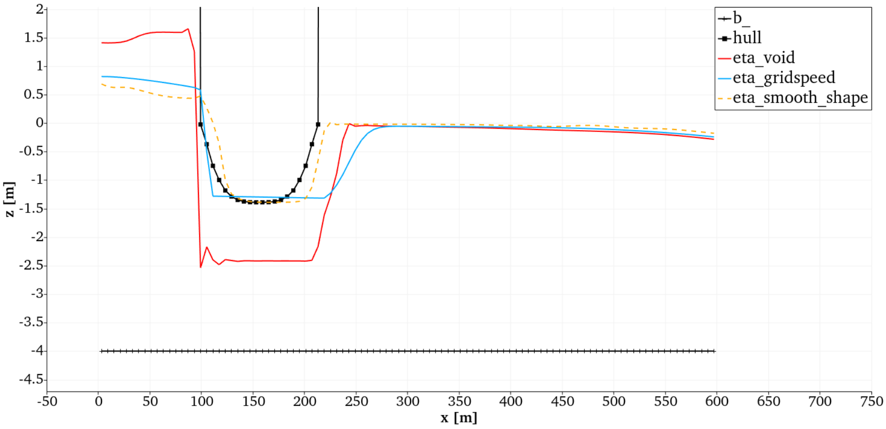


Figure 3.18. Concurrent application of the smooth-shape and smooth-movement algorithms. The undulation of the free-surface seen in Figure 3.17 is now under control.

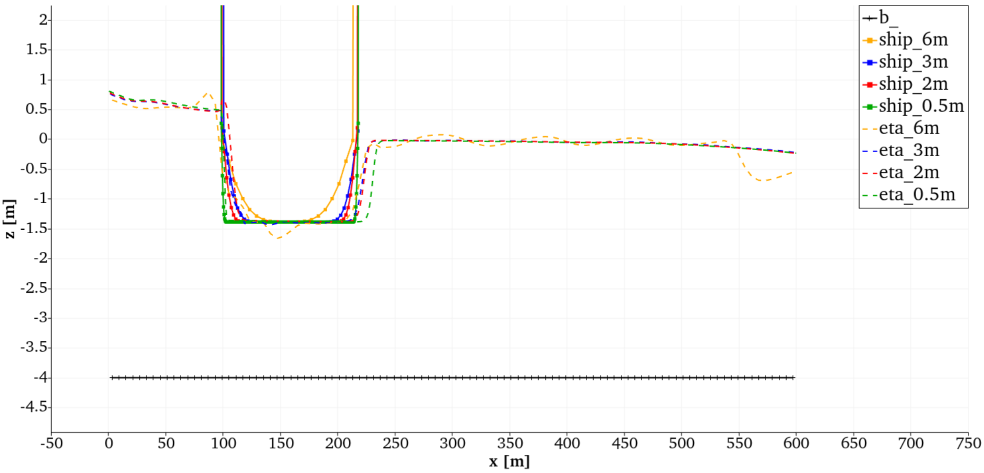


Figure 3.19. Grid convergence of the smooth-shape algorithm. The solid lines are the ship hulls; they converge to the input shape as the grid is refined. Setup: variable Δx, $\Delta t = 0.2$ s.

3.3.4 Cell merging

Figure 3.20 shows the results of the application of the cell-merging method, as explained in Section 2.13.6. We remind that before the nested-Newton algorithm starts to find the solution of the system, the newly-displaced volume term is removed from pressurized cells and it is added to the nearest free-surface cell.
The results are very promising, the pressure oscillations are strongly reduced and the void below the hull is stably filled with water. If we compare the η profile with the grid-speed solution and the other methods' solutions, it appears that our choice of destination cell as the nearest is adequate to maintain the physical relevance of the results.

In Figure 3.21, a grid convergence study is done in conjunction with the application of the cell-merging method. The results show little differences in terms of free-surface profile, demonstrating a good efficacy of the cell-merging method.

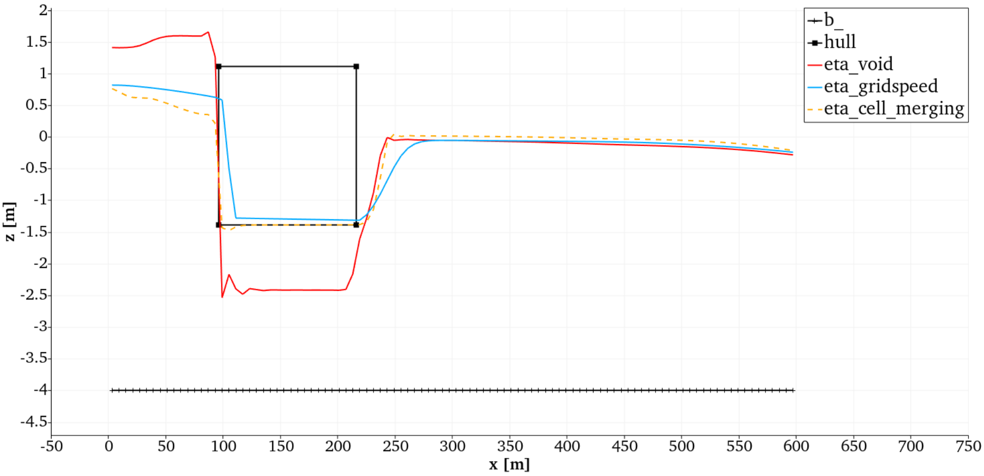


Figure 3.20. Application of the cell-merging technique.

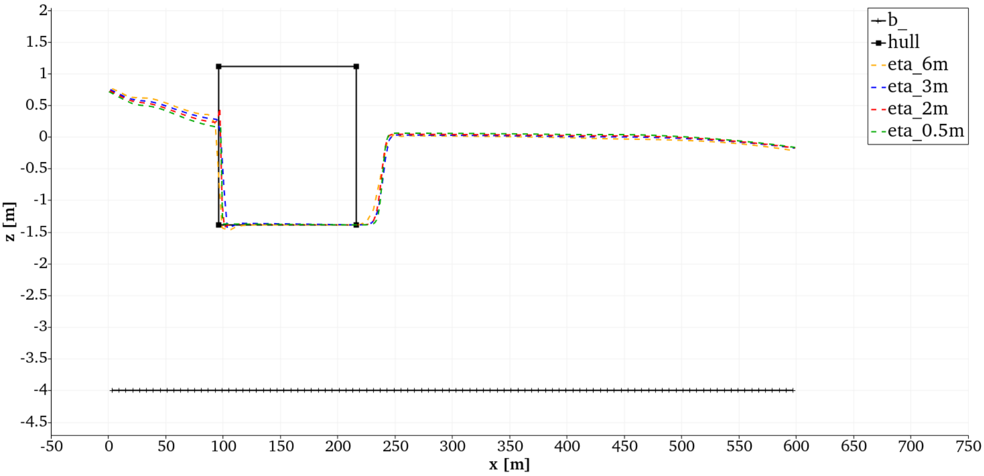


Figure 3.21. Grid convergence of the cell-merging technique. Setup: variable Δx, $\Delta t = 0.2$ s.

3.3.5 Rusanov dissipation

In this Section, we apply the Rusanov dissipation algorithm, as explained in Section 2.13.7.
This technique adds additional numerical dissipation to the scheme and penalizes pressure gradients. It requires a somehow arbitrary choice of the parameter α_d. In Figure 3.22 we computed multiple simulations where increasing values of α_d are used to give an idea of the parameter sensibility and the differences it induces in the solution. For a sufficiently high α_d coefficient, the amplitude is so small that the pressure on the hull is always positive and the void-generation is completely removed.

The results illustrated in Figure 3.23 are interesting because they show a strong potential of the dissipation to influence the solution, especially in the ship region. Indeed, for a sufficiently high α_d, the pressure η gets close to being a straight line connecting the waterline points at the opposite sides of the ship, and the oscillations have a smaller amplitude. This is a behavior that was not achieved by the other methods, whose typical profile was a quick drop at the bow followed by a constant value.
Outside of the ship region the free-surface elevation is overall consistent with the other results, with the bow wave converging towards the gridspeed bow-wave elevation. A small but not negligible difference can be observed behind the ship, where the level does not return to the equilibrium elevation but it stops at a lower level; also, in the front the convergence is from the top, while in the other cases it was from the bottom. These two statements suggest that the dissipation, in its reduction of the pressure gradients, is restricting the flow below the ship.
In terms of stability during the horizontal motion, the Rusanov dissipation is the best among these five tools in keeping the solution self-similar, i.e. there are not much oscillations when a new cell is entered. The possibility to choose the Rusanov-dissipation intensity is a strength, but it is also a weakness because it is not a feature that can simply be turned on or off. Its use requires a choice whenever the problem parameters vary sensibly. This choice can be guided by an observation of the stability of the solution or by the comparison with reference pressure data.

In Figure 3.24, the Rusanov dissipation is applied at different grid resolutions. The results outside of the ship region are very consistent with each other. In the ship region, the pressure converges non-monotonically.

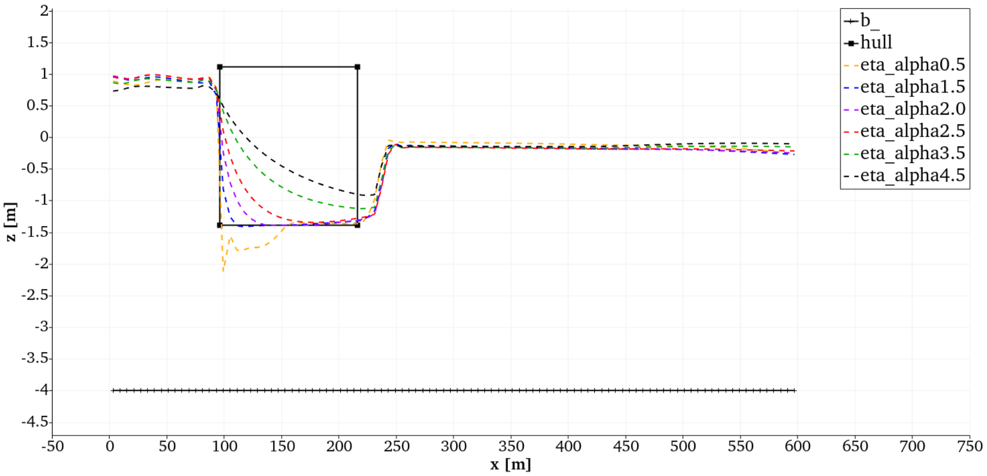


Figure 3.22. Comparison of η for different values of α_d. Setup: $\Delta x = 6$ m, $\Delta t = 0.2$ s.

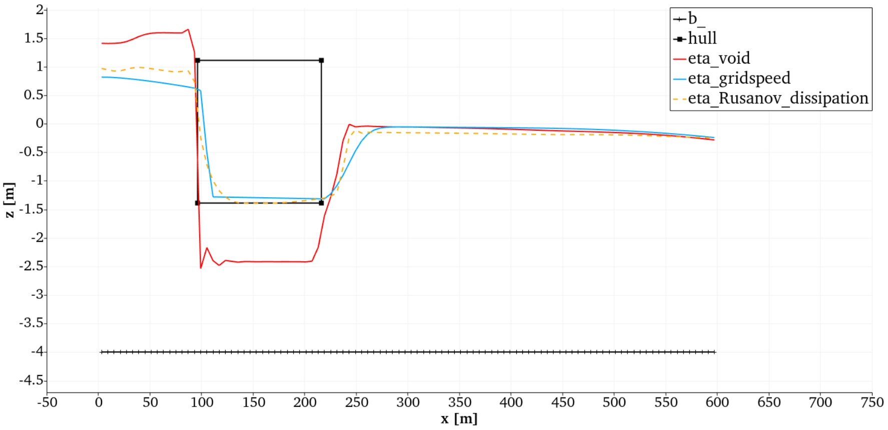


Figure 3.23. Application of the Rusanov dissipation. Setup: $\Delta x = 6$ m, $\Delta t = 0.2$ s, $\alpha_d = 2$.

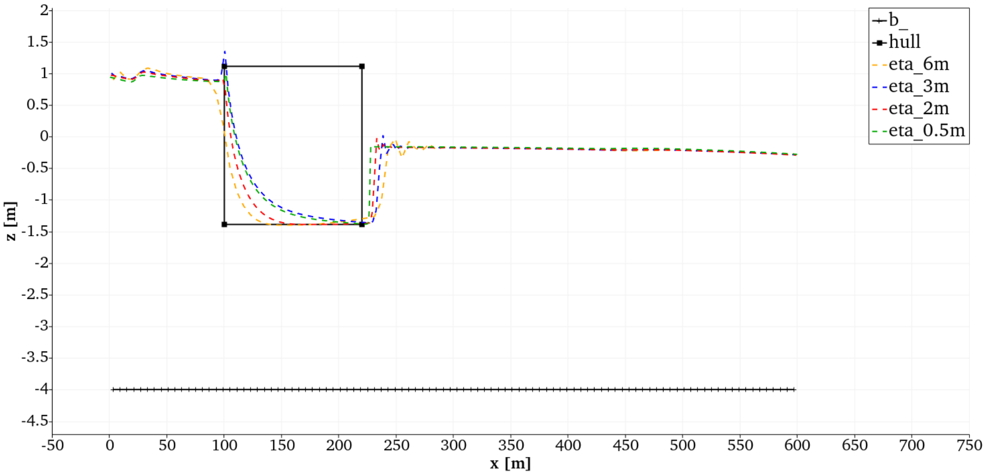


Figure 3.24. Grid convergence of the Rusanov dissipation. Setup: variable Δx, $\Delta t = 0.02$ s, $\alpha_d = 2$.

3.3.6 Combination

Figure 3.25 shows the results of the concurrent application of all the techniques studied from Section 2.13.3 to Section 2.13.7. In Section 2.13.8 we gave an overview of the mechanisms of interation of the different solutions.
Thanks to the results of the individual applications provided in the Sections from 3.3.1 to 3.3.5, we can recognize the use of some of the techniques from a simple visual observation of Figure 3.25; in particular: the absence of vertical walls thanks to the smooth-shape algorithm, the penetration of the water at the bow due to the roof-relaxation method, and the straightening of η in the ship region from the application of the Rusanov dissipation. The influence of the smooth-movement algorithm and the cell-merging method is less evident.

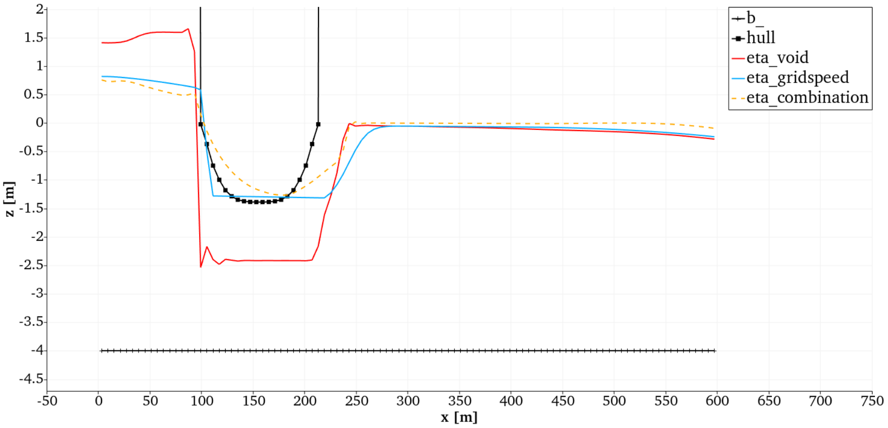


Figure 3.25. Concurrent application all the techniques developed to attenuate the pressure oscillations. Setup: $\Delta x = 6$ m, $\Delta t = 0.2$ s.

3.4 Validation of the waves, squat and return flow

In this Section, we apply the frictional resistance as explained in Section 2.18. The test case we choose is the passage of a ship in a straight canal.
We recreate the setup of the field experiment performed by BAW in 2002 in Wesel-Datteln Kanal (WDK) in the German federal state of North Rhine-Westphalia. Here, a ship with length of 105 m, width of 11 m and draft of 2.8 m was moved at different speeds and it was placed at different distances from the mid-channel axis, for a total of nine configurations. They measured the sinkage, the current velocity and the ship-induced waves height. Measurements like these have great value to us because they allow us to validate our model. The channel cross section is trapezoidal. The central part has a width of 33.4 m, while the banks have a slope of 1 : 2.43. The maximum water depth is 3.95 m. Figure 3.26 shows the channel's bathymetry and the initial positions of the ship.

In the numerical implementation, the water is initially at rest, and the ship is at hydrostatic equilibrium on one side of the domain. To avoid the generation of a large bow wave due to a sudden acceleration, the speed of the ship is increased gradually up to the target value. The boundary conditions are non-reflecting Dirichlet. We always make use of the smooth-movement algorithm of Section 2.13.3. Horizontal viscosity is disabled. We perform 2D$_{xy}$ simulations, i.e. with one vertical layer. We use a Gauckler-Strickler coefficient of $k_{St} = 40$ m$^{\frac{1}{3}}$/s. The total duration of the simulation and the domain size are chosen to reach a condition of stationarity, meaning that the target speed has been

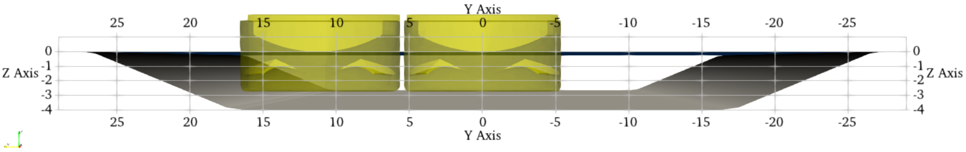


Figure 3.26. Axonometric view of the cross section and the two ship positions for the WDK test cases. Grid values in meters.

reached, the velocity field around the ship is fully developed and the ship has a stable vertical position. The free-surface elevation is sampled close to the exit boundary, at the distance of 21.5 m from the mid-channel axis. We make use of the Referenzschiff's STL, whose main dimensions have been adjusted to correspond to the ship used in the field. Among the nine position/velocity configurations of the field experiment, we show here the results for the four extremes, as summarized in Table 3.3.

Case	Distance from mid channel [m]	Speed [m/s]	Froude number [-]
C-V1	0	2.384	0.383
C-V3	0	2.621	0.421
OBT-V1	11.46	2.179	0.350
OBT-V3	11.46	2.677	0.430

Table 3.3. Description of the position/velocity combinations we used to test our model, as done by Doychev et al. [48].

3.4.1 WDK without skin friction

In this Section, we show the results of the four test cases described in Table 3.3 without the application of the frictional resistance of Section 2.18. We present the comparison between the field measurements and our numerical results. In Table 3.4 there are the squat and return-flow velocity, while in Figures 3.32, 3.33, 3.34 and 3.35 the free-surface elevation at the gauge point is shown. A top view of the velocity field is shown in Figures 3.27, 3.28, 3.29 and 3.30.

The behavior of the squat, return flow and ship-generated waves with the ship speed is credible, as their proportionality is physically correct. We observe a general underestimation of all these three variables. This correlation is not

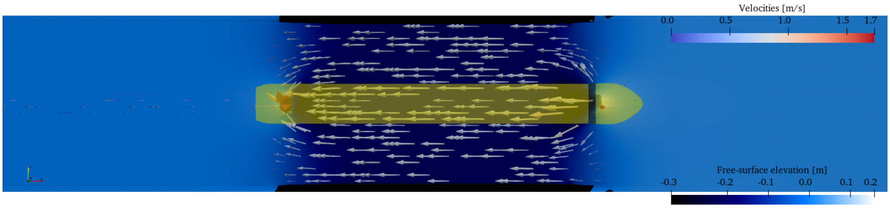


Figure 3.27. Velocity field at stationary conditions. Test case C-V1 without skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s.

surprising in itself, as they are related to each other: if we are at steady state and the drawdown is smaller than expected, the cross-sectional wet area is larger and then the velocity of the return current lowers; with a smaller drawdown and a lower velocity, the squat also lowers. However, the origin of this underestimation is not completely clear, further mesh refiements do not bring great improvements.

For all the test cases, the wave-passage duration is in perfect agreement with the reference data, which is roughly equivalent to the ratio of the ship length and the ship speed L_s/u_s.

When the ship is closer to one of the banks, the ship-bank interaction is evident in a loss of symmetry of the flow field.

Case	Measured squat [m]	Computed squat [m]	Measured return-flow velocity [m/s]	Computed return-flow velocity [m/s]
C-V1	0.28	0.21	1.00	0.72
C-V3	0.48	0.29	1.90	0.90
OBT-V1	0.17	0.18	1.30	0.80
OBT-V3	0.49	0.36	1.60	(0.80-2.60)

Table 3.4. Comparison of the measured and computed squat and return-flow velocity. Test cases without skin friction. Measured data from Doychev et al. [48].

For the test case C-V1, Figure 3.36 shows the results of a grid refinement and the application of the nonhydrostatic algorithm. The refining of the grid in the hydrostatic case and the inclusion of dispension effects for the coarser grid slightly improve the drawdown elevation. The simulation with a fine grid and dispersion effects displays strong oscillations at the stern wave that are related to the appearance of vortices, as seen in Figure 3.31.

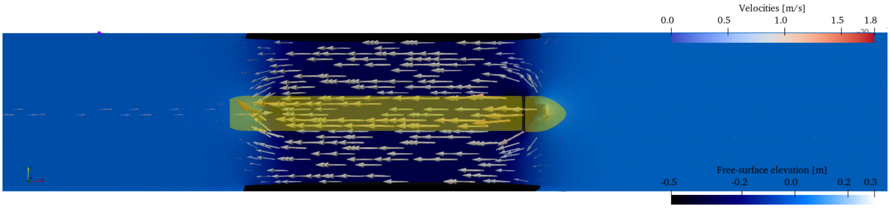


Figure 3.28. Velocity field at stationary conditions. Test case C-V3 without skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s.

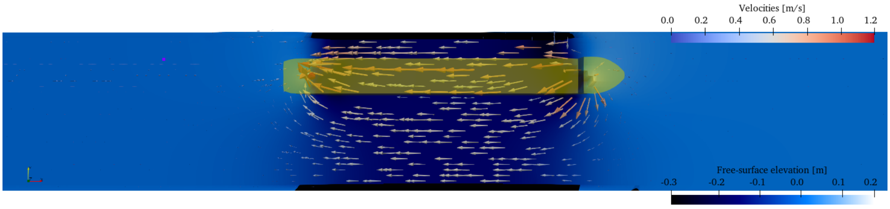


Figure 3.29. Velocity field at stationary conditions. Test case OBT-V1 without skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s.

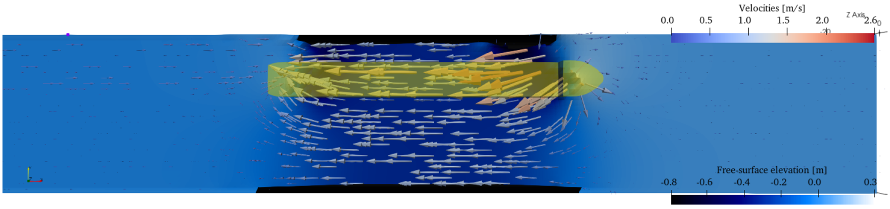


Figure 3.30. Velocity field at stationary conditions. Test case OBT-V3 without skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s.

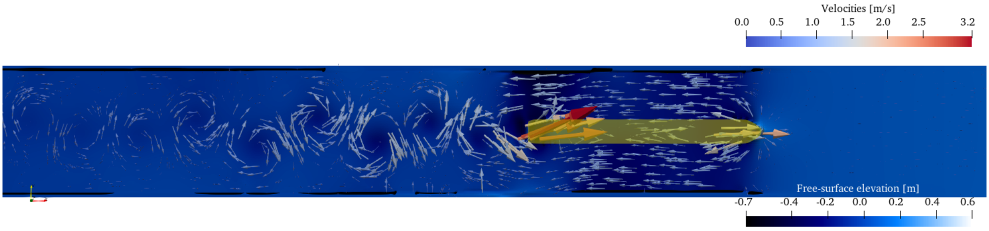


Figure 3.31. Velocity field at stationary conditions in the case of grid refinement and application of the nonhydrostatic algorithm. Test case C-V1 without skin friction. Setup: $\Delta x = 0.8$ m, $\Delta y = 1$ m, subgrid 3x3, $\Delta t_{max} = 0.5$ s.

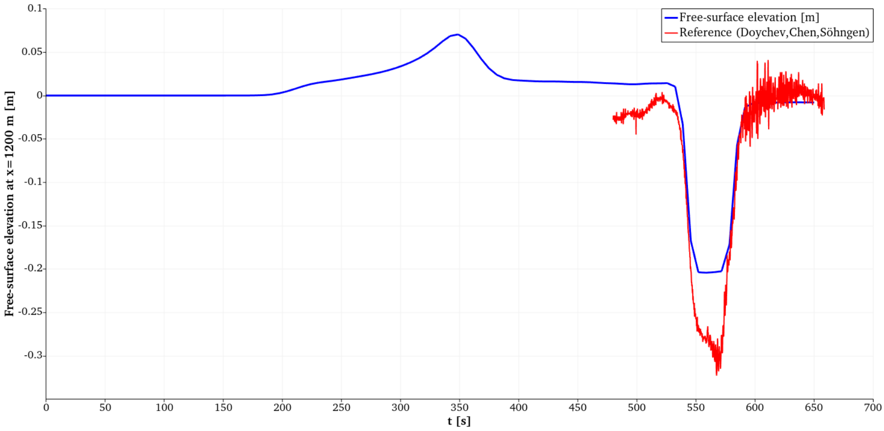


Figure 3.32. Free-surface elevation compared to the value measured at the gauge. Test case C-V1 without skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s.

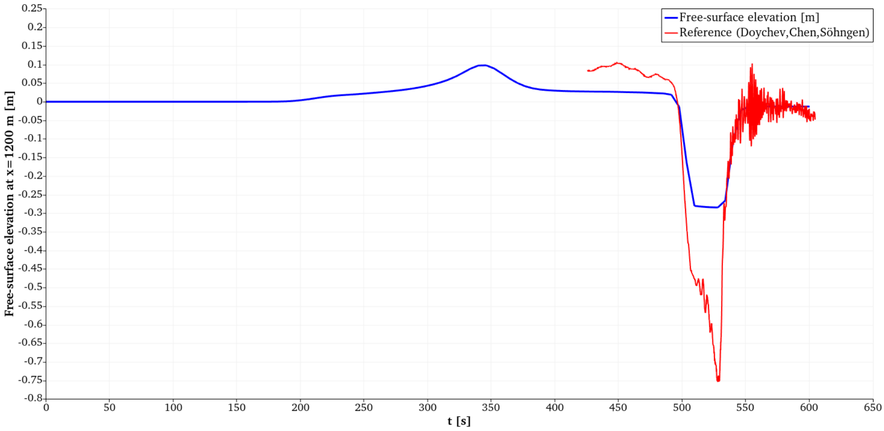


Figure 3.33. Free-surface elevation compared to the value measured at the gauge. Test case C-V3 without skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s.

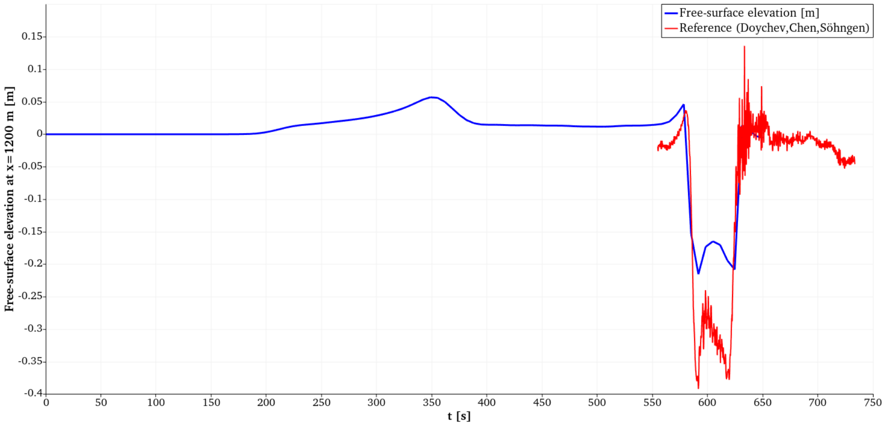


Figure 3.34. Free-surface elevation compared to the value measured at the gauge. Test case OBT-V1 without skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s.

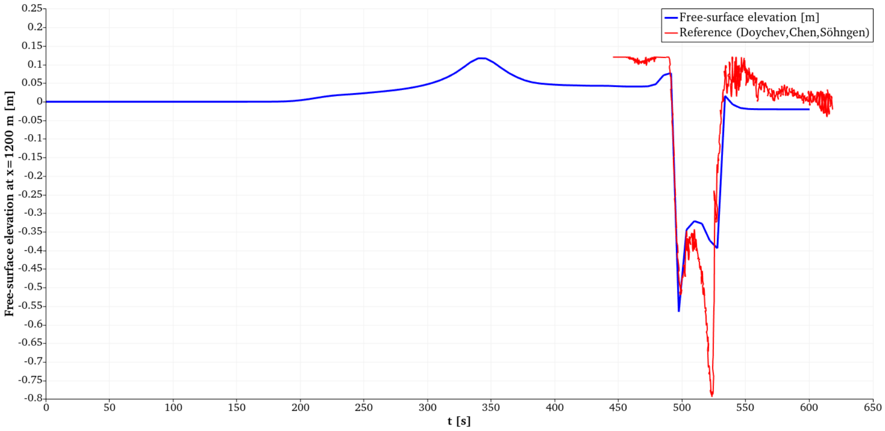


Figure 3.35. Free-surface elevation compared to the value measured at the gauge. Test case OBT-V3 without skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s.

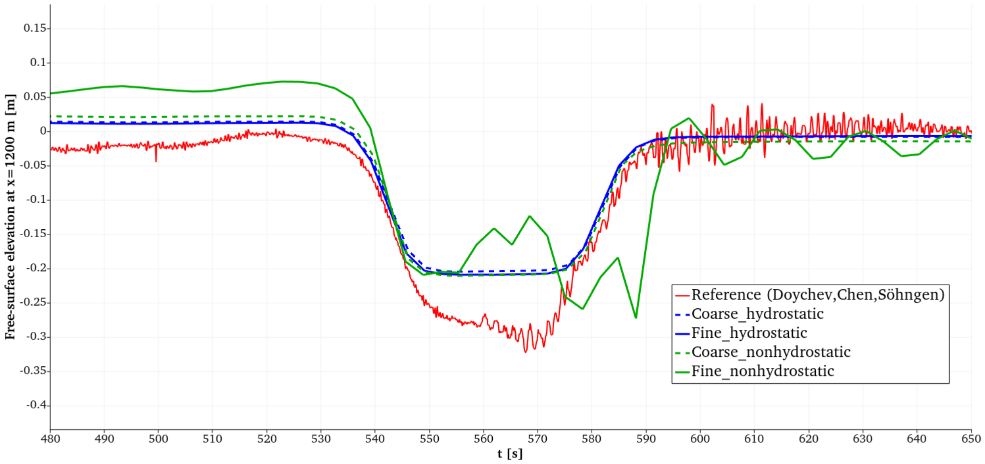


Figure 3.36. Comparison of the free-surface elevations in the case of grid refinement and/or application of the nonhydrostatic algorithm. Test case C-V1 without skin friction. $\Delta t_{max} = 0.5$ s. Setup "coarse": $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4. Setup "fine": $\Delta x = 0.8$ m, $\Delta y = 1$ m, subgrid 3x3.

3.4.2 WDK with skin friction

In this Section we repeat the test cases of Section 3.4.1, but with the skin friction explained in Section 2.18. We explained there that the skin friction has the potential to affect the three variables we consider for this validation. For every test case, the skin friction parameter γ_{sf} is calibrated to get a good comparison with the reference wave pattern.

It is clear that the application of the skin friction has a substantial impact on the numerical results. The ship-generated waves are generally improved, with values of the drawdown that are close to the measurements. The return flow varies more spatially, getting near the reference values. The ship squat is the only variable that seems to be negatively affected by the use of the skin friction.

The additional resistance draws more water to the front, generating a big bow wave. After the initial ship acceleration, the ship and the wave profile arrive to a balance where the drag action of the frictional force is in equilibrium with the action of the free-surface gradient. This mechanism explains why, compared to the frictionless case, the bow wave is higher and the stern wave is lower. There is a proportionality between this difference in free-surface elevation and the γ_{sf} we use. However, an analysis of the generated waves shows that it is highly nonlinear. Starting from $\gamma_{sf} = 0$, which is the friction-less case, and increasing its value, there is a weak effect on the solution. Then, by getting closer to a critical value $\gamma_{sf,crit}$, the solution changes dramatically and a deep trough emerges in the stern region. This $\gamma_{sf,crit}$ varies with the setup. As γ_{sf} continues to increase the trough deepens, but not indefinitely; after a certain point the solution is virtually insensible to an increase of γ_{sf}.

Another effect of the skin friction is the prolongation of the wave passage, which is more extended than the friction-less case, especially for the stern wave. If we observe the velocity field behind the ship, a pattern of vortices becomes visible, and it resembles a von Kármán vortex street. It is only visible when the ship is in the middle of the channel; when it is closer to the bank, the ship-bank interaction inhibits the formation of the vortices.

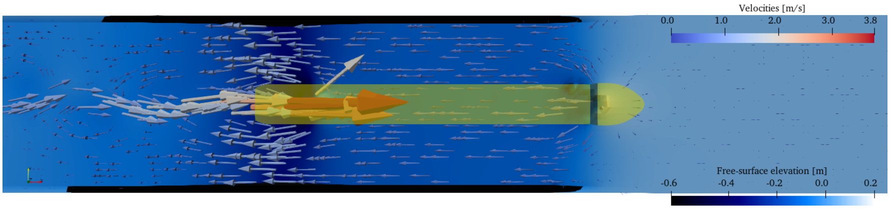


Figure 3.37. Velocity field at stationary conditions. Test case C-V1 with skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s, $\gamma_{sf} = 45$ (constant).

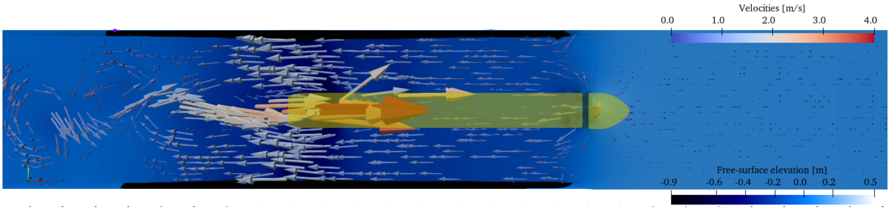


Figure 3.38. Velocity field at stationary conditions. Test case C-V3 with skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s, $\gamma_{sf} = 180$ (constant).

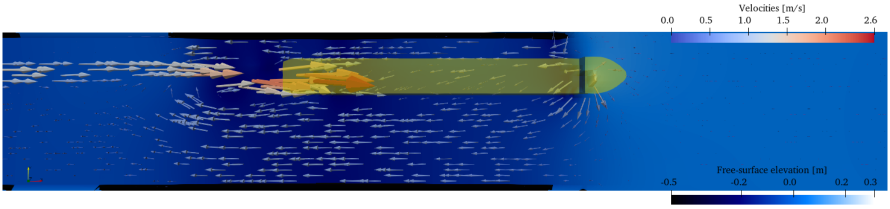


Figure 3.39. Velocity field at stationary conditions. Test case OBT-V1 with skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s, $\gamma_{sf} = 65$ (constant).

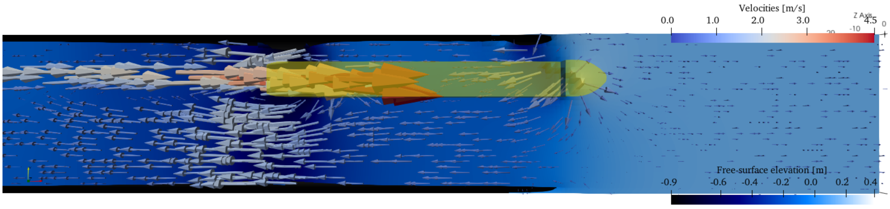


Figure 3.40. Velocity field at stationary conditions. Test case OBT-V3 with skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s, $\gamma_{sf} = 120$ (constant).

Case	Measured squat [m]	Computed squat [m]	Measured return-flow velocity [m/s]	Computed return-flow velocity [m/s]
C-V1	0.28	0.15	1.00	(0.60-1.25)
C-V3	0.48	0.21	1.90	(0.70-1.40)
OBT-V1	0.17	0.15	1.30	0.65
OBT-V3	0.49	0.20	1.60	(0.60-2.00)

Table 3.5. Comparison of the measured and computed squat and return-flow velocity. Test cases with skin friction. Measured data from Doychev et al. [48].

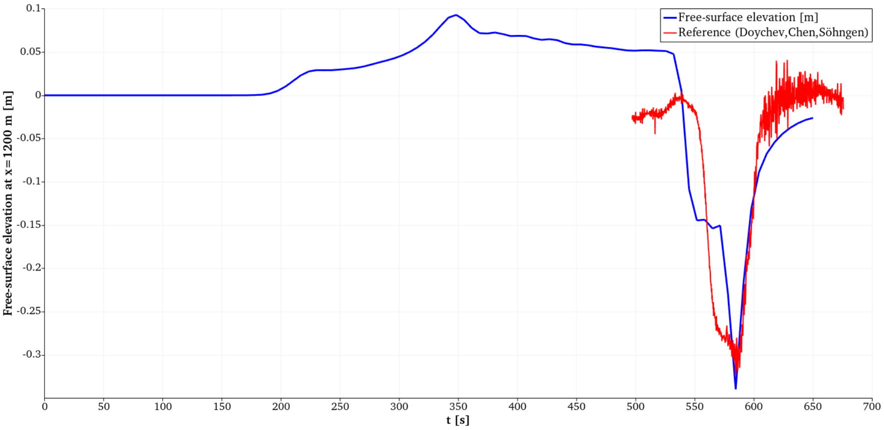


Figure 3.41. Free-surface elevation compared to the value measured at the gauge. Test case C-V1 with skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s, $\gamma_{sf} = 45$ (constant).

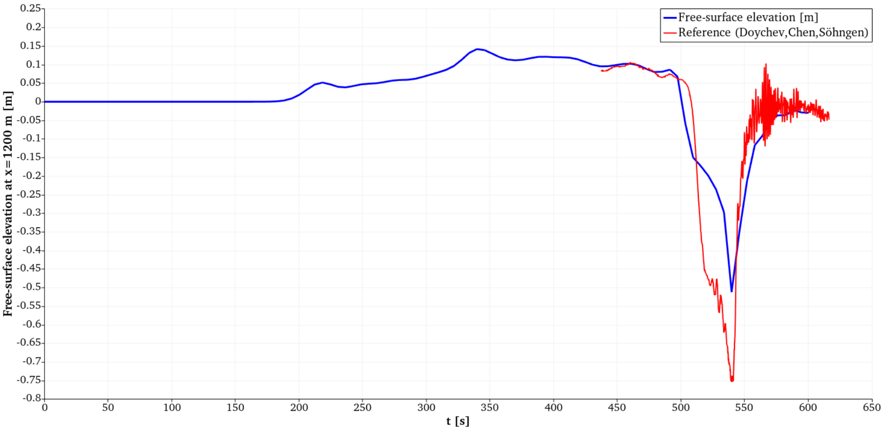


Figure 3.42. Free-surface elevation compared to the value measured at the gauge. Test case C-V3 with skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s, $\gamma_{sf} = 180$ (constant).

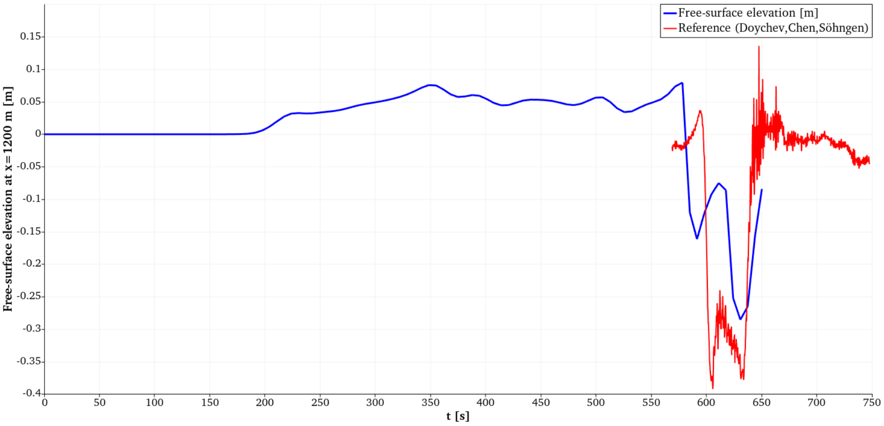


Figure 3.43. Free-surface elevation compared to the value measured at the gauge. Test case OBT-V1 with skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s, $\gamma_{sf} = 65$ (constant).

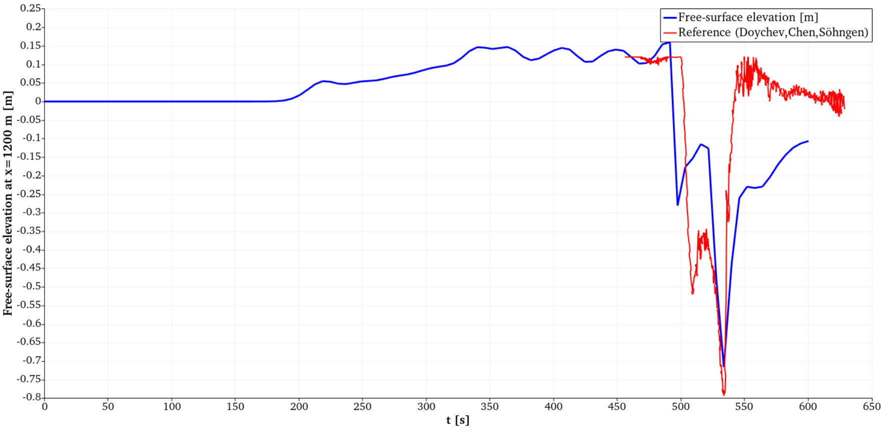


Figure 3.44. Free-surface elevation compared to the value measured at the gauge. Test case OBT-V3 with skin friction. Setup: $\Delta x = 1.6$ m, $\Delta y = 2$ m, subgrid 4x4, $\Delta t_{max} = 0.5$ s, $\gamma_{sf} = 120$ (constant).

3.5 Validation of the reaction forces

In this Section, we compare the results of our model to measurements done in a laboratory in a resistance test. The tests were conducted at the Development Centre for Ship Technology and Transport *DST* [67].
In the experiment, a model ship is towed in a open channel at a constant speed of 0.556 m/s and fixed elevation. The researchers modified the depth-to-draft ratio and the drift angle and measured the forces and torques acting on the bare hull.
In the numerical implementation, the boundary conditions are non-reflecting Dirichlet on all four sides. The hull shape is defined by the Referenzschiff STL. The body of water is initially at rest. The bottom bathymetry is an horizontal plane and its elevation is defined by the depth-to-draft ratio. At the initial time, the ship's draft is 0.2188 m, it is only allowed to move horizontally and its speed is constant throughout the simulation. The smooth-movement algorithm of Section 2.13.3 is used in all the simulations.

3.5.1 Drift without skin friction

In this Section, we show the results of the towed-ship test case where the skin friction is not applied.
In Table 3.6, the horizontal reaction forces are compared with the experimental values. Both forces are expressed in the ship frame of reference. Satisfactory results are obtained for the sway force F_y, which gets very close to the reference data. As expected, for a null drift angle the force is also null. The model tends to slightly underestimate the sway force for higher drift angles. As assessed in the experiments, the force lowers when the height-to-draft ratio increases. Conversely, the surge-force prediction is unreliable, a variation of the drift angle has no sensible effect on the results.
In Table 3.7, the yaw torque comparison is made. The torque predicted by the model is plausible but strongly underestimated, especially at low depth-to-draft ratios.

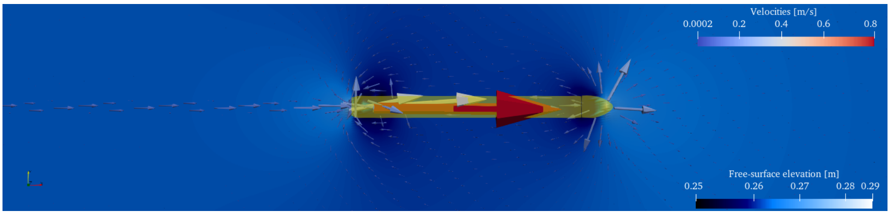


Figure 3.45. Velocity field at stationary conditions. Test case of $\alpha = 0^o$, $H/T = 1.2$ without skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s.

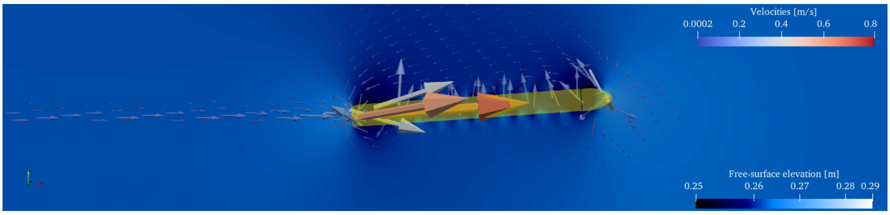


Figure 3.46. Velocity field at stationary conditions. Test case of $\alpha = 4^o$, $H/T = 1.2$ without skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s.

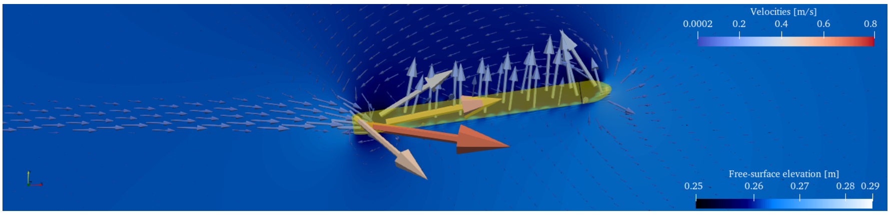


Figure 3.47. Velocity field at stationary conditions. Test case of $\alpha = 8^o$, $H/T = 1.2$ without skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s.

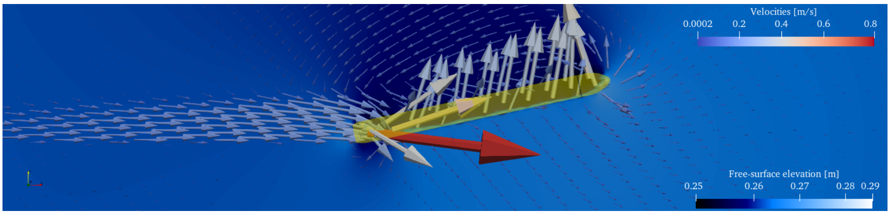


Figure 3.48. Velocity field at stationary conditions. Test case of $\alpha = 12^o$, $H/T = 1.2$ without skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s.

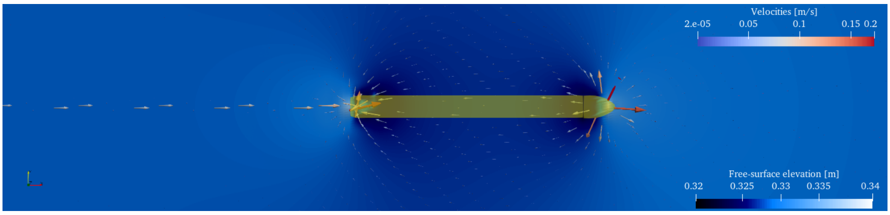


Figure 3.49. Velocity field at stationary conditions. Test case of $\alpha = 0^o$, $H/T = 1.5$ without skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s.

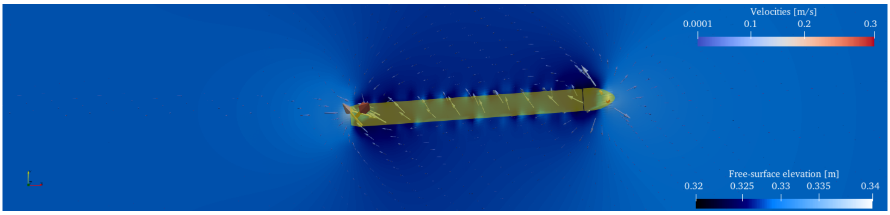


Figure 3.50. Velocity field at stationary conditions. Test case of $\alpha = 4^o$, $H/T = 1.5$ without skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s.

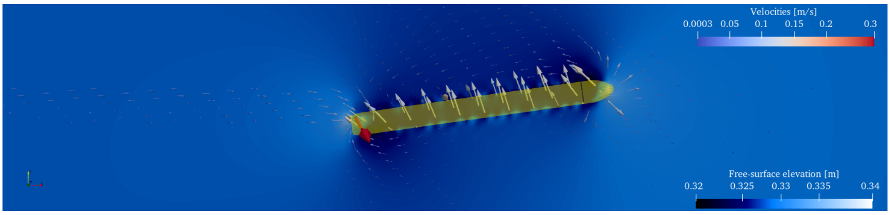


Figure 3.51. Velocity field at stationary conditions. Test case of $\alpha = 8^o$, $H/T = 1.5$ without skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s.

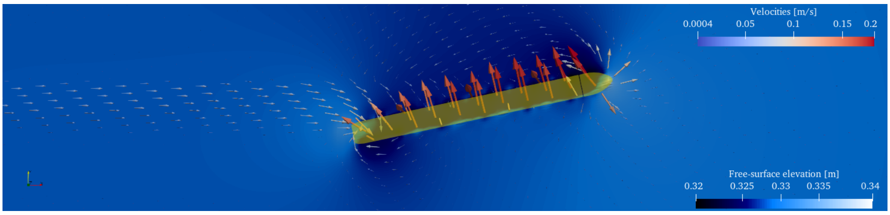


Figure 3.52. Velocity field at stationary conditions. Test case of $\alpha = 12^o$, $H/T = 1.5$ without skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s.

H/T	Drift angle [°]	Measured F_x [N]	Computed F_x [N]	Measured F_y [N]	Computed F_y [N]
1.2	0	-2.61	-20	-0.98	1
1.2	4	-6.36	-22	51.97	55
1.2	8	-4.78	-21	128.78	128
1.2	12	4.14	-20	230.86	210
1.5	0	-5.08	-16	2.43	-0.3
1.5	4	-6.79	-16	16.87	15.5
1.5	8	-8.44	-16	50.02	40
1.5	12	-8.62	-15	86.33	70

Table 3.6. Comparison of the measured and computed horizontal forces. Test cases without skin friction. Measured data from the DST research institute.

H/T	Drift angle [°]	Measured τ_z [Nm]	Computed τ_z [Nm]
1.2	0	-1.75	-5
1.2	4	94.93	5
1.2	8	229.86	55
1.2	12	510.08	103
1.5	0	4.91	-1
1.5	4	36.79	29
1.5	8	84.46	62
1.5	12	129.58	95

Table 3.7. Comparison of the measured and computed yaw torque. Test cases without skin friction. Measured data from the DST research institute.

3.5.2 Drift with skin friction

In this Section, we show the results of the towed-ship test case where the skin friction is applied. The calibration of the γ_{sf} factor is done on F_y in the case of depth-to-draft ratio of 1.2 and drift angle of 12^o; an optimal factor $\gamma_{sf} = 35$ is determined.

In Table 3.8, the horizontal reaction forces are compared with the experimental values. The considerations on the general behavior are similar to the ones done for the frictionless case of Section 3.5.1. The sway force is predicted well overall. In the specific case used for the calibration, there is a perfect match by design; in the other cases, the skin friction application has lead to a slight overestimation of the sway force. Quite surprisingly, the surge force F_x is mostly unaffected by the skin friction and remains a poor prediction of the measured force.

In Table 3.9, the yaw torque τ_z comparison is made. The torque predicted by the model is still strongly underestimated. The effect of the skin friction is opposite depending on the depth-to-draft ratio. For the lower H/T ratio the torque has increased compared to the frictionless case, while for the higher ratio it has decreased.

H/T	Drift angle [°]	Measured F_x [N]	Computed F_x [N]	Measured F_y [N]	Computed F_y [N]
1.2	0	-2.61	-20	-0.98	1.5
1.2	4	-6.36	-22	51.97	70
1.2	8	-4.78	-21	128.78	147
1.2	12	4.14	-21	230.86	231
1.5	0	-5.08	-18	2.43	0
1.5	4	-6.79	-18	16.87	26
1.5	8	-8.44	-15	50.02	53
1.5	12	-8.62	-15	86.33	83

Table 3.8. Comparison of the measured and computed horizontal forces. Test cases with skin friction. Measured data from the DST research institute.

Figures from 3.53 to 3.56 show the velocity field and elevation profile in the case of H/T= 1.2 for different drift angles, while Figures from 3.57 to 3.60 refer to the H/T= 1.5 case. The decrease of flow speed for higher depth-to-draft ratios is quite evident, as is the recirculation caused by the drift angle and proportional to it. The effect of the skin friction is mostly indicated by a general intensification of the velocities magnitude.

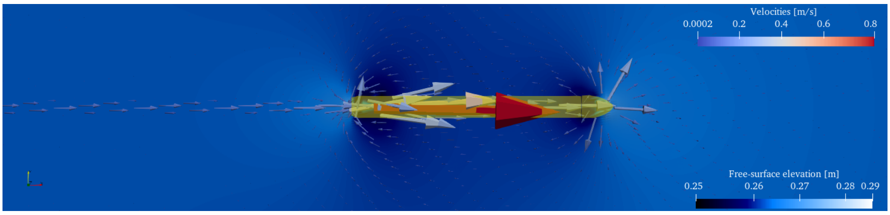


Figure 3.53. Velocity field at stationary conditions. Test case of $\alpha = 0^o$, $H/T = 1.2$ with skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s, $\gamma_{sf} = 35$ (H/T calibrated).

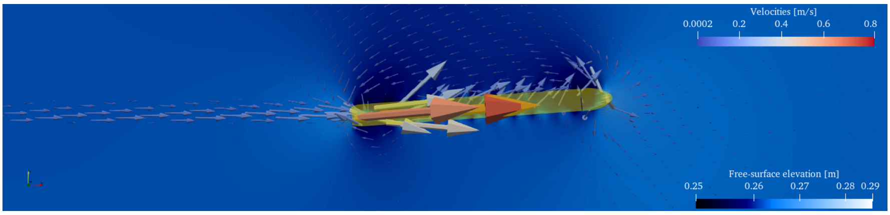


Figure 3.54. Velocity field at stationary conditions. Test case of $\alpha = 4^o$, $H/T = 1.2$ with skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s, $\gamma_{sf} = 35$ (H/T calibrated).

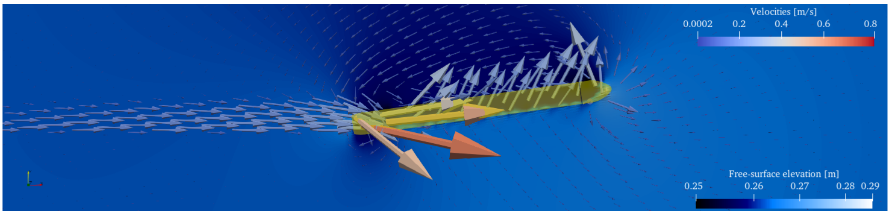


Figure 3.55. Velocity field at stationary conditions. Test case of $\alpha = 8^o$, $H/T = 1.2$ with skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s, $\gamma_{sf} = 35$ (H/T calibrated).

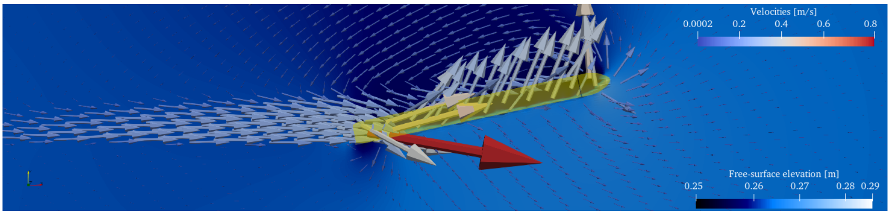


Figure 3.56. Velocity field at stationary conditions. Test case of $\alpha = 12^o$, $H/T = 1.2$ with skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s, $\gamma_{sf} = 35$ (H/T calibrated).

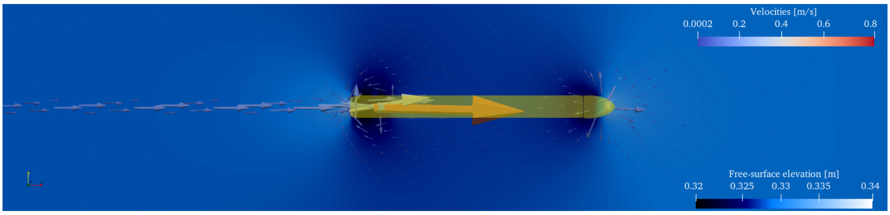


Figure 3.57. Velocity field at stationary conditions. Test case of $\alpha = 0^o$, $H/T = 1.5$ with skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s, $\gamma_{sf} = 35$ (H/T calibrated).

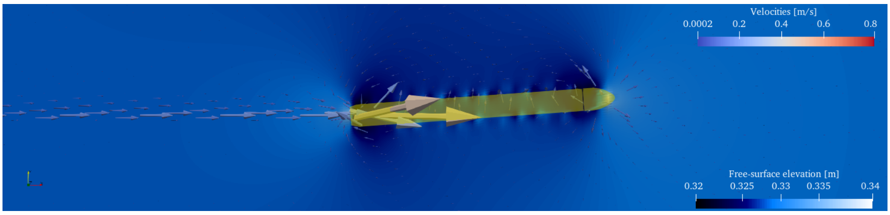


Figure 3.58. Velocity field at stationary conditions. Test case of $\alpha = 4^o$, $H/T = 1.5$ with skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s, $\gamma_{sf} = 35$ (H/T calibrated).

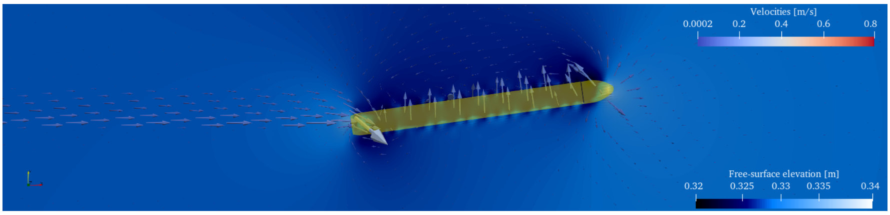


Figure 3.59. Velocity field at stationary conditions. Test case of $\alpha = 8^o$, $H/T = 1.5$ with skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s, $\gamma_{sf} = 35$ (H/T calibrated).

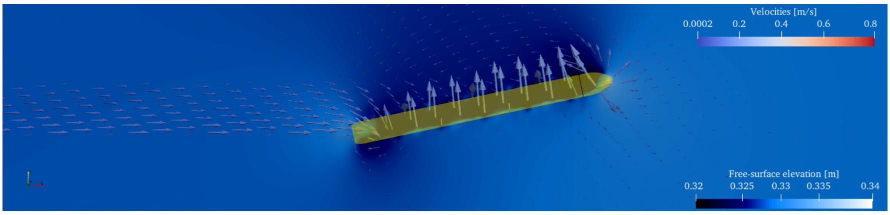


Figure 3.60. Velocity field at stationary conditions. Test case of $\alpha = 12^o$, $H/T = 1.5$ with skin friction. Setup: $\Delta x = 0.2$ m, $\Delta y = 0.08$ m, subgrid 4x4, $\Delta t_{max} = 0.05$ s, $\gamma_{sf} = 35$ (H/T calibrated).

H/T	Drift angle [°]	Measured τ_z [Nm]	Computed τ_z [Nm]
1.2	0	-1.75	-2
1.2	4	94.93	21
1.2	8	229.86	70
1.2	12	510.08	110
1.5	0	4.91	-0.5
1.5	4	36.79	11
1.5	8	84.46	48
1.5	12	129.58	82

Table 3.9. Comparison of the measured and computed yaw torque. Test cases with skin friction. Measured data from the DST research institute.

4 Conclusions and outlook

In this thesis a novel staggered semi-implicit finite volume scheme for the numerical solution of fluid-structure-interaction problems is presented. In particular, the interaction of hydrostatic geophysical free surface flows with floating ships is studied. The ship is considered as a moving rigid body with 6 degrees of freedom and its dynamics is described at the aid of a first order system of ordinary differential equations. The PDE and ODE systems communicate with each other through the nonlinear volume function in the discrete hydrodynamics equations and the pressure field which provides the forces for the rigid body dynamics.

The approach presented in this thesis is based on - but not limited to - the hydrostatic pressure assumption and is therefore computationally very efficient compared to a fully non-hydrostatic 3D CFD simulation of the flow around floating ships and with a moving free surface. The mathematical model for the free surface flows based on the depth function including the wetting and drying process as integral part of the algorithm makes it possible to study natural bodies of water ranging from rivers and lakes to the open sea.

Mass and linear momentum conservation is ensured globally and locally at all time steps.

The main contributions of this thesis are the coupling of the pressurized model of Casulli and Stelling [41] with the ODE system that describes the motion of a ship, the verification of the vertical motion, the development of tools for the stabilization of the horizontal motion and the validation of the model in terms of ship-generated waves, squat, return flow and reaction forces.

4.1 Grid, subgrid and domain decomposition

The model is designed for applications at the geophysical mesoscale typical for inland waterways where objects like a ship represent a small portion of the domain. In such setting, capturing the physical phenomenae in the ship surroundings requires a fine grid, while for the river flow far from the vessel a coarser grid could suffice. In simulations with different spatial scales and where considerable parts of the domain are always dry, a good choice could be an unstructured grid. However, if the ship moves horizontally the grid would need do be redefined to follow the floating object. Remeshing can be computationally expensive and an appropriate algorithm covering the domain between the river banks to the ship waterline is not trivial to implement. Also, even if the object moves only vertically, the spatial accuracy is always limited to first order due to the transition from fine to coarse cells. For these reasons, a Cartesian grid was a preferred choice compared to an unstructured grid. With a Cartesian grid, the locally-needed fine grid would extend everywhere and the number of computational cells would become large, causing an excessive computational effort. To overcome this disadvantage of the Cartesian grid, the efficiency and code speed have been improved using subgrid and high performance computing (HPC) techniques based on MPI parallelization.

The subgrid incorporates the bottom bathymetry and the geometry of the ship, allowing to maintain a coarse grid and, at the same time, to compute the water volumes with higher precision. Detailed input does not determine the grid resolution but still has a direct influence on the results through the subgrid, which retains the information provided by the Digital Elevation Models (DEM) of the terrain and the fine tessellations that define the ship hull. The subgrid modifies the depth function, especially where its first derivative has jump discontinuities. Intermediate states of cell pressurization and filling with water (dry/wet state) are explicitly taken into account, making the transition between states smoother. Applying the subgrid requires some additional computational resources which are, however, usually noticeably smaller than those required for an equivalently-fine full grid. With subgrids the gap between different scales of the river and the ship is bridged, resulting in a balance of accuracy, speed, and flexibility.

Even with the subgrid, the number of computational cells could be high for a single processor core. The parallelization of the code with the integration of the message passing interface (MPI) allows to run the simulations on massively parallel distributed memory supercomputers. The parallel execution of the program fits well with the requirements of this model and the intended applications. It is still possible to improve the implementation, especially for the cases with a high number of ever-dry cells. In such situations, an introduction of connectivity tables similar to the approach in unstructured grids would be required.

4.2 Dispersion effects

The range of applications of the model is extended with a non-hydrostatic update of the hydrostatic pressure. Dispersive effects are introduced by a higher-order Boussinesq-type term in the momentum equation.

The nonhydrostatic algorithm has also been tested in the validation case of a ship towed in a confined channel. In this case, ambiguous results have been obtained depending on the grid resolution: for a coarser resolution, the solution is similar to the hydrostatic one, with a slightly improved drawdown prediction, while for a finer resolution the vortices originating at the stern get stronger and make the wave profile more oscillatory than the reference solution. At this stage of the analysis, it is not clear if this last result is acceptable; in the reference case the ship had a propeller which could have prevented the vortices formation.

4.3 Horizontal motion

The numerical instability generated during the horizontal motion of the ship was one of the main challenges that were faced during the development of the SIFSI model. The need to reduce the pressure oscillations gave the opportunity to analyze the role of the $\partial\left[H(\eta)\right]/\partial t$ derivative term from different perspectives and led to the development of multiple tools.

The smooth-movement algorithm (Sections 2.13.3 and 3.3.1), and the subgrid as well (Section 2.2), highlighted the importance of transitioning cells smoothly to the pressurized state. Compared to how the displaced volumes were previously computed, the smooth-movement algorithm is superior in terms of stability because it considers the fluid-structure interaction in each time iteration.

The roof-relaxation method (Sections 2.13.4 and 3.3.2) proves useful in the suppression of the oscillations but it also affects the wave generation and the ship-floating condition negatively, which may cause erroneous results. This risk was limited by applying the relaxation only near the waterline and for a fixed amount of cells; in this way, the relaxations acts only as a transition buffer and its influence reduces with grid refinements.

The smoothing of the input geometry (Sections 2.13.5 and 3.3.3) spreads the displaced volume among multiple cells and thus increases the transition time to the pressurized state for a cell in consideration. With forcing terms divided among a larger number of computational cells, the solution system converges to a smoother pressure distribution, and the oscillations are dampened. If the forces are computed directly on the modified shape, its draft at equilibrium changes, altering also the ship-induced waves. To reduce the impact of this problem, the smoothing is applied locally, limiting its action to an area close to the vertical-walls, and causing this area to shrink with grid refinements. As an alternative approach, one could compute the forces on the input shape and use the smoother shape only when computing the water depth.

The cell-merging technique (Sections 2.13.6 and 3.3.4) leverages the understanding of what causes the oscillations to avoid them. In the example shown in this thesis the cell-merging technique was successful, the oscillations were reduced and the wave profile was similar to the reference profile. However, this technique relies on the prediction of the destination cell. The choice of displacing the volumes in the closest free-surface cell seems accurate enough for objects that travel in a straight line; its behavior should be verified for more generic ship maneuvers.

The introduction of a Rusanov-type dissipation in the scheme (Sections 2.13.7 and 3.3.5) showed the greatest potential for the control of the oscillations and the pressure function smoothness in general. With no limitation on the dissipation-term intensity, the smoothing effect can be so strong that the pressure becomes the plane that best fits the waterline points. This impacts heavily the waves generation and the reaction forces on the hull. The use of the Rusanov-type dissipation should be calibrated to the ship characteristics and speed in order to reach the optimum of accuracy and stability.

Among the suggested solutions, the smooth movement algorithm is the only one which seems to be the most advantageous. Other solutions are effective in the smoothing of the oscillations, but they require ad-hoc adjustments and/or they affect the solution negatively. Since all these methods can work concurrently, their use with the smooth-movement algorithm or with a sufficiently fine subgrid can be considered, for example if the void below the hull persists and grid refinement results in unacceptable computational resource usage.

4.4 Verification and validation

An important part of the model development was the verification done in Section 3.2 against theoretical solutions known for simplified cases. When there are no floating objects, the model provides the results that would be expected from a usual shallow water model. It can be shown that when we include floating objects, their response is consistent with the solution of the reference problems for the case of the vertical degree of freedom. Additionally, the damping and the added mass forces are naturally included, without the need to model them. An approximate knowledge of the added mass is needed in case the added mass exceeds the actual mass of the ship, which would make the interaction unconditionally unstable [79]. The parameters of the waves generated by the fluid-structure interaction, tested in the case of the heaving ship motion, are coherent with the analytical value for this case.

The verification is an important step because it shows how the model behaves and it helps in the correction of the implementation. It reassures that the numerical prediction is in line with the theory on which the model is based. However, every theory has assumptions, limitations and a range of validity, so the verification per se does not reassure that its predictions can be trusted for real-world applications.

Actually, in order to assess limitations of the model, a comparison with experimental data was carried out in Sections 3.4 and 3.5. The measurements, provided by BAW, involved a full-scale ship moving at different speeds and positions relative to the waterway, and a model ship moving at constant speed but at variable drift angles. In the numerical simulation, the precise geometry of the vessel (read directly from a STL file) was directly applied for the computation of the reaction forces, while for the displaced volumes the input shape has been simplified to a series of piecewise-constant elements. The use of the detailed ship geometry from the STL file, which obviously has elements on the vertical sides, significantly improves the accuracy of the horizontal forces computation.

For the test cases concerning ship-generated waves the results delivered a correct prediction of the duration of the wave passage. The wave profile responds in a plausible way to the variation of the vessel speed and it is sensitive to the position of the ship within a confined waterway. The amplitude of the bow and stern wave is underestimated in general. However, a calibration of the drag forces yields noticeably better results.

The squat prediction was evaluated by letting the ship move horizontally and allowing the free adjustment of the ship position in the vertical direction according to balance of the weight force and the vertical forces acting on the hull. Once the acceleration phase ends, the primary-wave system around the ship stabilizes, and the sinkage can be evaluated. The squat, influenced also by the smaller-than-expected wave amplitude, is underestimated but it varies in the correct direction depending on the vessel speed. It must be noticed that more accurate results can be obtained in the case without the application of skin friction.

The lateral return flow follows the same underestimation trend of the ship-generated waves and squat variables. For this variable, however, it is not specified if the measured value is an instant velocity or if it is a time-average, which could be relevant because the flow speed is not constant in the lateral region. When the drag forces are included, the wave profile changes, and since we are at steady conditions there are local flow accelerations. If the spatial variability of the velocity is considered, the experimental value falls between the minimum and maximum numerical values.

The validation of the reaction forces was done in the presence of sub-critical flows for the surge and sway forces, and for the yaw torque. The total resistance registered by the model is due mainly to the wave profile and the water flow around the ship, which provide the boundary conditions for the pressure function in the ship region. In our calculations the pressure resistance is the main component of the total resistance, while the friction resistance is a smaller component. Strangely, the skin friction seems to affect only the sway force. The sway force is predicted well by the numerical model for different drift angles and depth-to-draft conditions. Contrarily, the surge force is not sensitive to the different configurations of the problem. The yaw torque responds correctly to the variation of the drift angle and the depth-to-draft ratio, but it is strongly underestimated.

Considering that the model described in this thesis is designed for applications in the mesoscale, the comparison of experimental and numerical primary-wave system is satisfactory, even if generally underestimated. The reaction forces computed from the model do not seem to be accurate enough in order to be trusted in applications concerning ship maneuvers. One possible explanation is the absence of turbulence, which may be important considering the generation of eddies in the drift test case. Another explanation could be the insufficient number of elements in the input geometry of the ship. Indeed, since the sides of the ship have low curvature there are not many elements that describe them, which may impact the force estimation for a large portion of the hull. It is also possible that a very fine grid is needed; in this case, to avoid excessive computational costs it may be worthwhile to consider alternative methods, for example those that base the forces prediction only on some parameters as the ship speed and the drift angle. Additionally, in order to assess the possible influences of the horizontal motion instabilities and the algorithms applied to diminish them, the test cases should be repeated in the galilean-invariance setup. Consequently, there is a clear need for further research concerning the modelling of horizontal reaction forces.

4.5 Computational efficiency

Securing the computational efficiency has always been one of the guiding principles of the SIFSI model development. This requirement is a necessity dictated by the different scales that are coupled: the dimension of the domain is determined by the river-stretch extension and by the type of ship maneuvers, while the dimension of the cell depends on the ship geometry. These two scales usually differ by order of magnitudes, which makes 2D$_{xy}$ or 3D simulations computationally demanding. For this reason, efficiency characterizes many of the aspects of the model: the three-dimensional free-surface model of Casulli and Cheng [38] made the implementation very flexible, because the same program can be used for 3D, 2D$_{xy}$, 2D$_{xz}$ and 1D simulations, simply by controlling the number of computational cells in each direction. Also, even for 3D simulations the size of the solution system is independent of the number of vertical layers, which makes their use cost-effective. Each tridiagonal linear system of the vertical components can be solved by a direct method, such as the Thomas algorithm.

The semi-implicit discretization allowed to remove the most influential terms from the CFL condition, the wave celerity and the vertical component of the viscosity, increasing the admissible timestep. Hovewer, using the semi-implicit method adds the cost of solving an (implicit) equation system, which fortunately is a mildly-nonlinear system that can be solved by a nested-Newton-type algorithm. During each iteration, the system is linearized and preconditioned, allowing to use a matrix-free conjugate gradient method.

The introduction of a subgrid allowed to maintain a reasonable accuracy for coarser grids. Since a consistent part of the time is spent to find the solution of the nonlinear system, the reduction in the number of computational cells speeds up the convergence of the conjugate-gradient algorithm. Also, a coarser grid is subject to a milder CFL condition, increasing the allowed timestep.

The choice of using Fortran for the implementation was based on its speed and on its compatibility with the MPI standard, which made it possible to use the domain-decomposition method to split the computational load among many cores and processors.

The Boussinesq-type solution, being based on a depth-averaged model, is not excessively demanding in terms of computational time.

4.6 Outlook on future research

In the following, a brief outlook on the recommended future work and improvements is provided. Starting from the mathematical model and the numerical scheme developed and implemented in this thesis, it is possible to progress in many ways:

- Obtain a more realistic ship dynamics model, considering propulsion and steering.

- Investigate in more depth the generation of ship-bank effects, ship induced waves interacting with banks, the squat of a ship in motion, ship-ship interactions [118] and the interactions affecting a ship within a lock on a waterway.

- Validate the model in the case of river flows, using simulations with input data (bathymetry, friction coefficients, shape and mass distribution of the ship, boundary conditions) from real test cases. The ship geometry could be read, for example, in the form of a surface triangulation (STL file) and the river bathymetry defined from available Digital Elevation Models (DEM) at the subgrid level of varying accuracy.

- Improve the dispersion relation by the use of the Madsen-Sørensen model, instead of Peregrine-Abbott model. Such model includes a higher spatial derivative of the pressure η_{xxx}, which would need to be taken as an implicit term. The coefficients of this model are calibrated to obtain the best fit with Stokes first-order theory.
- Update the hydrostatic pressure with a fully non-hydrostatic pressure correction approach according to the seminal work of Casulli [31], which is in principle similar to the SIMPLE method of Patankar and Spalding [91], but with a particularly sophisticated initial guess for the pressure based on the efficient solution of the underlying hydrostatic problem. It would be very interesting to compare it with the Boussinesq-type solution in terms of computational costs and accuracy, and determine the operating conditions that produce similar results.
- Increase the space-time order of accuracy of the method, for example using IMEX schemes [16] as well as higher order discontinuous Galerkin finite element methods [111].
- Removing the ever-dry mesh cells from the computation.
- Improve the reaction forces, which become important for the dimensioning of waterways for ship maneuvers in the presence of external currents. The introduction of turbulence with a k-ϵ or k-ω model could be a first promising attempt, even though they are computationally intensive. The chosen solution must work for relatively coarse grid resolutions. Alternatively, the computation of the Riemann sum on the hull could be avoided and the maneuvers could be controlled by another model; the current standard is the Abkowitz-type model [2, 121, 122].

List of Figures

List of Tables

References

[1] M. Abbott, H. Petersen, and O. Skovgaard. Computations of short waves in shallow water. *Coastal Engineering Proceedings*, 1, 23, 1978.

[2] M. Abkowitz. Measurement of Hydrodynamic Characteristics from Ship Manoeuvring Trials by System Identification. *SNAME Transactions*, 88, 283–318, 1980.

[3] J. Ahrens, B. Geveci, and C. Law. ParaView: An End-User Tool for Large-Data Visualization. In: *Visualization Handbook.* Ed. by C. D. Hansen and C. R. Johnson. Burlington: Butterworth-Heinemann, 2005. 717–731.

[4] U. Ayachit. *The ParaView Guide: A Parallel Visualization Application.* Clifton Park, NY, USA: Kitware, Inc., 2015.

[5] Y. Bai and K. F. Cheung. Dispersion and kinematics of multi-layer non-hydrostatic models. *Ocean Modelling*, 92, 11–27, 2015.

[6] J. Banks, W. Henshaw, D. Schwendeman, and Q. Tang. A stable partitioned FSI algorithm for rigid bodies and incompressible flow. Part I: Model problem analysis. *Journal of Computational Physics*, 343, 432–468, 2017.

[7] J. Banks, W. Henshaw, D. Schwendeman, and Q. Tang. A stable partitioned FSI algorithm for rigid bodies and incompressible flow. Part II: General formulation. *Journal of Computational Physics*, 343, 469–500, 2017.

[8] J. Banks, W. Henshaw, D. Schwendeman, and Q. Tang. A stable partitioned FSI algorithm for rigid bodies and incompressible flow in three dimensions. *Journal of Computational Physics*, 373, 455–492, 2018.

[9] C. Bassi, L. Bonaventura, S. Busto, and M. Dumbser. A hyperbolic reformulation of the Serre-Green-Naghdi model for general bottom topographies. *Computers & Fluids*, 212, 104716, 2020.

[10] G. Beck and D. Lannes. Freely Floating Objects on a Fluid Governed by the Boussinesq Equations. *Annales de l'Institut Henri Poincaré C, Analyse non linéaire*, `https://doi.org/10.4171/aihpc/15`, 2022.

[11] A. Bermúdez and M. E. Vázquez-Cendón. Upwind Methods for Hyperbolic Conservation Laws with Source Terms. *Comput. Fluids*, 23, 1049–1071, 1994.

[12] V. Bertram. *Practical Ship Hydrodynamics.* Elsevier Science, 2011.

[13] H. Bingham. A hybrid Boussinesq-panel method for predicting the motion of a moored ship. *Coastal Engineering*, 40, `https://doi.org/10.1016/S0378-3839(00)00002-8`, 21–38, 2000.

[14] E. Blayo and L. Debreu. Revisiting open boundary conditions from the point of view of characteristic variables. *Ocean Modelling*, 9, 231–252, 2005.

[15] E. Bocchi, J. He, and G. Vergara Hermosilla. Modelling and simulation of a wave energy converter. *ESAIM: Proceedings and Surveys*, 70, `https://doi.org/10.1051/proc/202107005`, 68–83, 2021.

[16] W. Boscheri and L. Pareschi. High order pressure-based semi-implicit IMEX schemes for the 3D Navier-Stokes equations at all Mach numbers. *Journal of Computational Physics*, 434, `https://doi.org/10.1016/j.jcp.2021.110206`, 110206, 2021.

[17] U. Bosi, A. P. Engsig-Karup, C. Eskilsson, and M. Ricchiuto. A spectral/hp element depth-integrated model for nonlinear wave–body interaction. *Computer Methods in Applied Mechanics and Engineering*, 348, `https://doi.org/10.1016/j.cma.2019.01.020`, 222–249, 2019.

[18] S. Bousso, M. Daynou, and M. Fuamba. Numerical Modeling of Mixed Flows in Storm Water Systems: Critical Review of Literature. *Journal of Hydraulic Engineering*, 139, `https://doi.org/10.1061/(ASCE)HY.1943-7900.0000680`, 385–396, 2013.

[19] S. F. Bradford. Nonhydrostatic model for free surface flow interaction with structures. *International Journal for Numerical Methods in Fluids*, 93, `https://doi.org/10.1002/fld.4985`, 2508–2530, 2021.

[20] C. Brennen. *A Review of Added Mass and Fluid Inertial Forces.* Tech. rep. Port Hueneme, CA, USA: Department of the Navy, Jan. 1982.

[21] L. Brugnano and V. Casulli. Iterative solution of piecewise linear systems. *SIAM Journal on Scientific Computing*, 30, 463–472, 2007.

[22] L. Brugnano and V. Casulli. Iterative solution of piecewise linear systems and applications to flows in porous media. *SIAM Journal on Scientific Computing*, 31, 1858–1873, 2009.

[23] C. Brutto and M. Dumbser. A semi-implicit finite volume scheme for a simplified hydrostatic model for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 95, 107–142, 2023.

[24] S. Busto and M. Dumbser. A staggered semi-implicit hybrid finite volume/finite element scheme for the shallow water equations at all Froude numbers. *Applied Numerical Mathematics*, 175, 108–132, 2022.

[25] S. Busto, M. Dumbser, C. Escalante, S. Gavrilyuk, and N. Favrie. On high order ADER discontinuous Galerkin schemes for first order hyperbolic reformulations of nonlinear dispersive systems. *Journal of Scientific Computing*, 87, 48, 2021.

[26] S. Busto, J. L. Ferrín, E. F. Toro, and M. E. Vázquez-Cendón. A projection hybrid high order finite volume/finite element method for incompressible turbulent flows. *Journal of Computational Physics*, 353, 169–192, 2018.

[27] R. Caiden, R. P. Fedkiw, and C. Anderson. A Numerical Method for Two-Phase Flow Consisting of Separate Compressible and Incompressible Regions. *Journal of Computational Physics*, 166, 1–27, 2001.

[28] R. A. Carmigniani and D. Violeau. Optimal sponge layer for water waves numerical models. *Ocean Engineering*, 163, 169–182, 2018.

[29] M. Castro, J. Gallardo, and C. Parés. High order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. Applications to shallow-water systems. *Math. Comput.*, 75, https://doi.org/10.1090/S0025-5718-06-01851-5, 1103–1134, 2006.

[30] O. Castro-Orgaz and H. Chanson. Ritter's dry-bed dam-break flows: positive and negative wave dynamics. *Environmental Fluid Mechanics*, 17, https://doi.org/10.1007/s10652-017-9512-5, 2017.

[31] V. Casulli. A semi-implicit finite difference method for non-hydrostatic, free-surface flows. *International Journal for Numerical Methods in Fluids*, 30, https://doi.org/10.1002/(SICI)1097-0363(19990630)30:4<425::AID-FLD847>3.0.CO;2-D, 425–440, 1999.

[32] V. Casulli and E. Cattani. Stability, accuracy and efficiency of a semi-implicit method for three-dimensional shallow water flow. *Computers & Mathematics with Applications*, 27, https://doi.org/10.1016/0898-1221(94)90059-0, 99 –112, 1994.

[33] V. Casulli, M. Dumbser, and E. Toro. Semi–implicit numerical modelling of axially symmetric flows in compliant arterial systems. *International Journal for Numerical Methods in Biomedical Engineering*, 28, 257–272, 2012.

[34] V. Casulli and R. A. Walters. An unstructured grid, three–dimensional model based on the shallow water equations. *International Journal for Numerical Methods in Fluids*, 32, 331–348, 2000.

[35] V. Casulli and P. Zanolli. A Nested Newton-Type Algorithm for Finite Volume Methods Solving Richards' Equation in Mixed Form. *SIAM Journal on Scientific Computing*, 32, `https://doi.org/10.1137/100786320`, 2255–2273, 2010.

[36] V. Casulli and P. Zanolli. Iterative solutions of mildly nonlinear systems. *Journal of Computational and Applied Mathematics*, 236, `https://doi.org/10.1016/j.cam.2012.02.042`, 3937 –3947, 2012.

[37] V. Casulli. A high-resolution wetting and drying algorithm for free-surface hydrodynamics. *International Journal for Numerical Methods in Fluids*, 60, `https://doi.org/10.1002/fld.1896`, 391–408, 2009.

[38] V. Casulli and R. T. Cheng. Semi-implicit finite difference methods for three-dimensional shallow water flow. *International Journal for Numerical Methods in Fluids*, 15, 629–648, 1992.

[39] V. Casulli and G. S. Stelling. Numerical Simulation of 3D Quasi-Hydrostatic, Free-Surface Flows. *Journal of Hydraulic Engineering*, 124, `https://doi.org/10.1061/(ASCE)0733-9429(1998)124:7(678)`, 678–686, 1998.

[40] V. Casulli and G. S. Stelling. Semi-implicit subgrid modelling of three-dimensional free-surface flows. *International Journal for Numerical Methods in Fluids*, 67, `https://doi.org/10.1002/fld.2361`, 441–449, 2011.

[41] V. Casulli and G. S. Stelling. A semi-implicit numerical model for urban drainage systems. *International Journal for Numerical Methods in Fluids*, 73, `https://doi.org/10.1002/fld.3817`, 600–614, 2013.

[42] X.-N. Chen. Study of one-dimensional ship squat theories. *Chinese Journal of Hydrodynamics*, 28, 399, 2013.

[43] X.-N. Chen and S. D. Sharma. A slender ship moving at a near-critical speed in a shallow channel. *Journal of Fluid Mechanics*, 291, `https://doi.org/10.1017/S0022112095002692`, 263–285, 1995.

[44] Courant, R., Friedrichs, K. & Lewy, H. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100, `https://doi.org/10.1007/BF01448839`, 32–74, 1928.

[45] J. A. Cunge and M. Wegner. Numerical integration of Barré de Saint-Venant's flow equations by means of an implicite scheme of finite differences. Applicants in the case of alternately free and pressurised flow in a tunnel. *La Houille Blanche*, 50, `https://doi.org/10.1051/lhb/1964002`, 33–39, 1964.

[46] O. Delestre, C. Lucas, P.-A. Ksinant, F. Darboux, C. Laguerre, T.-N.-T. Vo, F. James, and S. Cordier. SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies. *International Journal for Numerical Methods in Fluids*, 72, `https://doi.org/10.1002/fld.3741`, 269–300, 2013.

[47] L.-C. Dempwolff, G. Melling, C. Windt, O. Lojek, T. Martin, I. Holzwarth, H. Bihs, and N. Goseberg. Loads and effects of ship-generated, draw-down waves in confined waterways - A review of current knowledge and methods. *Journal of Coastal and Hydraulic Structures*, 2, 46, 2022.

[48] S. Doychev, X.-N. Chen, and B. Söhngen. "Theoretical and numerical modelling of ship-induced waves in restricted waterways". Unpublished, under consideration for publication in J. Fluid Mech. 2015.

[49] F. Ducros, F. Laporte, T. Soulères, V. Guinot, P. Moinat, and B. Caruelle. High-Order Fluxes for Conservative Skew-Symmetric-like Schemes in Structured Meshes: Application to Compressible Flows. *Journal of Computational Physics*, 161, 114–139, 2000.

[50] D. R. Durran. In: Nonreflecting Boundary Conditions. 395–438. New York, NY: Springer New York, 1999.

[51] A. Engsig-Karup, J. Hesthaven, H. Bingham, and T. Warburton. DG-FEM solution for nonlinear wave-structure interaction using Boussinesq-type equations. *Coastal Engineering*, 55, 197–208, 2008.

[52] C. Escalante, M. Dumbser, and M. Castro. An efficient hyperbolic relaxation system for dispersive non-hydrostatic water waves and its solution with high order discontinuous Galerkin schemes. *Journal of Computational Physics*, 394, 385–416, 2019.

[53] C. Escalante and T. Morales. A general non–hydrostatic hyperbolic formulation for Boussinesq dispersive shallow flows and its numerical approximation. *Journal of Scientific Computing*, 83, 62, 2020.

[54] N. Favrie and S. Gavrilyuk. A rapid numerical method for solving Serre-Green-Naghdi equations describing long free surface gravity waves. *Nonlinearity*, 30, 2718–2736, 2017.

[55] D. Ferrari and M. Dumbser. A mass and momentum-conservative semi-implicit finite volume scheme for complex non-hydrostatic free surface flows. *International Journal for Numerical Methods in Fluids*, 93, `https://doi.org/10.1002/fld.5017`, 2946–2967, 2021.

[56] J. Ferziger, M. Perić, and R. Street. *Computational Methods for Fluid Dynamics*. Springer International Publishing, 2019.

[57] A. Filippini, S. Bellec, M. Colin, and M. Ricchiuto. On the nonlinear behaviour of Boussinesq type models: Amplitude-velocity vs amplitude-flux forms. *Coastal Engineering*, 99, 109–123, 2015.

[58] E. Gaburro, M. Castro, and M. Dumbser. A well balanced diffuse interface method for complex nonhydrostatic free surface flows. *Computers & Fluids*, 175, 180–198, 2018.

[59] P. Gauckler. Etudes Théoriques et Pratiques sur l'Ecoulement et le Mouvement des Eaux. *Comptes Rendues de l'Académie des Sciences*, 64, 818 –822, 1867.

[60] G. Giorgi, M. Penalba, and J. Ringwood. Nonlinear Hydrodynamic Models for Heaving Buoy Wave Energy Converters. In: *Asian Wave and Tidal Energy Conference*. 2016. 1–10.

[61] D. Givoli. Non-reflecting boundary conditions. *Journal of Computational Physics*, 94, 1–29, 1991.

[62] D. Givoli and B. Neta. High-order nonreflecting boundary conditions for the dispersive shallow water equations. *Journal of Computational and Applied Mathematics*, 158, Selection of papers from the Conference on Computational and Mathematical Methods for Science and Engineering, Alicante University, Spain, 20-25 September 2002, 49–60, 2003.

[63] E. Godlewski, M. Parisot, J. Sainte-Marie, and F. Wahl. Congested shallow water model: on floating body. *The SMAI journal of computational mathematics*, 6, `https://doi.org/10.5802/smai-jcm.67`, 227–251, 2020.

[64] Godlewski, Edwige, Parisot, Martin, Sainte-Marie, Jacques, and Wahl, Fabien. Congested shallow water model: roof modeling in free surface flow. *ESAIM: M2AN*, 52, `https://doi.org/10.1051/m2an/2018032`, 1679–1707, 2018.

[65] G. H. Golub and C. F. Van Loan. *Matrix Computations - 4th Edition.* Philadelphia, PA: Johns Hopkins University Press, 2013.

[66] R. Graham. An efficient algorith for determining the convex hull of a finite planar set. *Information Processing Letters*, 1, 132–133, 1972.

[67] A. Gronarz, J. Zöllner, and R. Henn. *Projekt "Referenzschiff" PMM-Versuche.* Tech. rep. DST e.V. - Development Center for Ship Technology and Transport Systems, 2016.

[68] M. Ioriatti and M. Dumbser. A posteriori sub-cell finite volume limiting of staggered semi-implicit discontinuous Galerkin schemes for the shallow water equations. *Applied Numerical Mathematics*, 135, 443–480, 2019.

[69] J. Journée and W. Massie. *Offshore Hydromechanics.* Delft University of Technology, 2001.

[70] T. Karambas and E. Loukogeorgaki. A Boussinesq-Type Model for Nonlinear Wave-Heaving Cylinder Interaction. *Energies*, 15, `https://doi.org/10.3390/en15020469`, 2022.

[71] R. I Karlsson. The effect of irregular surface roughness on the frictional resistance of ships. In: *Proceedings of International Symposium on Ship Viscous Resistance.* SSPA. Goteborg, Sweden, 1978.

[72] T. Katsaounis, D. Mitsotakis, and G. Sadaka. Boussinesq-Peregrine water wave models and their numerical approximation. *Journal of Computational Physics*, 417, 109579, 2020.

[73] F. Kemm, E. Gaburro, F. Thein, and M. Dumbser. A simple diffuse interface approach for compressible flows around moving solids of arbitrary shape based on a reduced Baer-Nunziato model. *Computers & Fluids*, 204, 104536, 2020.

[74] C. H. Kim. *The influence of water depth on the heaving and pitching motions of a ship moving in longitudinal regular head waves.* Tech. rep. 44. Gothenburg, Sweden: Chalmers University of Technology, Department of Naval Architecture and Marine Engineering, Division of Ship Hydromechanics, June 1968.

[75] S. Kontos. *Robust Numerical Methods for Nonlinear Wave-Structure Interaction in a Moving Frame of Reference.* PhD thesis. Technical University of Denmark. Department of Mechanical Engineering, 2016.

[76] N. Kornev. *Schiffstheorie I.* Berichte aus der Strömungstechnik. Aachen: Shaker Verlag, 2009.

[77] S. C. Kramer and G. S. Stelling. A conservative unstructured scheme for rapidly varied flows. *International Journal for Numerical Methods in Fluids*, 58, 183—212, 2008.

[78] D. Lannes. On the Dynamics of Floating Structures. *Annals of PDE*, 3, `https://doi.org/10.1007/s40818-017-0029-5`, 2017.

[79] G. J. Lee. Self-similarity in the equation of motion of a ship. *International Journal of Naval Architecture and Ocean Engineering*, 6, 333–346, 2014.

[80] B. van Leer. Towards the Ultimate Conservative Difference Scheme V. A Second-order Sequel to Godunov's Method. *Journal of Computational Physics*, 32, 101–136, 1979.

[81] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

[82] C.-C. Liao, Y.-W. Chang, C.-A. Lin, and J. McDonough. Simulating flows with moving rigid boundary using immersed-boundary method. *Computers & Fluids*, 39, 152–167, 2010.

[83] H. Luo, H. Dai, P. J. Ferreira de Sousa, and B. Yin. On the numerical oscillation of the direct-forcing immersed-boundary method for moving boundaries. *Computers & Fluids*, 56, 61–76, 2012.

[84] P. Madsen, H. Bingham, and H. Schäffer. Boussinesq-type formulations for fully nonlinear and extremely dispersive water waves: Derivation and analysis. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 459, 1075–1104, 2003.

[85] P. Madsen, R. Murray, and O. Sørensen. A new form of the Boussinesq equations with improved linear dispersion characteristics. *Coastal Engineering*, 15, 371–388, 1991.

[86] P. A. Madsen and O. R. Sørensen. A new form of the Boussinesq equations with improved linear dispersion characteristics. Part 2. A slowly-varying bathymetry. *Coastal Engineering*, 18, `https://doi.org/10.1016/0378-3839(92)90019-Q`, 183–204, 1992.

[87] R. Manning. On the flow of water in open channels and pipes. *Transactions of the Institution of Civil Engineers of Ireland*, 20, 161 –207, 1891.

[88] D. Pan and T.-T. Shen. Computation of incompressible flows with immersed bodies by a simple ghost cell method. *International Journal for Numerical Methods in Fluids*, 60, 1378–1401, 2009.

[89] M. Parisot. Entropy-satisfying scheme for a hierarchy of dispersive reduced models of free surface flow. *International Journal for Numerical Methods in Fluids*, 91, 509–531, 2019.

[90] M. Parisot. Congested shallow water model: trapped air pockets modeling. *SIAM Journal on Scientific Computing*, 45, B828–B852, 2023.

[91] S. Patankar and D. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15, `https://doi.org/10.1016/0017-9310(72)90054-3`, 1787 –1806, 1972.

[92] A. Pecher and J. Kofoed. *Handbook of Ocean Wave Energy.* . Springer, Cham, Jan. 2017.

[93] D. H. Peregrine. Long waves on a beach. *Journal of Fluid Mechanics*, 27, 815–827, 1967.

[94] .Performance Prediction Method. In: *Proceedings of the 15th International Towing Tank Conference.* International Towing Tank Conference (ITTC). The Hague, The Netherlands, 1978.

[95] A. Preissmann and J. A. Cunge. Calcul des intumescences sur machinas electroniques. In: *Proceedings of 9th Congress of International Association for Hydraulic Research (IAHR).* International Association for Hydraulic Research. Dubrovnik, Yugoslavia, 1961. 656—664.

[96] .Resistance Test. In: *Proceedings of the 8th International Towing Tank Conference.* International Towing Tank Conference (ITTC). Madrid, Spain, 1957.

[97] D. P. Rijnsdorp, H. Wolgamot, and M. Zijlema. Non-hydrostatic modelling of the wave-induced response of moored floating structures in coastal waters. *Coastal Engineering*, 177, 104195, 2022.

[98] D. P. Rijnsdorp and M. Zijlema. Simulating waves and their interactions with a restrained ship using a non-hydrostatic wave-flow model. *Coastal Engineering*, 114, `https://doi.org/10.1016/j.coastaleng.2016.04.018`, 119–136, 2016.

[99] L. Río-Martín, S. Busto, and M. Dumbser. A massively parallel hybrid finite volume/finite element scheme for computational fluid dynamics. *Mathematics*, 9, 2316, 2021.

[100] A. Ritter. Die Fortpflanzung der Wasserwellen. *Zeitschrift des Vereines Deutscher Ingenieure*, 36, 947–954, 1892.

[101] V. Rusanov. The calculation of the interaction of non-stationary shock waves and obstacles. *USSR Computational Mathematics and Mathematical Physics*, 1, `https://doi.org/10.1016/0041-5553(62)90062-9`, 304 –320, 1962.

[102] H. Schlichting and K. Gersten. *Boundary-Layer Theory.* Springer Berlin Heidelberg, 2016.

[103] J. H. Seo and R. Mittal. A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. *Journal of Computational Physics*, 230, 7347–7363, 2011.

[104] Y. Shao, Z. Zheng, H. Liang, and J. Chen. A consistent second-order hydrodynamic model in the time domain for floating structures with large horizontal motions. *Computer-Aided Civil and Infrastructure Engineering*, 37, `https://doi.org/10.1111/mice.12782`, 894–914, 2022.

[105] S. Song, M. Terziev, T. Tezdogan, Y. K. Demirel, C. De Marco Muscat-Fenech, and A. Incecik. Investigating roughness effects on ship resistance in shallow waters. *Ocean Engineering*, 270, 113643, 2023.

[106] T. Soomere. Nonlinear Components of Ship Wake Waves. *Applied Mechanics Reviews*, 60, `https://doi.org/10.1115/1.2730847`, 120–138, 2007.

[107] P. K. Stansby and J. G. Zhou. Shallow-water flow solver with non-hydrostatic pressure: 2D vertical plane problems. *International Journal for Numerical Methods in Fluids*, 28, `https://doi.org/10.1002/(SICI)1097-0363(19980915)28:3<541::AID-FLD738>3.0.CO;2-0`, 541–563, 1998.

[108] G. Stelling and M. Zijlema. An accurate and efficient finite-difference algorithm for non-hydrostatic free-surface flow with application to wave propagation. *International Journal for Numerical Methods in Fluids*, 43, 1–23, 2003.

[109] G. S. Stelling and S. P. A. Duinmeijer. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. *International Journal for Numerical Methods in Fluids*, 43, `https://doi.org/10.1002/fld.537`, 1329–1354, 2003.

[110] J. Stoker. *Water waves : the mathematical theory with applications.* Pure and applied mathematics. Interscience Publishers, 1957.

[111] M. Tavelli and M. Dumbser. A high order semi-implicit discontinuous Galerkin method for the two dimensional shallow water equations on staggered unstructured meshes. *Applied Mathematics and Computation*, 234, 623–644, 2014.

[112] V. Te Chow. *Open-channel Hydraulics.* Civil Engineering Series. McGraw-Hill, 1959.

[113] W. C. Thacker. Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics*, 107, `https://doi.org/10.1017/S0022112081001882`, 499–508, 1981.

[114] C. Tong, Y. Shao, and H. Bingham. Modeling fully nonlinear wave-structure interaction by an adaptive harmonic polynomial cell method with immersed boundaries. In. 36th International Workshop on Water Waves and Floating Bodies. 2021.

[115] E. Toro. *Shock-Capturing Methods for Free-Surface Shallow Flows.* John Wiley & Sons, 2001.

[116] E. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction.* Springer Berlin Heidelberg, 2013.

[117] E. O. Tuck. Hydrodynamic Problems of Ships in Restricted Waters. *Annual Review of Fluid Mechanics*, 10, 33–46, 1978.

[118] M. Vantorre, K. Eloot, G. Delefortrie, E. Lataire, M. Candries, and J. Verwilligen. In: Maneuvering in Shallow and Confined Water. 1–17. John Wiley & Sons, Ltd, 2017.

[119] J. G. Vasconcelos, S. J. Wright, and P. L. Roe. Improved Simulation of Flow Regime Transition in Sewers: Two-Component Pressure Approach. *Journal of Hydraulic Engineering*, 132, `https://doi.org/10.1061/(ASCE)0733-9429(2006)132:6(553)`, 553–562, 2006.

[120] Verein für europäische Binnenschifffahrt und Wasserstraßen and Bundesanstalt für Wasserbau, eds. *Driving Dynamics of Inland Vessels.* eng. `https://hdl.handle.net/20.500.11970/104201`. Karlsruhe: Bundesanstalt für Wasserbau, 2016.

[121] Y. Yang, G. Chillcce, and O. el Moctar. Mathematical modeling of shallow water effects on ship maneuvering. *Applied Ocean Research*, 136, 103573, 2023.

[122] Y. Yang and O. el Moctar. A mathematical model for ships maneuvering in deep and shallow waters. *Ocean Engineering*, 295, 116927, 2024.

[123] M. Zijlema and G. S. Stelling. Further experiences with computing non-hydrostatic free-surface flows involving water waves. *International Journal for Numerical Methods in Fluids*, 48, `https://doi.org/10.1002/fld.821`, 169–197, 2005.