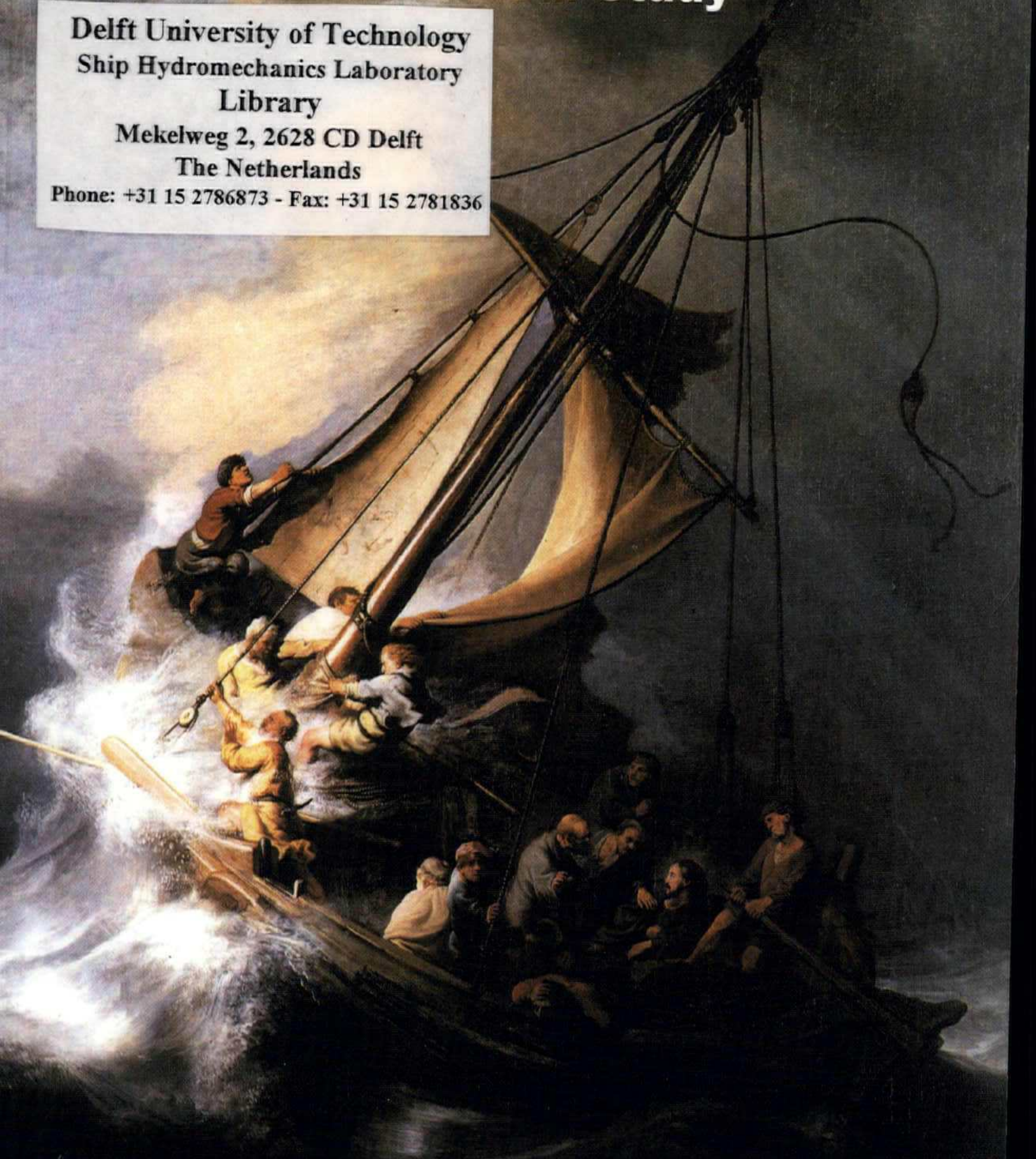
Water Impact Loading on Offshore Structures

- A Numerical Study -



Theresa Kleefsman

Water Impact Loading on Offshore Structures

A Numerical Study

Theresa Kleefsman

The research presented in this thesis was funded by the European Community under the 'Competitive and Sustainable Growth' Programme (EU Project No.: GRD1-2000-25656) and a group of 26 industrial participants (oil companies, shipyards, engineering companies, regulating bodies) as part of the SafeFLOW JIP, coordinated by MARIN. The author is solely responsible for the present thesis and it does not represent the opinion of the European Community.

On the cover: "Storm on the sea of Galilee", Rembrandt, 1633
"And behold there arose a great storm on the sea, so that the boat was being swamped by the waves...", Matthew 8.24

Rijks*universiteit* Groningen

Water Impact Loading on Offshore Structures

A Numerical Study

Proefschrift

ter verkrijging van het doctoraat in de
Wiskunde en Natuurwetenschappen
aan de Rijks*universiteit* Groningen
op gezag van de
Rector Magnificus, dr. F. Zwarts,
in het openbaar te verdedigen op
vrijdag 18 november 2005
om 14.45 uur

door

Kornelia Marchien Theresa Kleefsman

geboren op 18 maart 1977
te Veendam

Promotor: Prof. dr. A.E.P. Veldman

Beoordelingscommissie: Prof. N. Barltrop 1
Prof. dr. ir. B. Koren 2
Prof. dr. ir. J.A. Pinkster 2

ISBN 90-367-2385-X

Stellingen
behorende bij het proefschrift

Water Impact Loading on Offshore Structures

van Theresa Kleefsman

I.

De keuze van snelheidsrandvoorwaarden aan het vrije oppervlak is cruciaal voor de robuustheid van de code, de nauwkeurigheid van golfsimulaties en het ontstaan van pieken in het druksignaal. Helaas is er geen methode die voor bovenstaande drie effecten tegelijkertijd positief werkt, daarom moet gekozen worden voor een 'engineering mix'.

II.

Het idee om een methode te gebruiken met een Navier-Stokes domein ingebed in een domein met een meer eenvoudige methode om golfimpact problemen te simuleren, brengt het mogelijke gebruik van een Navier-Stokes code in praktisch ontwerp een stap dichterbij.

III.

Een goede simulatiemethode dient op alle aspecten waarvoor deze wordt gebruikt, gevalideerd te zijn met behulp van (full-scale) metingen.

IV.

Dat maar een klein deel van het publiek bij een wiskundecolloquium begrijpt waar het over gaat, lijkt zowel spreker als publiek niet te deren.

V.

Onderzoek naar golfkrachten op een winderige en beweeglijke planeet (die voor meer dan de helft met water is bedekt) lijkt geen overbodige luxe.

VI.

Een belangrijke aanwijzing voor het bestaan van God is het detail waarop je kunt promoveren.

VII.

De eerste simulatie van een nieuwe serie gaat altijd goed, de problemen komen bij de tweede.

VIII.

Katten drinken geen vers water dat je in hun bakje hebt gedaan, alleen het water uit gieter, sloot of plas is goed genoeg.

IX.

Hoe duurder het restaurant, hoe meer mannelijke obers.

X.

Het aantrekkelijkste speelgoed voor een klein kind is datgene waarvoor het net een speelverbod heeft gekregen.

Contents

Chapter 1

Introduction

1.1 Problem definition

In the offshore industry a structure for the production and storage of oil or gas is placed at one location for many years, often for over twenty years. These structures must survive all weather types, including heavy storms. The photograph in Figure 1.1 shows a Floating Production, Storage and Offloading vessel (FPSO) in a heavy storm at the North Sea.

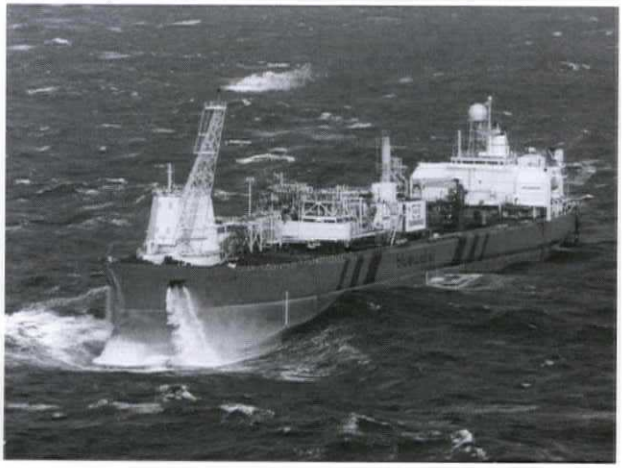
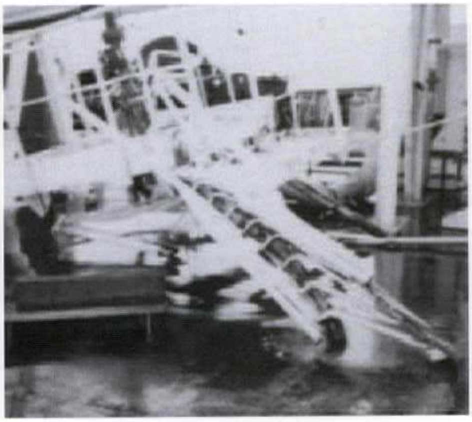


Figure 1.1: Floating Production Storage and Offloading vessel (FPSO) in heavy storm

In rough seas large masses of water can invade the ship's deck when the freeboard is exceeded by high waves. This is called green water after the solid water mass that is coloured green, in contrast to white water that consists of spray and foam mainly. Green water can occur at the ship's bow or at the side of the ship damaging equipment on the deck or living quarters of personnel. Especially on FPSO's, where a lot of sensitive equipment is present at the deck, green water causes a lot of damage. Ersdal et al. [21] and Morris et al. [66] report on green water events at production ships in the North Sea. Figure 1.2 shows two photos with damage caused by green water [7].



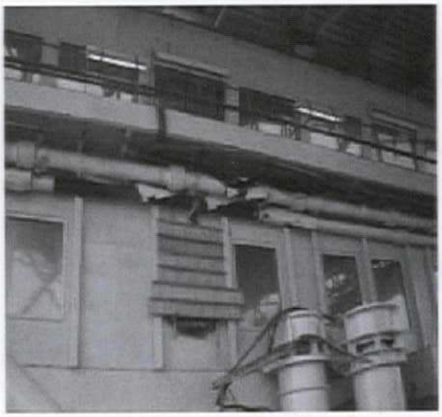


Figure 1.2: Damage to the fire fighting platform of the Emerald FSU (Floating Storage Unit, left) and the damage and first repair of a window after a green water incident on the Varg FPSO (right)

In seas with steep fronted waves offshore structures also face the problem of wave impact on their bow or bottom. One reported case is the Schiehallion FPSO where a rather steep wave had damaged the bow, which finally resulted in an evacuation of all personnel [34]. The photo in Figure 1.3 shows the damage to the front of the bow.

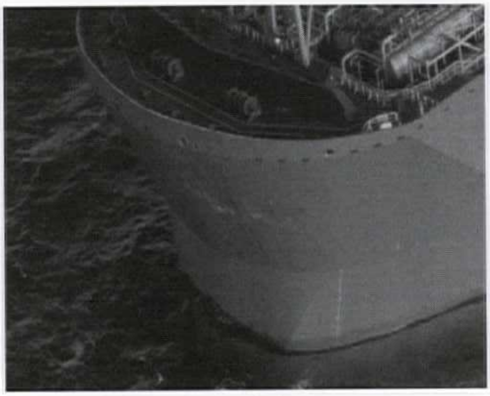


Figure 1.3: Wave impact damage causing dents to the bow of the Schiehallion FPSO

For the design of offshore structures these issues have to be taken into account to prevent economic loss due to damage to the equipment and guarantee as safe conditions as possible for the personnel. Very important in the design phase of a structure are model tests to calculate the loads on different parts of the structure. Model testing is expensive and it would be convenient to replace a part of the model testing by numerical simulation, or in any case focus model tests on critical events only. Therefore, there is a great need for calculation methods that can take into account the highly nonlinear situations that occur in heavy seas. Until recently, most calculation methods have been based upon potential theory and have not been able to calculate local flow phenomena like green water on the deck of a ship or slamming caused by (near-)breaking waves.

This thesis presents a simulation method based on the Navier-Stokes equations that is developed (and is still in development) for the simulation of water loading on offshore structures. The objective of the research presented in this thesis can be phrased by:

To develop a robust and validated numerical method
for the prediction of local wave impact loads on floaters.

This research builds on the work of Gerrits [26] and Fekken [23], who developed a method for local flow phenomena with the presence of a free liquid surface and moving objects. At the boundaries and at the object no-slip or free-slip conditions can be used. With their method sloshing problems, water-entry problems and problems with simple in- and outflow conditions can be handled (like channel flow with a prescribed uniform inflow). The emphasis of the current work has been on validation and further robustness improvement of the existing method and the extension of the method with complex in- and outflow conditions, such that wave impact simulations can be performed. To reach our goal, the following items need to be modelled, implemented and investigated:

- The design and implementation of wave generation options.
- The design and implementation of a zonal modelling method, where the outer domain provides the wave kinematics and vessel motions for the local domain calculation.
- The modelling of wave damping, due to the discretisation.
- The modelling and investigation of outflow conditions to prevent wave reflections.
- The investigation of the handling of the free surface (boundary conditions and displacement) for robust and accurate simulations.
- Generation of pressure output, to analyse impact results.
- The validation of the method on the different stages in impact problems, as there are wave propagation, water entry of a structure, and impact of water on the structure.

Throughout this thesis, all these items will get attention in the description of the model and the results of validation. After all the individual elements are visited a simulation can be performed containing all the elements, like green water on the deck of a moving vessel.

1.1.1 Focus on green water events

Some more attention will be paid to the green water phenomenon as an important application of the simulation method described in this thesis. Buchner [7] has conducted a thorough investigation to the characteristics of green water flow, and the effect of bow shape using model tests. Also some different structures were put on the deck to measure the effect of the shape of the structure (for example a deck house as shown in Figure 1.4).

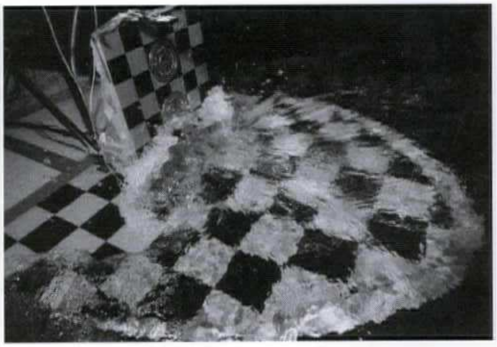


Figure 1.4: Experiment of a green water event on the bow of an FPSO

Green water loading is a highly complex and nonlinear process. Buchner [7] showed that the following phases can be distinguished in the process of green water on the foredeck, see Figure 1.5:

A. The combination of a high wave and the pitch motion of a vessel results in nonlinear swell-up around the bow.

B. The water is almost at rest around the bow, after which it starts to flow onto the deck in a 'dam breaking'-type of flow.

C. This results in a 'Hydraulic jump'-type shallow water flow on the moving deck, focussing into a high velocity water 'jet' when the water fronts from the sides meet.

D. Water impact and water run-up occur in front of the structure, and eventually the water is turning over.

When attempting to use a numerical method to describe these phenomena, the method should be able to deal with complex nonlinear flows. If the focus of the investigation is limited to the local flow around the bow, specifically it should be able to handle:

1. Water entry of a flared bow structure.

2. Complex flow onto the deck, including the discontinuity at the deck edge.

3. 'Hydraulic jump'-type shallow water flow on a moving ship deck.

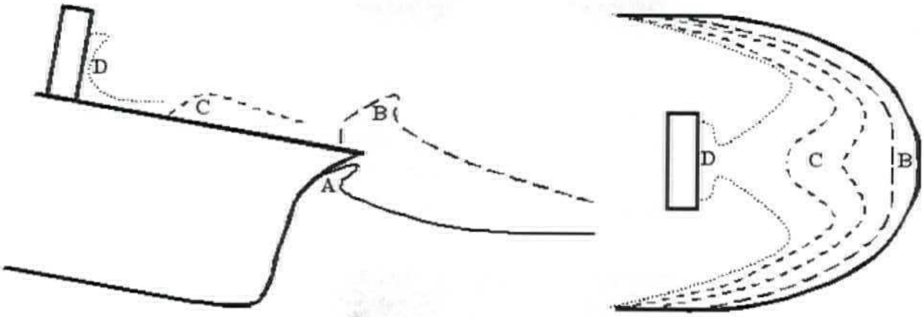


Figure 1.5: The main phases of the green water problem schematically in side view (left) and top view (right): from the nonlinear relative wave motions in front of the bow, via the complex flow onto and on the deck to the impact on deck structures

4. Meeting water flows on the deck.
5. Short duration water impact on a structure.
6. Overturning flow after run-up of the water in front of the structure.

Initial numerical investigations of green water loading focussed on the shallow water flow on the deck, using Glimm's method, see for instance Mizoguchi [63, 64], Zhou et al. [102], and Stansberg et al. [85]. Also (nonlinear) dambreaking theory is used for the simulation of the green water flow on the deck, e.g. Yilmaz et al. [97]. With this type of method only the 'hydraulic jump'-type shallow water flow on the moving deck can be simulated (Phase C). The computational domain can consequently be limited to the area on the deck. The freeboard exceedance around the deck and the related velocities were used as boundary conditions.

Greco et al. [35] use a two-dimensional fixed structure to represent the ship's deck. Waves are overtopping the freeboard of the ship resulting in green water on the deck. The calculations are performed with a Boundary Element Method and the resulting water motion at the deck is compared with experiments. Nielsen et al. [68] take into account all the phases of the green water phenomenon by using a Navier-Stokes solver with the Volume-of-Fluid method for the displacement of the free surface. Waves are generated at the inflow boundary. The ship is fixed, but the relative wave motion is modelled by velocity boundary conditions at the bottom of the domain. The results are compared with experiments of Buchner [7]. Gòmez-Gesteira et al. [33] use the Smoothed Particle Hydrodynamics (SPH) method for the simulation of waves overtopping a fixed deck. The SPH method is a purely Lagrangian method, so no mesh is needed. The resulting wave profile is compared with experiments.

1.2 Description of the simulation method

In this thesis a method is described that in principle can handle all six points needed for the simulation of the green water phenomenon. The method, incorporated in the computer program ComFLOW, is based on the Navier-Stokes equations, which describe the motion of an incompressible, viscous fluid. Originally, the method was developed to simulate liquid sloshing on board spacecraft [29]. In this application the surface tension is the driving force in absence of gravity. Also an accurate description of the free surface is essential. Another application has been found in medical science, where blood flow through elastic arteries has been studied [57, 58, 59].

In the maritime application area sloshing in anti-roll tanks was simulated [17, 18]. Also, a pilot study on the simulation of green water has been performed by Fekken [24]. An approach was used that deals with water impact and water run-up in front of the deck structure. Using the simulation method Fekken was able to simulate the flow on the deck and resulting impact accurately. However, the computational domain was limited to the area on the deck. In the left of Figure 1.6 the initial configuration is shown. The (measured) freeboard exceedance around the deck was used as boundary condition for a breaking-dam type flow. The right picture in Figure 1.6 shows the run-up and falling down of the water from the structure on the deck. The deck was not moving in this approach.

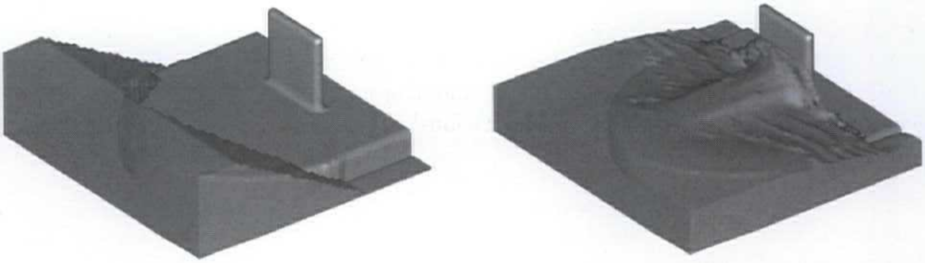


Figure 1.6: Initial configuration and run-up on a deck structure in the approach of Fekken et al. [24] for green water simulation

New applications in the marine field are sloshing in LNG-tankers [94] and wave loads calculation on subsea structures in the splash zone [9]. Also wave-in-deck calculations have been performed, which is an important issue on, for example, the Norwegian Ekofisk oil field [47].

1.2.1 Grid and discretisation

The simulation method described in this thesis numerically solves the Navier-Stokes equations. Thereto, the computational domain is covered by a fixed Cartesian grid. The geometry, which is in general not rectilinear of form, cuts through the Cartesian grid cells, resulting in cut cells. A Cartesian grid is very suitable when simulations are performed including highly distorted fluid interfaces. Then it is not an option to align the grid with the moving interface. Further, the Cartesian grid is a structured grid, which means that all the grid cells have the same number of cell faces per grid cell and the number of cells surrounding each grid point is constant. Another option is to use a boundary fitted unstructured grid. The advantage is that no cut cells are present in this method. But in our method and applications a structured grid is preferable above an unstructured grid, because the fluid interface needs to be kept sharp to correctly predict pressure impact peaks.

When using unstructured grids a diffusive interface method is commonly used, where the interface is smeared over a few cells. Certainly, the Volume-of-Fluid method adopted in our method is not suitable for unstructured grids for bookkeeping reasons. Another disadvantage of an unstructured boundary fitted grid is that when complex geometries are involved, as in simulations of offshore structures, the generation of a boundary fitted grid is very difficult. For moving structures, the grid has to be regenerated every time step, which is very time consuming. For these reasons, a fixed Cartesian grid has been chosen.

Another method that is sometimes used in hydrodynamic applications is the Smoothed Particle Hydrodynamics (SPH) method [65], which is a meshless method. Particles are put in the flow and every particle has a mass and velocity. A particle is influenced by other particles that are within a certain distance from the particle. The method can handle large deformations of the fluid interface automatically. The method has also been used in green water applications [33].

In the current method, the variables are staggered on the Cartesian grid as in the original Marker-and-Cell method [39], which means that the pressure is defined in cell centers and the velocities on cell faces. The advantage of a staggered grid is that mass conservation can be applied easily in a cell, without the need of interpolations. The finite volume method is used for the spatial discretisation of the Navier-Stokes equations. In control volumes conservation of mass and momentum is applied derived from the conservative form of the Navier-Stokes equations. This discretisation is performed in such a way, that the underlying symmetry properties of the continuous operators also hold for the discrete operators. This leads to a stable method where the kinetic energy is only dissipated (due to diffusion) [92]. For the time discretisation the first order Forward-Euler method is adopted. A Poisson equation for the pressure results, which is solved using Successive Over-Relaxation (SOR) with an automatically adapted relaxation parameter [5].

1.2.2 Free surface displacement

A very important aspect of the applications is the presence of a free liquid surface. Many methods for the treatment of the free surface are described in the literature; often the methods for flow calculations with a free surface are classed by the method for the interface treatment. An overview of the various methods available can be found in [80]. The most popular ones are the level-set method and the Volume-of-Fluid method, which is adopted in the current method.

In the level-set formulation a distance function $\phi(x, t)$ is introduced denoting the distance from x to the initial interface location at $t = 0$. The interface corresponds to the contour $\phi = 0$ at any instant [72]. In this method highly distorted interfaces can be treated and also topology changes are incorporated automatically. Although the interface is of finite thickness, the physical properties such as surface tension can be applied easily. A major problem of the level-set method is the lack of mass conservation (see e.g. [54]). Several strategies have been studied to overcome this problem, e.g. a combination with a VOF method has been used [87] or a re-distancing algorithm [86]. In our application area Iafrati et al. used the level-set technique for unsteady free-surface flows, where results of flow over a jump and flow inside a tank have been shown [45]. The level set method is an interface tracking method, where the interface is tracked explicitly. In interface capturing methods, the interface is resolved implicitly, no tracking function is used. Only conservation laws are used, see e.g. [6, 93].

In the Volume-of-Fluid (VOF) method, a VOF function F is introduced with values between zero and one, indicating the fractional volume of a cell that is filled with a certain fluid. Based on this volumetric data, the free surface is reconstructed and displaced, wherefore the method is termed a volume tracking method. The VOF method is extremely suitable in fixed grid simulation methods, where the free surface should be able to have an arbitrary complex topology. For example, in wave simulations the waves are sometimes overturning, such that the interface intersects itself and merges. The VOF-method automatically takes this into account. The earliest volume tracking methods were developed by Noh and Woodward [69], Hirt and Nichols [44] and Youngs [98]. Reviews of the different VOF-methods can be found in [78, 79].

The evolution of the VOF function is given by

$$\frac{DF}{Dt} = \frac{\partial F}{\partial t} + (\boldsymbol{u} \cdot \nabla)F = 0, \tag{1.1}$$

with $\boldsymbol{u} = (u, v, w)$ the velocity vector, t the time and ∇ the gradient operator. This equation states that the interface is moving with the liquid velocity. Every time step the interface is first reconstructed from the VOF data, after which it is advected using Equation 1.1. The different VOF methods are often classed by the features of the interface reconstruction algorithm and the advection of the interface. For the reconstruction step, three different methods can be distinguished: (1) a simple line interface calculation (SLIC), where the interface is said to be parallel to one of the coordinate axes, e.g. [69]; (2) SLIC with the possibility of a stair stepped interface within a cell, e.g. [44, 55]; and (3) piecewise linear interface calculation (PLIC), e.g. [40, 81, 98]. The PLIC method has become very popular in the last decade, because it results in a more accurate interface reconstruction.

In this thesis, both the SLIC method of Hirt and Nichols [44] and the PLIC method of Youngs [98] are used for the displacement of the free surface. To overcome problems in the original Hirt-Nichols' method, which are mass conservation problems and the occurrence of flotsam and jetsam (small droplets disconnecting from the free surface), a local height function is introduced. Both Hirt-Nichols' and Youngs' method with and without a local height function are used in standard kinematic tests and in a dambreak simulation to compare the performance of the different methods.

1.2.3 Moving objects

Moving objects in the domain can be accounted for in different ways. Commonly known in the fixed-grid field of methods are the immersed boundary method, the fictitious domain method and the cut-cell method. The first two methods are treating the boundaries of the object as a special region in a single phase. So, the whole domain is filled with liquid, and body forces (in cells containing the moving object boundaries) account for the presence of the moving objects. In the cut-cell method the object is solid and the sharp object boundary is cutting through the grid cells.

In the fictitious domain method, introduced by Glowinski [31], the flow computation is done on a fixed space region, which contains the moving objects, using a finite element method. Lagrange multipliers are defined on the regions occupied by the rigid bodies to match the fluid flow and rigid body motion velocities over the interface between the regions. A variational formulation is derived involving Lagrange multipliers to force the rigid body motion inside the moving objects.

The immersed boundary method was originally developed by Peskin [73] for the calculation of blood flow through the heart. The heart is embedded in a larger periodic box that is completely filled with fluid. The interface between the fluid and non-fluid regions is defined using polynomial fitting through markers on the interface. To describe the extra forces in the non-fluid regions, a force is introduced that differs from zero only in these non-fluid regions. The interface force is spread to the nearby grid points using a discrete δ-function. This δ-function typically influences a band of four cells. The material properties are smoothened over the interface in a transition zone with a size of approximately two times the cell size.

The cut-cell method differs from the other methods in that the interface stays sharp and is not smeared over a few cell widths. A sharp interface method is needed in the applications studied in this thesis, since the peak of the water impact pressure should not be flattened due to smearing the interface over a few cell widths. In the current method the initial geometry is filled with markers with accompanying small rectangular volumes, such that the unity of all the volumes equals the object geometry. Then the markers are moved every time step according to the motion of the object. In a cut-cell method the cells can become arbitrary small when a large part of the cell is occupied by geometry. In case of fixed objects, due to the choices in the discretisation of the current method the small cells induce no extra limitations on the time step for stability [20, 92]. In the case of moving objects, a modification in the governing equations is needed to return to the same stability criterion for the time step [23].

A disadvantage of the cut-cell method is that the sudden changes of the nature of cells, from fluid to body cell and vice versa, introduce discontinuities. But by avoiding the smearing of the interface, the velocity of the fluid along moving objects, that is important in the applications at hand, is not smoothened over the object interface. Udaykumar et al. [91] use a cut-cell method for the simulation of flow with complex moving boundaries. To account for the changing nature of a cell from solid to fluid, in their method such a cell is merged with a neighbouring cell during that time step, such that the fluid kinematics in that cell are known.

1.3 Generation and propagation of waves

For the calculation of loads on offshore structures a wave generation option in the simulation method is essential. Some parts of the loads calculation can be done without the presence of waves. Part of the calculation of loads due to green water can be done without waves by modelling the water around the bow using a breaking dam model. This has been used in the early stage of the development of this simulation method [24] and also by e.g. [74, 97, 100]. Water entry simulation can be used to model the early stage of the green water phenomenon when the bow of the vessel enters the water [8]. When waves are present in the domain the phenomenon can be studied entirely. Also for the calculation of, for example, bow slamming, waves (especially steep waves) are essential.

There are three different possibilities to model wave generation in a simulation method like ComFLOW. First, the waves can be generated using a wave maker as is also done in a wave tank. The wave maker is modelled by a moving flap that can move horizontally and/or rotate about different axes. In general, this option is not very convenient in a numerical wave tank, except when exactly the same conditions need to be generated as in an experiment, of which the flap motions are known. The second option is to generate waves at an inflow boundary by prescribing velocities and water height. The velocities and wave height can be calculated using description methods of waves. The third option is to use another efficient simulation method that calculates the wave field, and prescribe velocities calculated by this efficient method at the open boundaries of the ComFLOW domain. The second and third option are elaborated below.

1.3.1 Generation of waves using wave description theory

A wave description theory is used to generate the waves at the inflow boundary of the domain. Note that at the inflow boundary positive and negative velocities can occur, so fluid can flow in and out. The term is used to indicate a boundary where the wave is generated. The wave is generated by prescribing velocities at the inflow boundary. Different kind of wave descriptions can be used to determine the velocities at the inflow boundary. The easiest is a linear wave description (Airy wave). But according to Le Méhauté [61], the range of suitability of linear theory in deep water is $H/\lambda < 0.0062$, with H the wave height and λ the wavelength. This means that only waves with a very small amplitude can be generated using linear theory accurately. In the application area of our interest the waves are nonlinear. To prescribe nonlinear waves, 5th order Stokes theory has been implemented [83]. Irregular waves can be generated by making a superposition of linear wave components. The superposition principle only holds for linear waves, so the accuracy can be insufficient when using this in nonlinear circumstances.

In head waves the wave is only generated at one boundary of the domain. At the other open boundaries a condition should be used, such that the wave can leave the domain undisturbedly. This is a difficult problem, because no information is present about the wave near the outflow boundaries. There are different options to prevent the wave from reflecting against the open boundaries into the domain. Givoli [30] gives an overview of the different outflow boundary conditions. First, a dissipation zone can be used, in which the wave is damped. Second, a non-reflecting boundary condition based on the wave equation can be used that determines the velocities at the outflow boundary. An example is the Sommerfeld boundary condition, where the wave velocity, which occurs in the wave equation, has to be chosen on forehand [84]. Both methods for letting the wave flow out of the domain undisturbedly are investigated in the current simulation program.

1.3.2 Wave field calculation by an external program

Waves can be generated using a calculated wave field by an external program. For example, the initial flow field can be calculated by means of a linear diffraction calculation and a conversion to the time domain. The linearised motion of the object is then known in advance as well. During the simulation, the incoming, diffracted and radiated fluid velocities are imposed on all the open boundaries. There are several advantages in using this method. First, the simulation with ComFLOW can be limited to the close surroundings of the structure. The far wave field is calculated using the external code and imposed on the boundaries of the ComFLOW domain. Second, a long duration run can be performed with the fast external program, of which a time trace can be selected that is expected to give a critical event. This long duration run (typically three hours) cannot be performed with the ComFLOW program for calculation time reasons. Third, the simulation can be started with a fully developed wave field that is calculated by the external code. Also the motion of the structure is known in advance. Fourth, during the time domain simulation a good prediction of the velocities at the open boundaries of ComFLOW is given by the external code calculation.

This zonal modelling, where the domain is decomposed in a small ComFLOW domain and a large external domain is first developed using a linear diffraction code. An investigation should reveal how far this method can be stretched. The next step could be to prescribe the velocities from the incoming contribution using a higher order method. The radiated and diffracted velocity contributions and the vessel motion still follow from the linear code.

1.3.3 Wave generation for validation using an experiment

When an experiment where a wave is involved is used for validation of the simulation program, the exact wave conditions need to be generated. If the wave is not the same in simulation and experiment, the results in loading will be different and the validation results will be less valuable. There are different ways to imitate the conditions in a wave basin. First, the flap motions, with which the wave was generated in the experiment, can be used to prescribe the motion of flaps in the numerical wave tank. This seems to be a rather good option, but a large problem arises in practice. The distance from the wave maker to the structure in the wave is mostly several wave lengths. In a computational intensive program like ComFLOW this distance is too large to be able to accurately calculate the loads on the structure, since not enough computational cells can be put in the neighbourhood of the structure.

The second option is to generate the wave at the inflow boundary of the numerical wave tank using a wave description of linear or 5th order Stokes theory. The advantage is that the exact same wave can be prescribed as used in the experiment and that the inflow boundary can be put at any distance from the structure. The disadvantage is that the disturbance in the wave that is present due to the structure is not taken into account at the inflow boundary. ?

The third option is to use measurements of the wave elevation in front of the structure to prescribe the wave at the inflow boundary at the same distance from the structure. The measured wave elevation can be decomposed using Fourier analysis and the different components can be superposed at the inflow boundary. This is only valid in linear circumstances, so the error in the wave elevation should be investigated carefully. Another disadvantage is that the position of the inflow boundary is fixed at the position of the measured wave elevation. 2

1.3.4 Wave propagation

After the wave is generated at the inflow boundary of the computational domain, it propagates through the domain. The propagation of waves needs to be studied carefully, since the steepness and height of the waves determine the loads on the structures in the waves. A thorough wave propagation study with the current simulation program has been performed by Meskers [62]. He investigated the influence of grid sizes and time step and identified some issues that need to be taken into account for proper wave propagation. The influence of the boundary conditions at the free surface on the wave propagation is very large as was also concluded by Chan et al. [13]. So, valid boundary conditions need to be chosen that take care of accurate wave simulations. The accuracy of wave simulations is also influenced by the method for the displacement of the free surface. As described before, the method of Hirt and Nichols [44] and the method of Youngs [98], which are both Volume-of-Fluid methods, are used. Also a combination of these methods with a local height function is established; this combination ensures almost exact mass conservation. Further, attention has to be paid to the dissipation due to the artificial viscosity that could cause damping of the waves. The artificial viscosity is present due to the upwind discretisation of the convective term in the Navier-Stokes equations. The influence can be investigated by performing a simulation over a large number of periods in a large computational domain. The damping of the waves should be small, such that the loads on the structures in the waves are not underestimated.

1.4 Outline

In this thesis simulation of waves resulting in loading on offshore structures is presented. To describe the method to accomplish this task the thesis is outlined as follows. The first chapter is an introduction to the problem and the computational method. In the second chapter the simulation method is described without the presence of waves. Chapter 3 deals with the numerical aspects of wave simulation. In Chapter 4 the first steps in the development of a domain decomposition method for the generation of waves is presented. The last chapter, Chapter 5, contains a summary of the results. Conclusions are drawn and recommendations are made for future research. ?

Chapter 2 describes the method for fluid-flow simulation in a closed domain, so without the presence of waves. First, the mathematical model is described in Section 2.2. The governing equations are the continuity equation and the Navier-Stokes equations, which describe conservation of mass and momentum, respectively. The boundary conditions at the objects and the free surface are described. In Section 2.3 the spatial and temporal discretisation of the governing equations are discussed, and the stability of the resulting discretised equations is examined. Also the method for the determination of the discrete boundary conditions at the free surface is given.

The displacement of the free surface is discussed in Section 2.4. Two methods are described and compared using standard kinematic tests. The first method is the standard VOF method of Hirt and Nichols [44]. This SLIC method, which is efficient and easy to implement, has the drawback that flotsam and jetsam occur. To prevent this, a local height function is introduced. The second method that is discussed is Youngs' algorithm, which is a PLIC method. Besides the standard kinematic tests, where the velocity field ?

is prescribed, also a case study of a two-dimensional dambreak is presented. Section 2.5 shows that our treatment of the moving objects, using a cut-cell method, introduces numerical pressure spikes. A method to minimise the amount of spikes is discussed.

The simulation method is validated in Section 2.6 and Section 2.7. First, a dambreak simulation is performed, where the loading on a box in the domain is calculated. The computed pressures and water heights are compared with measurements. In Section 2.7 some water entry cases are studied and resulting free surface profiles and pressure time traces are compared with experimental results and theoretical predictions.

In Chapter 3 the numerical issues regarding the simulation of waves are discussed. In Section 3.2 the mathematical description of linear waves is given and the implementation of 5th order Stokes waves is shortly described. The theoretical values for velocities and wave height are used at the inflow boundary of the computational domain to generate waves. The issues at the open domain boundaries, which also include the outflow boundaries where waves should leave the domain undisturbed, are discussed in Section 3.3. The influence of the free surface velocity conditions and the displacement algorithm on the accuracy of wave propagation is shown in Section 3.4 and Section 3.5, respectively.

The validation of wave propagation is performed in Section 3.6. First, the waves are propagating without an object in the flow. The waves are two-dimensional of form. Regular waves are used and irregular wave events, where the waves are quite steep and/or high. Second, a fixed spar platform has been put into the waves, and the resulting wave loading is calculated and compared to experimental results. Finally, a floating FPSO is put into a high wave field resulting in green water on the deck. The water height on the deck and pressure time traces at the deck and the deck house are compared with measurements.

In Chapter 4 the waves are not generated within our simulation program, but the wave field is calculated by an external program that also provides the motion time traces of the object. Section 4.2 describes this domain decomposition and the interface between our simulation method and the external program. In Section 4.3 this method is used to simulate the propagation of an irregular wave. The same experimental results shown in Chapter 3 are used in Section 4.4 to compare this new method for wave generation with. Finally, green water loading on a floating FPSO is calculated using this method, so also the motion of the FPSO is prescribed from the external program, and again compared with experiments as in Chapter 3. In Section 4.5 the results are discussed and some recommendations for future research are given.

Chapter 2

Fluid Flow in a Closed Domain

2.1 Introduction

Wave impact physics and green water shipping on the bow of a vessel can be divided into several aspects. In the first stage of the research, the waves are left out of the problem, which gives the possibility to focus on local impact phenomena. To study this, impact simulations have been performed of dambreaking problems with and without an object in the flow. The advantage of studying a dambreak problem is that the simulations can be performed in a closed domain. The flow in the domain cannot be disturbed by issues at the inflow and outflow boundaries. To validate the method for moving objects, drop test simulations have been performed. The shape of the free surface and impact forces on the dropped objects have been compared with available theory and experiments. In this chapter, the mathematical and numerical model used in closed domain simulations are described.

The governing equations of fluid flow are the continuity and Navier-Stokes equations describing conservation of mass and momentum respectively. At the domain walls and at the objects in the domain no-slip boundary conditions are prescribed. At the free surface continuity of tangential and normal stresses is demanded. Also the capillary effects are taken into account (although not always necessary from a physical point of view).

To solve the mathematical equations, the domain is covered with a fixed Cartesian grid. The geometry is described by linear components, resulting in so-called cut cells where the geometry cuts through the grid cells. A labelling system has been used to distinguish between cells of different character, for example cells filled with fluid, empty cells or cells containing the free surface. The continuity and Navier-Stokes equations are being solved using a finite volume discretisation in space and a forward Euler method in time. This discretisation leads to an energy conserving method. For the free surface treatment two methods are described based on the Volume of Fluid (VOF) method as introduced by Hirt and Nichols [44] and Youngs [98]. To prevent jetsam and flotsam that occur in the original VOF method and to take care of full mass conservation, the method has been improved using a local height function.

2.2 Mathematical Model

The flow of a homogeneous incompressible viscous fluid can be described using the conservation laws of mass and momentum. Here, the resulting equations will be given in conservation form.

2.2.1 Governing Equations

Consider an arbitrary volume V with boundary $S = \partial V$, including part of a moving object as shown in Figure 2.1. The shaded area denotes the part of the moving object V_b, while V_f is the fluid part of the volume. The boundary of volume V can be divided in a solid geometry part $S_b = \partial V_b \cap \partial V$ and a fluid part $S_f = \partial V_f \cap \partial V$. The boundary between the fluid and solid part is called $S_{bf} = \partial V_b \cap \partial V_f$. For a homogeneous incompressible viscous fluid, conservation of mass in this domain over the boundary of V_f results in the equation

$$\oint\limits_{\partial V_f} \boldsymbol{u} \cdot \boldsymbol{n}_f d\partial V_f = \int\limits_{S_f} \boldsymbol{u} \cdot \boldsymbol{n}_f dS_f + \int\limits_{S_{bf}} \boldsymbol{u}_b \cdot \boldsymbol{n}_f dS_{bf} = 0. \tag{2.1}$$

Here, $\boldsymbol{u} = (u, v, w)$ is the velocity vector with u, v and w the velocities in the three coordinate directions x, y and z, respectively, and $\boldsymbol{u}_b$ is the velocity of the moving object. The normal vector of the fluid part of volume V is denoted by $\boldsymbol{n}_f$. Equation (2.1) is referred to as the continuity equation.

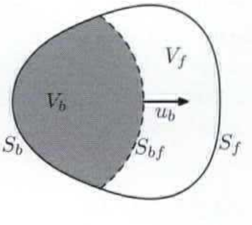


Figure 2.1: Volume V where conservation of mass and momentum is applied

Applying conservation of momentum to a homogeneous incompressible viscous fluid leads to the following equation (which can also be split into three equations, one for each coordinate direction)

$$\int\limits_{V_f} \frac{\partial \boldsymbol{u}}{\partial t} dV_f + \int\limits_{\partial V_f} \boldsymbol{u}(\boldsymbol{u} \cdot \boldsymbol{n}_f) dS_f = -\frac{1}{\rho} \int\limits_{\partial V_f} (p\boldsymbol{n}_f - \mu \nabla \boldsymbol{u} \cdot \boldsymbol{n}_f dS_f + \int\limits_{V_f} \boldsymbol{F} dV_f. \tag{2.2}$$

In these equations, the successive terms have the following meaning. The first term of the left-hand side describes the change of momentum in volume V_f, the second term describes convection of fluid. In the first term of the right-hand-side, two parts can be distinguished: a pressure term (given by p) and a diffusive term, where μ denotes the dynamic viscosity. The terms are divided by the density ρ. The second term on the right-hand-side represents external forces $\boldsymbol{F}$, like gravity, or forces coming from the motion of the coordinate system. In our case, the only external force present is gravity:

$$\boldsymbol{F} = \boldsymbol{g} = (g_x, g_y, g_z) = (0, 0, -9.81)\ m/s^2.$$

The continuity and Navier-Stokes equations will be solved inside one fluid, which in this thesis is water. In the simulations a second fluid, air, will be present, in which no equations will be solved. Both fluids are separated from each other by a free surface. The position of the free surface is not known in advance, so it has to be computed during the simulation process. The evolution of the free surface is given by

$$\frac{Ds}{Dt} = \frac{\partial s}{\partial t} + (u \cdot \nabla)s = 0$$

where $s(\boldsymbol{x}, t) = 0$ gives the actual position of the free surface.

2.2.2 Boundary Conditions

To solve the governing equations, boundary conditions are needed at the solid walls and the free surface. At the solid boundary the no-slip boundary condition for viscous fluids is applied

$$\boldsymbol{u} = \boldsymbol{u}_b,$$

with $\boldsymbol{u}_b = 0$ for fixed objects and solid domain boundaries. For fixed objects, this means that no fluid can go through the wall and also the velocity tangential to the solid boundary is zero.

At the free surface, the forces are balanced, and the resulting equation is split in the normal and tangential direction. When the fluid is incompressible and the curvature of the free surface is neglected in the viscous stress terms, this results in the following equations:

$$-p + 2\mu\frac{\partial u_n}{\partial n} = -p_0 + \sigma\kappa, \tag{2.3}$$

$$\mu\left(\frac{\partial u_n}{\partial t} + \frac{\partial u_t}{\partial n}\right) = 0. \tag{2.4}$$

In the first equation, n denotes the normal of the free surface and t is the tangential direction. Further, u_n and u_t are the normal and tangential component of the velocity respectively, and p_0 is the atmospheric pressure. The surface tension is given by σ, and κ denotes the total curvature of the free surface.

2.3 Discretisation of the governing equations

In this section the numerical method adopted for the fluid flow simulations will be described. The description will be in two dimensions. In most situations this can be extended to three dimensions straightforwardly.

2.3.1 Grid and geometry definition

The governing equations are discretised in space on a fixed Cartesian grid. This is in contrast to a geometry-aligned grid, which is also widely used. The advantage of a geometry-aligned grid is that the geometry does not cut through the grid cells as it will in a Cartesian grid. But in the case of a moving object that can be as complicated as a ship in the domain it is very difficult and time consuming to generate a new grid every time step.

To be able to simulate fluid flow in an arbitrary complex geometry, the geometry is defined using a finite element description. This finite element description is mapped to the Cartesian grid, by checking for points in the grid if they are inside or outside the union of elements. For the definition of the geometry on the Cartesian grid, volume and edge apertures are introduced. Every cell has a volume aperture F^b and two edge apertures (A^x and A^y, in three dimensions also A^z) that define which part of the cell and cell faces, respectively, are open to fluid. (So $1 - F^b$ is the solid body proportion.) The volume and edge apertures are calculated using a sub-grid of so-called integration points. In Figure 2.2 an example of the calculation of apertures is given for two integration points per direction. Using the 4 interior points, $1 - F^b$ is calculated as the number of points inside the geometry divided by the total number of points inside the cell. So in this case, $1 - F^b = 0.25$ and $F^b = 0.75$. The same procedure is used for the edge apertures, which results for the example in $A^x = 0.5$ and $A^y = 0.5$. The number of integration points can be chosen freely. One integration point results in a 'staircase' geometry; increasing the number of integration points gives smoother geomtries. The geometry and boundary apertures are restricted by $0 \leq F^b,\ A^x,\ A^y \leq 1$.

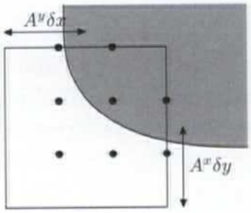


Figure 2.2: Apertures calculation in a cut cell using integration points; dark grey denotes solid body

Besides the volume aperture another function has been introduced to identify the fraction of a cell that is filled with fluid: the Volume-Of-Fluid (VOF) function, denoted by F^s. For every cell, the VOF-function is defined. Because there can be no more fluid in the cell than the open part of the cell, the VOF-function is limited by $0 \leq F^s \leq F^b \leq 1$.

2.3.2 Motion of a rigid object

Displacement of the object

In the domain an object can be present that moves according to a prescribed or calculated motion. Every time step the object is moved, so new geometry apertures for the cell volumes F^b and the cell edges A^x, A^y have to be calculated. This calculation must be as accurate as possible, because it has a large influence on the smoothness of the pressure field. When the apertures are not calculated exactly, the object seems to be 'breathing' in time, which causes irregularities in the pressure time signal.

In two dimensions the apertures can be calculated in an exact manner. In [23] a procedure to perform this calculation has been described. The object is defined as a polygon, with the vertices of the polygon stored in an array. The vertices are displaced every time step exactly, according to the motion of the object. Then, the cross-section of a side of the polygon with a computational cell is calculated, from which the volume and edge apertures of the cells are determined. In two dimensions this procedure is not very complicated, and the resulting apertures are determined almost exactly. To perform the same procedure in three dimensions, where polyhedrons instead of polygons are defined, is much more complicated. The calculation of cross-sections of polyhedrons with a rectangular grid is not straightforward. Therefore, this accurate procedure is not applied in three dimensions, but a more simple method has been adopted, which approximates the three-dimensional body geometry.

The general procedure can be described in three steps. First, the starting geometry is stored in a special way using markers. Then, every time step the volume apertures are calculated by moving the markers. Finally, the edge apertures are calculated, based on the volume apertures. Below, the three steps are described in detail.

1. At the start of a simulation the geometry is built from the finite element description given by the user. Integration points, forming a sub-grid, are used to calculate volume and edge apertures as explained in Section 2.3.1. If a moving object is present in the domain, the geometry of the object should be stored, such that it can be moved every time step. Therefore, the integration points of each cell, forming a set of markers, are stored in an array. Around each marker a small rectangular volume is defined, such that the union of all the volumes forms the object. In case of one marker per cell, these volumes are chosen exactly as one computational cell. When more than one marker per cell is used as in the left of Figure 2.3, the union of the volumes around the markers of one computational cell equals the volume of the cell (in the case that the complete cell is object). When this procedure is followed for all markers coming from the integration points, many points and volumes should be stored. If n_d is the number of integration points, the total number of points to

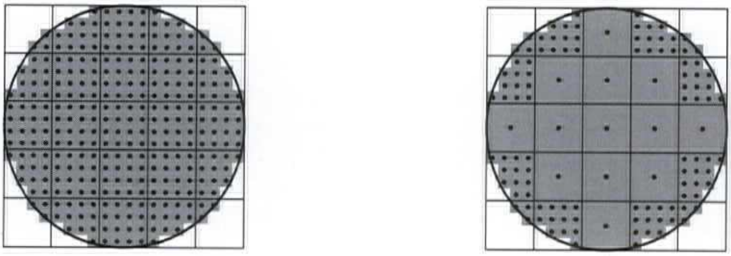


Figure 2.3: A circular cylinder with a sub-grid of markers and volumes; all markers (left); markers that will be stored (right)

be stored are $n_d^3\, n_{cells}$, with n_{cells} the number of computational cells. To prevent unnecessary storage, the markers in a computational cell that is completely solid will be replaced by one marker with accompanying volume equal to the volume of the computational cell. This significantly reduces the total number of markers. An example of the markers and volumes that are stored in case of a two-dimensional circular object is given in the right of Figure 2.3. The number of integration points equals 4 in this case. When a cell is completely occupied by solid, the 16 markers inside the cell are replaced by one marker.

2. Every time step, the volume and edge apertures in the computational grid change. In this step, the new volume apertures are calculated, with the use of the markers and volumes defined in the previous step. First, the markers are moved according to the motion of the rigid object. In case of a rotation of the object, also the volumes belonging to the marker cells should be rotated. To calculate volume apertures, the cross-sections of the marker volumes with the computational cells have to be calculated. For a general rotated volume, this is very complicated in three dimensions. To avoid the calculation of these difficult cross-sections, the marker volumes are not rotated, but are staying grid aligned as in the right of Figure 2.4. There are some errors introduced by keeping the volumes grid aligned. Firstly, small holes can be created inside the object and secondly, small overlapping regions can occur. This should be dealt with during the calculation of the volume apertures. But the errors introduced this way are not very large. According to [23], the errors in the calculation of the volume apertures F^b usually are smaller than 0.01%.

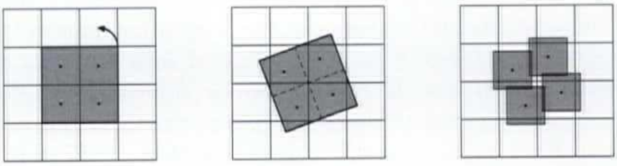


Figure 2.4: Rotation of a square: starting situation (left); exact rotation (middle); rotation where the marker volumes are kept grid aligned (right)

3. After the volume apertures have been calculated at the start of every time step, the edge apertures must be determined. The edge apertures are calculated using a piecewise linear reconstruction of the geometry. This method is often used for the reconstruction of the interface between two fluids as explained in e.g. [78] and also in Section 2.4.4 of this thesis. First, in every cell the normal of the body is calculated based on the filling ratio's of adjacent cells. Using this, a linear approximation of the body geometry in the cell is created, given the filling ratio of the cell (see the figures in Section 2.4.4 for a more detailed explanation of the procedure). The edge apertures are determined by the fractions of the cell faces that are cut by the linear approximation. In [23] it has been shown that the edge apertures calculated in this way behave smoothly in time.

The above described procedure for the calculation of volume and edge apertures is not exact, but only approximates the original geometry. To investigate the influence of the approximation, a simulation has been performed of a solid square moving to the right with uniform velocity in a domain completely filled with fluid. In Figure 2.5, the pressure signal in a cell left from the square has been shown. The pressure signal for the exact apertures calculation is smoother than the approximate calculation, but the overall picture is the same. The regular jumps in the pressure signal of the exact apertures calculation are due to the position of the pressure inside a cell in the discretisation of the momentum equations (see [23]). These jumps can still be recognised in the approximate apertures calculation.

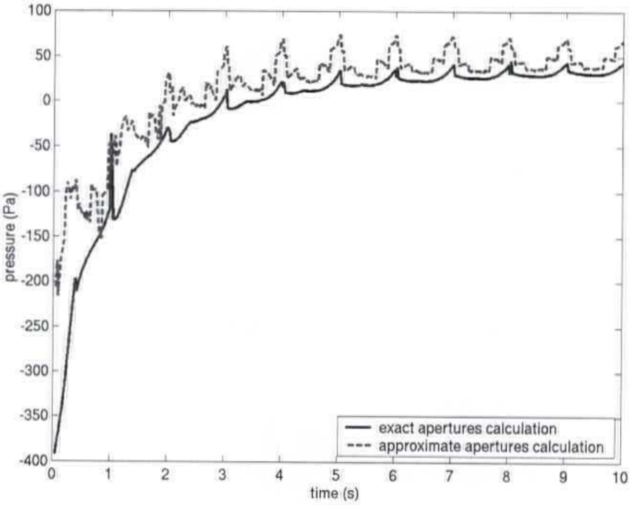


Figure 2.5: Pressure in the fluid containing a moving square: exact calculation of apertures versus approximate calculation

Definition object motion

During the simulation an object can be displaced using prescribed motion time series. Especially in the definition of the rotational motion of an object, it is very important to have the coordinate system and the meaning of rotation angles clearly defined. In our method the following convention has been adopted. The coordinate system is right handed. When the bow of a ship is pointing towards the positive x-axis, the positive angles are defined by:

- Positive roll means that the starboard side of the ship is going down.
- Positive pitch means that the bow of the ship is going down.

- Positive yaw means that the bow of the ship is turning to port side.

This has been illustrated in Figure 2.6 where a grid-aligned box has been given 10 degrees roll, pitch and yaw.

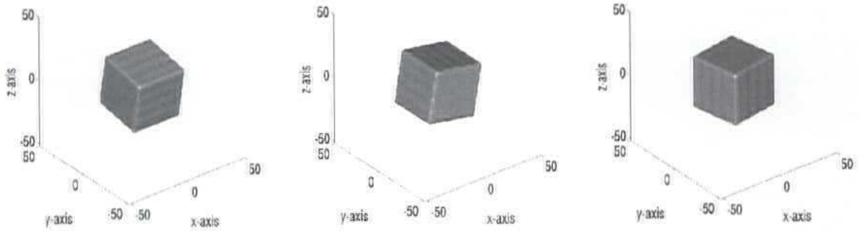


Figure 2.6: A box with 10 degrees roll (left), 10 degrees pitch (mid) and 10 degrees yaw (right)

The matrix describing the complete rotation is given as a product of the three matrices which describe rotation about the three coordinate axes. The general rotation matrices for rotations about the x, y and z-axis are given by

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix},$$

$$R_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix},$$

$$R_z(\gamma) = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The general rotation has been done in the order yaw, pitch, roll, which gives for the general rotation matrix

$$R(\alpha, \beta, \gamma) = R_x(\alpha)\ R_y(\beta)\ R_z(\gamma) =$$

$$\begin{pmatrix} \cos(\beta)\cos(\gamma) & -\cos(\beta)\sin(\gamma) & \sin(\beta) \\ \cos(\gamma)\sin(\alpha)\sin(\beta) + \cos(\alpha)\sin(\gamma) & \cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\beta)\sin(\gamma) & -\cos(\beta)\sin(\alpha) \\ -\cos(\alpha)\cos(\gamma)\sin(\beta) + \sin(\alpha)\sin(\gamma) & \cos(\gamma)\sin(\alpha) + \cos(\alpha)\sin(\beta)\sin(\gamma) & \cos(\alpha)\cos(\beta) \end{pmatrix}.$$

Any point $\boldsymbol{x}_P$ in the object can than be rotated about the center of gravity of the object (CG) using

$$\boldsymbol{x}_P = CG + R(\alpha, \beta, \gamma)(\boldsymbol{x}_{P_0} - \boldsymbol{CG}_0),$$

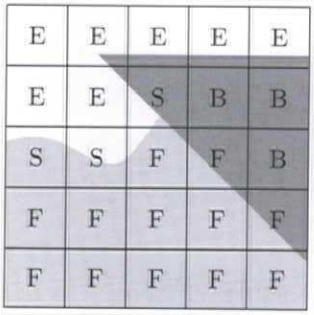
with $\boldsymbol{x}_{P_0}$ the initial coordinates of point $\boldsymbol{x}_P$ and $\boldsymbol{CG}_0$ the initial position of the center of gravity. The three velocity components of a point $\boldsymbol{x}_P$ in the object are given by $\boldsymbol{u}_{COM} + \boldsymbol{\omega} \times \boldsymbol{x}_P$ where $\boldsymbol{u}_{COM} = (u_{COM}, v_{COM}, w_{COM})$ contains the three linear velocity components and $\boldsymbol{\omega}$ is the angular velocity in three directions. When this is written out, the following formulas give the velocity of the object in point $\boldsymbol{x}_P$:

$$\begin{aligned} u_{obj} &= u_{COM} + \omega_2(z_P - z_{com}) - \omega_3(y_P - y_{com}), \\ v_{obj} &= v_{COM} + \omega_3(x_P - x_{com}) - \omega_1(z_P - z_{com}), \\ w_{obj} &= w_{COM} + \omega_1(y_P - y_{com}) - \omega_2(x_P - x_{com}), \end{aligned}$$

with $(x_{com}, y_{com}, z_{com})$ the coordinates of the center of mass.

2.3.3 Cell labelling

Based on the geometry definition and VOF-function (that defines for each cell which part of it is occupied by fluid) the cells are labelled to distinguish between cells with a different character. In Figure 2.7 an example of the cell labelling in the neighbourhood of a ship has been given. First, the cells that are completely in the solid geometry are labelled as B(oundary) cells. Then, the cells containing no fluid are labelled E(mpty) cells. The cells containing fluid adjacent to empty cells are labelled S(urface) cells and always contain part of the free surface. The remaining cells are labelled as F(luid). Note, that these cells do not have to be completely filled with fluid, as the central F-cell in the example.



E	E	E	E	E
E	E	S	B	B
S	S	F	F	B
F	F	F	F	F
F	F	F	F	F

Figure 2.7: Cell labelling: boundary, empty, surface, and full cells

On the Cartesian grid, the variables are staggered, which means the pressure is defined in the cell centers, whereas the velocities are defined on the cell faces. Therefore, also the cell faces are labelled. For example, the velocity on a cell face between a surface cell and an empty cell is called an SE-velocity.

2.3.4 Discretisation of the continuity equation

For the spatial discretisation of the continuity and Navier-Stokes equations the finite volume method is adopted. For the discretisation of the continuity equation, Equation (2.1), consider a computational cell cut by a part of a moving geometry as shown in Figure 2.8. Here, u_e and u_w are the horizontal velocities defined in the center of the open part of the eastern and western cell face, respectively. The same definition holds for the northern and southern vertical velocities v_n and v_s.

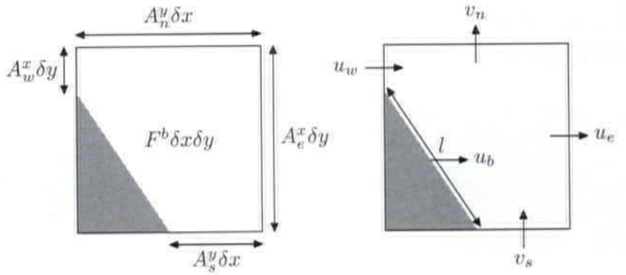


Figure 2.8: Conservation cell for the continuity equation

When applying conservation of mass in this cell, which means that the sum of the mass fluxes through the cell boundaries should vanish, the discretisation results in

$$u_e A^x_e \delta y + v_n A^y_n \delta x - u_w A^x_w \delta y - v_s A^y_s \delta x + l(\boldsymbol{u}_b \cdot \boldsymbol{n}^b) = 0, \tag{2.5}$$

where the notation is explained in Figure 2.8. The normal of the boundary is based on the edge apertures and given by $\boldsymbol{n}^b = (\delta y(A^x_e - A^x_w), \delta x(A^y_n - A^y_s))/||\delta y(A^x_e - A^x_w), \delta x(A^y_n - A^y_s)||$. Recognising l in the denominator of $\boldsymbol{n}^b$, the discrete continuity equation can be written as

$$\begin{aligned} u_e A^x_e \delta y \;+\; v_n A^y_n \delta x \;-\; u_w A^w_x \delta y \;-\; v_s A^y_s \delta x \;+ \\ u_b(A^x_e - A^x_w)\delta y \;+\; v_b(A^y_n - A^y_x)\delta x \;=\; 0. \end{aligned} \tag{2.6}$$

This equation can be applied in every computational cell, regardless the configuration of the geometry.

2.3.5 Spatial discretisation of the Navier-Stokes equations

The discretisation of the Navier-Stokes equations, Equation (2.2), is only discussed for the equation in x-direction, the other directions follow a similar approach. The control volumes in which the conservation law of momentum is applied, are chosen around velocities which are defined at cell faces. For uncut cells the control volumes consist of the left half of the cell right of the velocity and the right half of the cell left of the velocity (see Figure 2.9).

When the cells are cut by an object, the control volumes are defined by bisecting the adjacent computational cells. In Figure 2.10 two examples of control volumes in the neighbourhood of cut cells are given.

Time derivative

The time derivative term in the Navier-Stokes equation is discretised in space using the midpoint rule.

$$\int\limits_{V_f} \frac{\partial u}{\partial t} dV_f = \frac{\partial u_c}{\partial t} F^b_c \delta x_c \delta y.$$

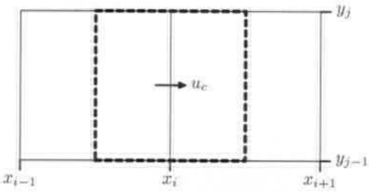


Figure 2.9: Control volume for the momentum equation in x-direction (indicated with dashed line) in case of uncut cells

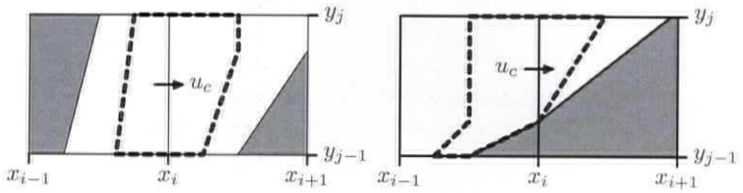


Figure 2.10: Control volumes for the momentum equation in x-direction (indicated with dashed line) in case of cut cells

Here, V_f is the control volume belonging to the horizontal central velocity u_c. The size of the control volume is computed as $F_c^b \delta x_c \delta y$ where $F_c^b = (F_w^b \delta x_w \delta y + F_e^b \delta x_e \delta y)/2$. The subscripts e and w stand for eastern and western cell respectively (see Figure 2.11 for an explanation of the notation).

Convective term

The convective term of the Navier-Stokes equation in x-direction is given by

$$\int\limits_{\partial V_f} u(\boldsymbol{u} \cdot \boldsymbol{n}_f) dS_f.$$

This term represents the advection of momentum through the boundaries of volume V_f. To distinguish between the velocity in the horizontal momentum u that is advected, and the x-component of the velocity $\boldsymbol{u}$ in the mass fluxes $\boldsymbol{u} \cdot \boldsymbol{n}_f dS_f$, the velocity in the horizontal momentum u will be replaced by the symbol ϕ in the remainder of this paragraph. First, an example of the convection in a control volume in the neighbourhood of a moving object will be considered. After that, the general formula for the convective terms will be given.

In Figure 2.11 an example of a control volume in the neighbourhood of a moving body is given. The grey area denotes the body geometry which is moving with velocity $\boldsymbol{u}_b$. The object velocity $\boldsymbol{u}_{b_l}$ holds for the complete part of the object in the left cell, whereas in the right cell the object velocity is given by $\boldsymbol{u}_{b_r}$

The momentum velocities ϕ on the different boundaries of the control volume are calculated as a simple average of the velocities at the cell faces. The simple average is

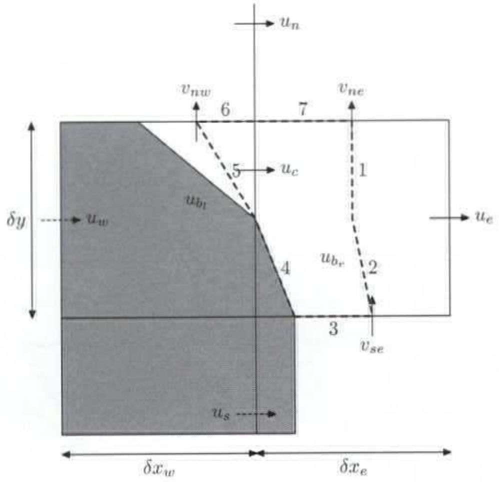


Figure 2.11: Control volume for the discretisation of convective terms

chosen every time an average is needed, because weighted averages would influence the symmetry of the discretisations [92]. Using this average, the momentum velocities on the left (ϕ_l), right (ϕ_r), upper (ϕ_u), and lower (ϕ_d) boundaries of the control volumes are given by

$$\begin{aligned} \phi_l &= \tfrac{1}{2}(\phi_w + \phi_c), \quad & \phi_r &= \tfrac{1}{2}(\phi_e + \phi_c), \\ \phi_d &= \tfrac{1}{2}(\phi_s + \phi_c), \quad & \phi_u &= \tfrac{1}{2}(\phi_n + \phi_c), \end{aligned}$$

where ϕ_c, ϕ_n, ϕ_e, ϕ_s and ϕ_w are the horizontal velocities u_c, u_n, u_e, u_s and u_w.

The convective discretisation can be written in terms of mass fluxes through the boundaries of the control volume multiplied by the velocities ϕ_r, ϕ_u, ϕ_l and ϕ_d

$$\int_{\partial V_f} \phi(\boldsymbol{u} \cdot \boldsymbol{n}_f) dS_f = m_r\phi_r - m_d\phi_d - m_l\phi_l + m_u\phi_u.$$

In the example control volume of Figure 2.11 where the boundaries are divided into seven segments, m_r consists of the sum of the horizontal mass fluxes through segments 1, 2 and 4. Mass flux m_l is formed by the horizontal mass flux through segment 5, m_u by the (vertical) mass fluxes through segments 6 and 7, and m_d by the vertical mass fluxes through segments 2, 3, 4 and 5. In general, the mass fluxes through the different boundaries are given by

$$\begin{aligned}
m_r &= \tfrac{1}{2} && (A^x_e u_e \delta y + A^x_c u_c \delta y + (A^x_c - A^x_e) u_{b_r} \delta y), \\
m_u &= \tfrac{1}{2} && (A^y_{ne} v_{ne} \delta x_e + \max(0, (A^y_{se} - A^y_{ne}) v_{b_r}) \delta x_e + A^y_{nw} v_{nw} \delta x_w + \\
& && \max(0, (A^y_{sw} - A^y_{nw}) v_{b_l}) \delta x_w), \\
m_l &= \tfrac{1}{2} && (A^x_w u_w \delta y + A^x_c u_c \delta y + (A^x_c - A^x_w) u_{b_l} \delta y), \\
m_d &= \tfrac{1}{2} && (A^y_{se} v_{se} \delta x_e + \max(0, (A^y_{ne} - A^y_{se}) v_{b_r}) \delta x_e + A^y_{sw} v_{sw} \delta x_w + \\
& && \max(0, (A^y_{nw} - A^y_{sw}) v_{b_l}) \delta x_w).
\end{aligned} \tag{2.7}$$

Here A^x_* and A^y_* are the boundary apertures belonging to the cell face where u_* and v_* are defined. The mass fluxes through the right and left boundary consist of a part from the mass flux which is an average of the mass flux through the right and left cell face (the first two terms) and a part from the moving body (the third term). The mass flux due to the moving body is positive when the body is moving out of the cell. This results for the mass flux through the right boundary m_r in the term $(A^x_c - A^x_e)u_{b_r}$ where u_{b_r} is the velocity of the body in the eastern cell.

The fluxes through the upper and lower cell contain the fluxes due to the fluid velocities, where half of the left cell and half of the right cell have been taken. The fluxes due to the moving object contain a max-function, which distinguishes between the situation that the body is moving into or out of the cell. When in the example in Figure 2.11 the body is moving downwards, the resulting mass flux in the eastern cell should be positive. In this case, $\max(0, (A^y_{se} - A^y_{ne})v_{b_r})$ is a positive number, so this will give a contribution in m_r. Then the part $\max(0, (A^y_{ne} - A^y_{se})v_{b_r})$ in m_d equals zero. So the vertical mass flux in the eastern cell due to the moving body only gives a contribution in m_u, which results in a positive contribution in Equation (2.7).

Now the discretisation of the convective terms can be written down by substituting the mass fluxes and the momentum velocities in Equation (2.7), resulting in

$$\begin{aligned}
\int\limits_{\partial V_f} \phi(\boldsymbol{u} \cdot \boldsymbol{n}_f) dS_f = m_r \phi_r - m_d \phi_d - m_l \phi_l + m_u \phi_u \quad &= \\
\tfrac{1}{2} m_r \phi_e - \tfrac{1}{2} m_d \phi_s - \tfrac{1}{2} m_l \phi_w + \tfrac{1}{2} m_u \phi_n + (\tfrac{1}{2} m_r - \tfrac{1}{2} m_d - \tfrac{1}{2} m_l + \tfrac{1}{2} m_u) \phi_c \quad &= \\
\tfrac{1}{4}(A^x_e u_e \delta y + A^x_c u_c \delta y + (A^x_c - A^x_e) u_{b_r} \delta y) \quad \phi_e \quad &+ \\
-\tfrac{1}{4}(A^y_{se} v_{se} \delta x_e + \max(0, (A^y_{ne} - A^y_{se}) v_{b_r}) \delta x_e + & \\
A^y_{sw} v_{sw} \delta x_w + \max(0, (A^y_{nw} - A^y_{sw}) v_{b_l}) \delta x_w) \quad \phi_s \quad &+ \\
-\tfrac{1}{4}(A^x_w u_w \delta y + A^x_c u_c \delta y + (A^x_c - A^x_w) u_{b_l} \delta y) \quad \phi_w \quad &+ \\
\tfrac{1}{4}(A^y_{ne} v_{ne} \delta x_e + \max(0, (A^y_{se} - A^y_{ne}) v_{b_r}) \delta x_e + & \\
A^y_{nw} v_{nw} \delta x_w + \max(0, (A^y_{sw} - A^y_{nw}) v_{b_l}) \delta x_w) \quad \phi_n \quad &+ \\
\tfrac{1}{4}(A^x_e u_e \delta y - A^x_w u_w \delta y + A^y_{ne} v_{ne} \delta x_e + A^y_{nw} v_{nw} \delta x_w - A^y_{se} v_{se} \delta x_e - A^y_{sw} v_{sw} \delta x_w + & \\
(A^x_c - A^x_e) u_{b_r} \delta y - (A^x_c - A^x_w) u_{b_l} \delta y + (A^y_{sw} - A^y_{nw}) v_{b_l} \delta x_w + (A^y_{se} - A^y_{ne}) v_{b_r} \delta x_e) \quad \phi_c . &
\end{aligned} \tag{2.8}$$

In the central coefficient (which is the coefficient in front of ϕ_c) the max-functions have disappeared. To explain that, look at the term $\max(0, (A^y_{se} - A^y_{ne})v_{b_r}) - \max(0, (A^y_{ne} -$

$A^y_{se})v_{b_r})$. If $A^y_{se} - A^y_{ne}$ is positive, the first term gives a positive contribution and the second term is zero; if $A^y_{se} - A^y_{ne}$ is negative, the first term is zero, whereas the second term gives a positive contribution (without the minus sign). So the complete term can be replaced by $(A^y_{se} - A^y_{ne})v_{b_r}$.

When examining the central coefficient further, the discrete continuity equations (Equation (2.6)) for the eastern and western cell are recognised. Since both are zero, the total central coefficient is zero. When also looking at the coefficients in front of ϕ_e, ϕ_w ,ϕ_n, and ϕ_s it can be clearly seen that the resulting matrix with the convective coefficients is skew symmetric. So the skew-symmetric property of the continuous convective operator is conserved using this discretisation. This results in favourable numerical properties, which will be used later in Section 2.3.7.

Diffusive term

The discretisation of the diffusive term of the Navier-Stokes equation

$$\frac{\mu}{\rho}\oint_{\partial V} \nabla u \cdot \boldsymbol{n} dS \tag{2.9}$$

is not so straightforward. The discretisation can not be performed independent of the exact location of the solid boundary, because derivatives of the horizontal velocity are needed at the boundary of the control volumes, which do not need to be zero. In the discretisation of the convective term only velocities at the boundaries are needed, which are zero at solid boundaries, giving rise to a discretisation independent of the exact boundary.

When writing the integrand in Equation (2.9) as $\frac{\partial u}{\partial n}$, a straightforward discretisation of the diffusive terms in case of Figure 2.11 would be

$$\frac{\mu}{\rho}\oint_{\partial V} \nabla u \cdot \boldsymbol{n} = \frac{\mu}{\rho}\sum_{k=1}^{7}\frac{u_k - u_c}{|\boldsymbol{n}_k|}\int_k dS.$$

The length of the normal on segment k can be approximated by

$$|\boldsymbol{n}_k| = \frac{V_k}{A_k}, \quad \text{with } A_k = \int_k dS,$$

where V_k is a volume corresponding to segment k of the control volume and A_k is the area of segment k. In [26] the precise choice for V_k is explained. A problem with this formulation is the division by volume V_k, which can become arbitrarily small in cut cells. This introduces instabilities in the discretisation. To prevent these instabilities another discretisation is adopted, in which the geometry is handled in a 'staircase' way. This pretty inaccurate way of discretising the diffusive term does not influence the calculations in this thesis, since these simulations are convection driven. And on the relatively coarse grids, without much stretching towards the boundary of the bodies, the boundary layer cannot be resolved. The diffusive term is rewritten to a volume integral as

$$\frac{\mu}{\rho}\oint_{\partial V} \nabla u \cdot \boldsymbol{n} dS = \frac{\mu}{\rho}\int_V \nabla \cdot \nabla u \, dV.$$

Now, the midpoint rule has been used, which is also adopted for the spatial discretisation of the time derivative. In two dimensions the integrand can be written as the sum of the second order horizontal and vertical derivatives of the horizontal velocity. Using Figure 2.12, first the first order derivatives are calculated at the boundaries of the control volume. Using the first order derivatives, the second order derivative is calculated at the position of u_c. But to prevent division by short distances, the first order derivatives are not positioned at the boundaries of the control volume, but in the center of the cells, thus treating the cells as uncut. This results in

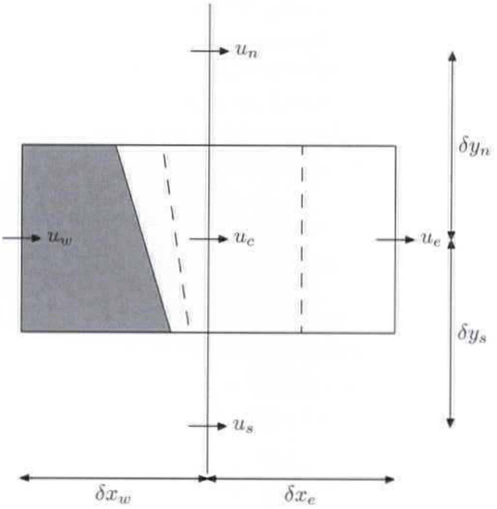


Figure 2.12: Control volume for the discretisation of diffusive terms

$$\nabla \cdot \nabla u = \frac{1}{\delta x_c}(\frac{u_e - u_c}{\delta x_e} - \frac{u_c - u_w}{\delta x_w}) + \frac{1}{\delta y_c}(\frac{u_n - u_c}{\delta y_n} - \frac{u_c - u_s}{\delta y_s}),$$

with $\delta x_c = \frac{1}{2}(\delta x_e + \delta x_w)$ and $\delta y_c = \frac{1}{2}(\delta y_n + \delta y_s)$. Now, the discretised diffusive term can be written as

$$\begin{aligned}\frac{\mu}{\rho}\oint_{\partial V} \nabla u \cdot \boldsymbol{n} &= \frac{\mu}{\rho} F_c^b \delta x_c \delta y_c \left(\frac{1}{\delta x_c \delta x_e} u_e + \frac{1}{\delta x_c \delta x_w} u_w + \frac{1}{\delta y_c \delta y_n} u_n + \frac{1}{\delta y_c \delta y_s} u_s \right. \\ &\quad \left. - (\frac{1}{\delta x_c \delta x_e} + \frac{1}{\delta x_c \delta x_w} + \frac{1}{\delta y_c \delta y_n} + \frac{1}{\delta y_c \delta y_s}) u_c \right).\end{aligned}$$

From this expression it is clear that the matrix containing the diffusive coefficients is symmetric. Furthermore, the diagonal entries (obtained by the coefficient of u_c) are negative, whereas the off-diagonal entries are positive. The sum of the off-diagonal entries equals the diagonal entry, apart from a minus sign. So the resulting discrete diffusive operator is a symmetric negative-definite matrix like the underlying difference operator [20, 92].

Upwind discretisation for convection: artificial viscosity

In the above derivation of the discretisation of the convective term a central discretisation is used. To prevent wiggles that can occur in a central discretisation an upwind discretisation is adopted. The upwind discretisation can be seen as a central discretisation with an extra diffusion term that increases the diffusion with an amount of $uh/2$ with h the mesh size. In the implementation of the upwind discretisation this interpretation is used, so the diffusion coefficients have been altered by increasing the viscosity coefficient μ/ρ. For example, when looking at the convective term in the x-direction, the convective coefficients in front of the eastern and western cell velocities are given by (using a central discretisation, see Equation (2.8))

$$\begin{aligned} c_e &= \tfrac{1}{4}(A^x_e u_e + A^x_c u_c + (A^x_c - A^x_e)u_{b_r})\delta y, \\ c_w &= -\tfrac{1}{4}(A^x_w u_w + A^x_c u_c + (A^x_c - A^x_w)u_{b_l})\delta y. \end{aligned}$$

In case of a positive horizontal velocity no contribution from the eastern velocity is present in the upwind discretisation. So this contribution has to be cancelled by the extra term in the diffusion coefficient. In general, the diffusion coefficients for the western and eastern cell velocity with the artificial viscosity term are given by

$$\begin{aligned} d_e &= \frac{\mu}{\rho}F^b_c \delta x_c \delta y_c \frac{1}{\delta x_c \delta x_e} + \tfrac{1}{4}|A^x_e u_e + A^x_c u_c + (A^x_c - A^x_e)u_{b_r}|\delta y, \\ d_w &= \frac{\mu}{\rho}F^b_c \delta x_c \delta y_c \frac{1}{\delta x_c \delta x_w} + \tfrac{1}{4}|A^x_w u_w + A^x_c u_c + (A^x_c - A^x_w)u_{b_l}|\delta y, \end{aligned}$$

where the first part originates from Equation (2.10). Clearly, the second part of the eastern diffusive coefficient d_e cancels the eastern convective coefficient c_e when the horizontal velocity is positive, since c_e and d_e appear in front of the eastern velocity in the momentum equation as $(-c_e + d_e)u_e$.

The extra contribution in the diffusive terms due to the upwind discretisation is always positive, resulting in dissipation of energy. In the problems studied in this thesis the artificial viscosity is much larger than the kinematic viscosity μ/ρ. In the convection and gravity dominated simulations studied in this chapter, the extra energy dissipation is not important. However, when studying wave simulations in a large domain for a simulation time of many periods, the artificial viscosity induces unphysical wave damping as shown in Section 3.6.1.

Pressure term

The discretisation of the pressure term in x-direction given by

$$\oint_{\partial V} p n_x dS,$$

is done in the same manner as the convective terms: the integrand is evaluated over the different segments of the control volume. Using the situation in Figure 2.13, where p_e is the pressure in the eastern cell and p_w the pressure in the western cell (the pressure has

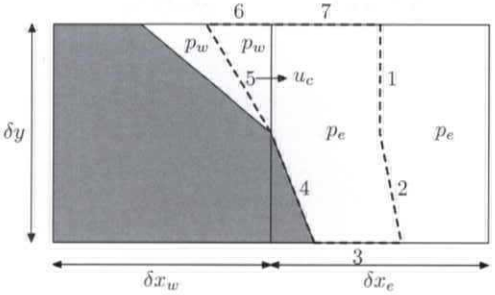


Figure 2.13: Control volume for discretisation of the pressure term

a constant value throughout the cell), the pressure term is written as

$$\oint_{\partial V} pn_x dS = \sum_{k=1}^{7} p_k \int_k n_x dS,$$

where the summation is over the seven segments of the control volume with n_x the first component of the normal vector and p_k the pressure along segment k. For segments 1, 2, 3, 4, and 7 the pressure is equal to p_e and for segments 5 and 6 the pressure is equal to p_w. Since n_x is zero along segments 3, 6, and 7, these segments do not contribute to the sum. Then, for the other segments

$$\int_k n_x dS = s_k \int_k \cos \alpha_k dS,$$

where α_k is the angle between segment k and the vertical ($0 \leq \alpha_k < \frac{\pi}{2}$) and $s_k = 1$ for segment 1 and 2 and $s_k = -1$ for segment 4 and 5. The integral in the right-hand side of this equation is equal to the vertical length of segment k. Then, for the example in Figure 2.13 the pressure term is discretised as

$$\oint_{\partial V} pn_x dS = p_e A_c^x \delta y + p_e(1 - A_c^x)\delta y - p_e(1 - A_c^x)\delta y - p_w A_c^x \delta y.$$

The contributions of segments 2 and 4 cancel each other, since these segments lie in the same computational cell. This results in a general formulation that is independent of the exact location of the solid boundary

$$\oint_{\partial V} pn_x dS = (p_e - p_w) A_c^x \delta y. \tag{2.10}$$

When examining the coefficients that appear in the pressure matrix in the discrete momentum equation, it is seen that the same geometrical information is present as in the discrete divergence matrix. In fact, the coefficient A_c^x of p_e also appears in the continuity equation of the left cell and $-A_c^x$ of p_w in the right cell's continuity equation. This shows that the analytic property $\nabla = -(\nabla \cdot)^{\mathrm{T}}$ is also present in the properties of the matrices for the discrete divergence and gradient operators.

External force: gravity

Since gravity is the only external force present in this study, the discretisation of external forces will only be shown for that case given by

$$\int_{V_f} F_z dV_f. \tag{2.11}$$

When all velocities are equal to zero and gravity is the only external force, the pressure term should cancel the gravity term since all other terms are equal to zero, and should result in a hydrostatic pressure field. Therefore, the discretisation of the gravity term has to be similar to the discretisation of the pressure term. When using Gauss' divergence theorem for the vector $(0, 0, -gz)$ Equation (2.11) can be written as a boundary integral

$$\int_{V_f} F_z dV = \int_{V_f} \nabla \cdot \begin{pmatrix} 0 \\ 0 \\ -gz \end{pmatrix} dV_f = \oint_{\partial V_f} \begin{pmatrix} 0 \\ 0 \\ -gz \end{pmatrix} \cdot \boldsymbol{n} dS_f = \oint_{\partial V_f} -gz\ n_z\ dS_f.$$

Evaluating this boundary integral of the hydrostatic pressure potential in the xz-plane leads to the correct discretisation for the gravity force given by

$$-\oint_{\partial V_f} gz\ n_z\ dS_f = -A_c^z \delta x (gz_n - gz_s) = -A_c^z \delta x g \delta z,$$

where z_n and z_s are the coordinates of the northern and southern cell centers respectively.

2.3.6 Temporal discretisation and solution method

After the continuity and Navier-Stokes equations have been discretised in space, they can be written as

$$M^0 \boldsymbol{u}_h = -M^b \boldsymbol{u}_b, \tag{2.12}$$

$$\Omega \frac{\partial \boldsymbol{u}_h}{\partial t} = -C(\boldsymbol{u}_h, \boldsymbol{u}_b)\boldsymbol{u}_h + \frac{1}{\rho}(D\boldsymbol{u}_h - G\boldsymbol{p}_h) + \boldsymbol{F}_h. \tag{2.13}$$

Here, $\boldsymbol{u}_h$ is the vector containing all discrete velocities and $\boldsymbol{p}_h$ contains all discrete pressures. In the continuity equation the discrete divergence operator has been split into a contribution M^0 working on the interior velocities $\boldsymbol{u}_h$ and a contribution M^b working on boundary velocities $\boldsymbol{u}_b$. Then the property that the discrete divergence equals minus the transpose of the discrete gradient can be written as

$$M^0 = -G^{\mathrm{T}}$$

since the discrete gradient operator only works on pressures inside the fluid.

In the spatially discretised momentum equation, Equation (2.13), which consists of three equations for every cell (in three dimensions), the matrix Ω is a diagonal matrix containing the momentum control volumes. The matrix C contains the convective coefficients, which are dependent on the interior and boundary velocities, and is skew symmetric. Matrix D contains the diffusive coefficients and is symmetric and negative definite. The vector $\boldsymbol{F}_h$ contains the gravitational force.

For the time discretisation the explicit forward Euler method is adopted. Using a superscript to indicate the time level this results in

$$M^0 \boldsymbol{u}_h^{n+1} = -M^b \boldsymbol{u}_b^{n+1}, \tag{2.14}$$

$$\Omega \frac{\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n}{\delta t} = -C(\boldsymbol{u}_h^n, \boldsymbol{u}_b)\boldsymbol{u}_h^n + \frac{1}{\rho}(\mu D \boldsymbol{u}_h^n + (M^0)^{\mathrm{T}} \boldsymbol{p}_h^{n+1}) + \boldsymbol{F}_h^n. \tag{2.15}$$

The continuity equation is discretised at the new time level to ensure a divergence free velocity field at this time level.

To solve this system of equations, the terms in Equation (2.15) are rearranged to

$$\boldsymbol{u}_h^{n+1} = \tilde{\boldsymbol{u}}_h^n - \delta t \Omega^{-1} \frac{1}{\rho}(M^0)^{\mathrm{T}} \boldsymbol{p}_h^{n+1}, \tag{2.16}$$

where

$$\tilde{\boldsymbol{u}}_h^n = \boldsymbol{u}_h^n - \delta t \Omega^{-1}(C(\boldsymbol{u}_h^n, \boldsymbol{u}_h^b)\boldsymbol{u}_h^n - \frac{\mu}{\rho} D \boldsymbol{u}_h^n - \boldsymbol{F}_h^n). \tag{2.17}$$

First, an auxiliary vector field $\tilde{\boldsymbol{u}}_h^n$ is calculated using Equation (2.17). Next, Equation (2.16) is substituted in Equation (2.14), which results in

$$M^0(\tilde{\boldsymbol{u}}_h^n - \delta t \Omega^{-1} \frac{1}{\rho}(M^0)^{\mathrm{T}} \boldsymbol{p}_h^{n+1}) = -M^b \boldsymbol{u}_b^{n+1}.$$

This can be rewritten to

$$M^0 \Omega^{-1} (M^0)^{\mathrm{T}} \boldsymbol{p}_h^{n+1} = \frac{\rho}{\delta t}(M^0 \tilde{\boldsymbol{u}}_h^n + M^b \boldsymbol{u}_b^{n+1}), \tag{2.18}$$

where a Poisson equation for the pressure is recognised. From this equation the pressure is solved using the SOR (Successive Over Relaxation) method where the optimal relaxation parameter is determined during the iterations [5]. Once the pressure field is known, the new velocity field is calculated from $\tilde{\boldsymbol{u}}_h^n$ using the pressure gradient.

2.3.7 Stability

In this section, the stability of the adopted method is investigated. Thereto, first the spatial discretisation is examined by evaluating the evolution of kinetic energy. The temporal discretisation induces limits on the time step for the convective as well as the diffusive terms.

Evolution of kinetic energy

For the stability of the spatial discretisation, the evolution of kinetic energy is interesting to examine. Using the properties of the spatial discretisation matrices derived above, it can be shown that in the absence of moving objects and external forces, the energy is only dissipated due to diffusion. To see that, the kinetic energy is discretised in space as

$$\tfrac{1}{2} \int_V \rho |\boldsymbol{u}^2| dV = \tfrac{1}{2} \langle \boldsymbol{u}_h, \rho \Omega \boldsymbol{u}_h \rangle \equiv E_h,$$

where $\langle\cdot,\cdot\rangle$ denotes the standard inner product for a finite-dimensional vector space. The evolution of the discrete kinetic energy is then given by

$$\begin{aligned}\frac{dE_h}{dt} &= \tfrac{1}{2}\langle \tfrac{d}{dt}\boldsymbol{u}_h, \rho\Omega\boldsymbol{u}_h\rangle + \tfrac{1}{2}\langle \boldsymbol{u}, \tfrac{d}{dt}\rho\Omega\boldsymbol{u}_h\rangle \\ &= \tfrac{1}{2}\langle \rho\Omega\tfrac{d\boldsymbol{u}_h}{dt}, \boldsymbol{u}_h\rangle + \tfrac{1}{2}\langle \boldsymbol{u}_h, \rho\Omega\tfrac{d\boldsymbol{u}_h}{dt}\rangle.\end{aligned}$$

Using the discretised momentum equation as given in Equation (2.13) and omitting the external force, the last expression can be written as

$$\begin{aligned}\frac{dE_h}{dt} &= \frac{1}{2}((\langle -\rho C\boldsymbol{u}_h, \boldsymbol{u}_h\rangle + \langle D\boldsymbol{u}_h, \boldsymbol{u}_h\rangle + \langle -G\boldsymbol{p}_h, \boldsymbol{u}_h\rangle) + \\ &\quad \frac{1}{2}((\langle \boldsymbol{u}_h, -\rho C\boldsymbol{u}_h,\rangle + \langle \boldsymbol{u}_h, D\boldsymbol{u}_h\rangle + \langle \boldsymbol{u}_h, -G\boldsymbol{p}_h\rangle \\ &= \frac{1}{2}((\langle -\rho(C + C^{\mathrm{T}})\boldsymbol{u}_h, \boldsymbol{u}_h\rangle + \langle (D + D^{\mathrm{T}})\boldsymbol{u}_h, \boldsymbol{u}_h\rangle) - \langle G\boldsymbol{p}_h, \boldsymbol{u}_h\rangle.\end{aligned}$$

Since the diffusive matrix is a symmetric matrix ($D = D^{\mathrm{T}}$), the convective matrix is skew symmetric ($C + C^{\mathrm{T}} = 0$), and $G^{\mathrm{T}} = -M^0$, the last expression can be written as

$$\frac{dE_h}{dt} = \langle D\boldsymbol{u}_h, \boldsymbol{u}_h\rangle + \langle \boldsymbol{p}_h, M^0\boldsymbol{u}_h\rangle.$$

In the absence of moving objects ($\boldsymbol{u}_b = 0$), it follows from Equation (2.12) that $M^0\boldsymbol{u}_h = 0$. So the kinetic energy is only dissipated, since the discrete diffusive matrix D is negative definite.

In the case that moving objects are present, the kinetic energy is affected by the pressure, since from Equation (2.12) it follows that $\langle \boldsymbol{p}_h, M^0\boldsymbol{u}_h\rangle = \langle \boldsymbol{p}_h, -M^b\boldsymbol{u}_b\rangle$. When also the gravity force is taken into account, the evolution of the kinetic energy is given by

$$\frac{dE_h}{dt} = \langle D\boldsymbol{u}_h, \boldsymbol{u}_h\rangle + \langle \boldsymbol{p}_h, M^0\boldsymbol{u}_h\rangle + \langle \rho\boldsymbol{F}_h, \boldsymbol{u}_h\rangle$$

Since the discretisation of the external force due to gravity is performed consistent with the pressure discretisation, the external force contribution can be written as

$$\langle \rho\boldsymbol{F}_h, \boldsymbol{u}_h\rangle = \langle -G\rho gz, \boldsymbol{u}_h\rangle = \langle (M^0)^{\mathrm{T}}\rho gz, \boldsymbol{u}_h\rangle = \langle \rho gz, M^0\boldsymbol{u}_h\rangle.$$

Thus, in the presence of gravity and moving objects, the energy evolution is given by

$$\frac{dE_h}{dt} = \langle D\boldsymbol{u}_h, \boldsymbol{u}_h\rangle + \langle \boldsymbol{p}_h + \rho gz, -M^b\boldsymbol{u}_b\rangle.$$

The first term only gives dissipation of energy, since D is a negative definite matrix. Further change of energy due to pressure is completely controlled by the motion of the object. So the spatial discretisation will not cause uncontrolled energy increase.

Convection

In the case of uncut cells with fixed objects the stability of the equation containing the time integration term and the convective term is given by the CFL-restriction [16], which in one dimension reads $\delta t|u|/h \leq 1$ (h is the size of the uncut cell). When cut cells are present, for the chosen convective discretisation this criterion is not changed. This result is not directly straightforward when looking at the equation containing the time derivative and the convective term

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\Omega^{-1}C(\boldsymbol{u}, \boldsymbol{u}_b)\boldsymbol{u}. \tag{2.19}$$

The matrix Ω is a diagonal matrix containing volumes of the cells, so these entries can become arbitrarily small for cut cells, hence the elements in Ω^{-1} can become arbitrarily large. To examine stability, the eigenvalues of the convective matrix C, generated by Equation (2.8), have to be determined. These eigenvalues can be estimated as being of order $O(\Omega u/h)$. The Ω in this estimation cancels the contribution of Ω^{-1} in $\Omega^{-1}C$, leaving the stability criterion for cut cells the same as for uncut cells [23]. The proof of this result (described in [19]) uses Gerschgorin circles and is not straightforward. To get a feeling that the uncut CFL-criterion indeed is enough for stability, consider the cell shown in Figure 2.14 that is partly blocked by fixed geometry. The CFL-number belonging to the horizontal velocity $CFL_u = u_c\delta t/h_x$ has a division by h_x, which is a small number. To see that the order of the CFL-number is in the uncut cell size, estimate the horizontal velocity u_c in terms of the vertical velocities using the continuity equation for this cell:

$$h_y u_c + v_n h_x - v_s h_x = 0 \Rightarrow u_c = \frac{(v_s - v_n)h_x}{h_y}$$

This is inserted into the CFL-number resulting in

$$CFL_u = \frac{u_c \delta t}{h_x} \approx \frac{(v_s - v_n)h_x \delta t}{h_y h_x} = \frac{(v_s - v_n)\delta t}{h_y},$$

where the division by h_x has disappeared. So, in the case of Figure 2.14, the CFL-number can be approximated by a number that is only dependent on an uncut cell size.

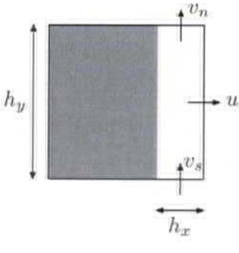


Figure 2.14: The stability criterion for a cut cell is the same as for an uncut cell

When moving objects are present, the story becomes somewhat different. A distinction can be made between the object moving normal to its boundary and tangential to its boundary (as shown in the left and right of Figure 2.15 respectively). When the object is moving tangential to its boundary, the eigenvalues of the matrix $C(\boldsymbol{u}, \boldsymbol{u}_b)$ can again be estimated by $O(\Omega u/h)$, which means that stability is guaranteed when the CFL-restriction is used. But when the object is moving normal to itself, the eigenvalues of $C(\boldsymbol{u}, \boldsymbol{u}_b)$ are $O(hu_b)$. Now stability is not guaranteed anymore, with eigenvalues of $\Omega^{-1}C(\boldsymbol{u}, \boldsymbol{u}_b)$ in Equation (2.19) of order $O(\Omega^{-1}hu_b)$. They can become arbitrarily large due to the factor Ω^{-1}.

To cancel the effect of Ω^{-1}, a formulation based on a weighted average of the fluid velocity and the boundary velocity is applied in the cells cut by the moving object. To avoid smearing of the interface in cases where it is not necessary to stabilise the convective term (namely when the object is moving tangentially to its boundary), the following discretisation is used

$$\boldsymbol{u}^{n+1} = \Lambda(\boldsymbol{u}^n - \delta t(\Omega^{n+1})^{-1}C^n\boldsymbol{u}^n) + (I - \Lambda)\boldsymbol{u}_b^{n+1}, \tag{2.20}$$

with weight factor $\Lambda = \Omega^{n+1}(\Omega^{n+1} + |\Delta\Omega|)^{-1}$, where $\Delta\Omega = \Omega^{n+1} - \Omega^n$ is the difference between cell volumes at two different time steps. The weight factor Λ has been chosen such that the stabilising term is only used when the body is moving; note that it equals unity for fixed objects. Furthermore, maximum stabilisation is established when the object is moving normal to its boundary, whereas no stabilisation is used when the object is moving tangential to its boundary (as $\Delta\Omega = 0$ then).

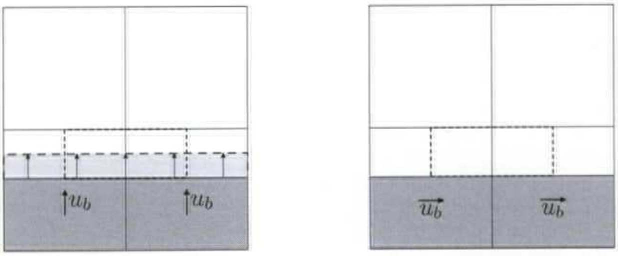


Figure 2.15: Left: boundary moving normal to itself: maximum stabilisation is required; right: boundary moving tangential to itself: no stabilisation is required

Diffusion

Also from the diffusive term, a stability criterion follows with a restriction on the time step. In the case of uncut cells, this criterion is given by $\delta t \leq h^2/2\nu$, where ν denotes kinematic viscosity. Since the diffusive term is discretised as if all cells were uncut ('staircase' approach) as explained before, the above criterion is also valid in the cut-cell model. In most cases studied in this thesis, the diffusive time step limit is much less restrictive than the limit on the time step following from the CFL-criterion.

2.3.8 Solid wall boundary conditions

At solid walls and inside fixed or moving objects boundary conditions for the velocity are needed. Therefore, the no-slip boundary condition $\boldsymbol{u} = \boldsymbol{u}_b$ is used. Velocities between two B-cells and between a B- and F-cell are set equal to the velocity of the object. So in fixed objects and walls $\boldsymbol{u} = 0$ is adopted, so the BB- and FB-velocities are set equal to zero.

2.3.9 Pressure at the free surface

A boundary condition for the pressure is needed in surface cells. The pressure in surface cells can be calculated from interpolation between the pressure at the free surface and the pressure in an adjacent fluid cell. The boundary condition that defines the pressure at the free surface is given by Equation (2.3), which describes the continuity of normal stresses at the free surface. The term containing the viscosity is neglected, which leaves

$$p = p_0 - \sigma\kappa, \tag{2.21}$$

with p_0 the atmospheric pressure, σ the surface tension and κ the total curvature of the free surface. Although surface tension is not the driving force in the simulations studied in this thesis, it can not always be neglected. To calculate the contribution of the surface tension in the pressure at the free surface, the total curvature of the free surface has to be determined in every S-cell.

If the free surface is given by a level-set function $s(x, y, t) = 0$, the total curvature is given by $\kappa = \nabla \cdot \boldsymbol{n}$, where $\boldsymbol{n} = \nabla s / |\nabla s|$ is the normal at the free surface. In this study, a local height function is introduced to calculate the total curvature. The local height function is defined based on the orientation of the free surface. If the orientation of the free surface is more vertical than horizontal, the local height function is defined parallel with the x-axis, otherwise it is defined parallel to the y-axis (in 2D). The values of the local height function are calculated using VOF-fractions of three adjacent computational cells (see Figure 2.31 for a visual explanation). If for example the orientation of the free surface is vertical as in Figure 2.16, the local height function is defined by $h(y, t) = x$. In terms of the level set function this corresponds to $s(x, y, t) = x - h(y, t)$. In this case the curvature of the free surface is given by

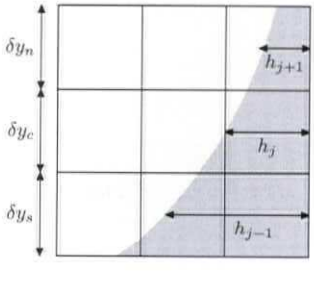


Figure 2.16: Local height function to calculate the total curvature κ

$$\kappa = \frac{\partial}{\partial y}\left(\frac{\partial h/\partial y}{\sqrt{1 + (\partial h/\partial y)^2}}\right).$$

The derivative of the local height function needed at positions $y_{j+\frac{1}{2}}$ and $y_{j-\frac{1}{2}}$ are discretised using standard finite differences. Then the discrete curvature in the centre S-cell of Figure 2.16 is given by

$$\kappa = \frac{1}{\delta y_c}\left(\frac{h'_n}{\sqrt{1 + h'^2_n}} - \frac{h'_s}{\sqrt{1 + h'^2_s}}\right),$$

with

$$h'_n = \frac{h_{j+1} - h_j}{\frac{1}{2}(\delta y_n + \delta y_c)} \qquad \text{and} \qquad h'_s = \frac{h_j - h_{j-1}}{\frac{1}{2}(\delta y_c + \delta y_s)}.$$

In three dimensions the procedure for the calculation of the curvature is a bit more complicated, but follows the same approach. Details can be found in [26].

At the intersection of the free surface and the solid body a boundary condition is needed for computing the total curvature. This is given by a static contact angle, which is the angle between the normal of the free surface and the normal of the solid body. The discretisation of the contact angle will not be explained in this thesis, but is described in [26].

Once the total curvature has been calculated, the pressure at the free surface p_{fs} is determined using Equation (2.21). The pressure in surface cells p_S is now calculated from the pressure at the free surface and the pressure in an adjacent fluid cell p_F [44]. The fluid cell that is used for the interpolation is chosen based on the orientation of the free surface. If for example the orientation of the free surface is mainly horizontal (as in Figure 2.17) with the fluid below the free surface, the fluid cell below the surface cell is used for the interpolation. The pressure in the centre of the surface cell is calculated by linear interpolation as

$$p_S = \eta p_{fs} + (1 - \eta) p_F$$

using the notation in Figure 2.17 and $\eta = h/d$. If no fluid cell is found as neighbour of the surface cell, the pressure in the surface cell is set equal to the atmospheric pressure, corrected with a hydrostatic pressure contribution based on the local height of the fluid.

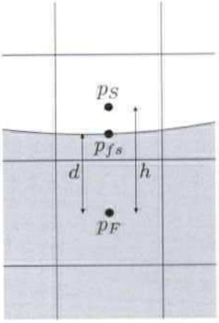
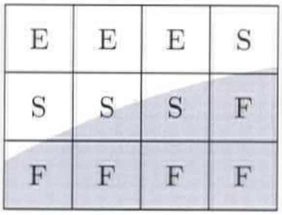


Figure 2.17: Pressure interpolation in surface cells

2.3.10 Velocities at the free surface: SE-velocities

Velocities in the neighbourhood of the free surface can be grouped in different classes, based on the velocity labels (see Figure 2.18). Recalling that the velocities are defined at the cell faces, there are 5 different labels of the velocities in the neighbourhood of the free surface (without the presence of solid boundaries): FF, FS, SS, SE and EE-velocities. The first class of velocities that all have the same numerical treatment consists of the velocities between two F-cells, between two S-cells and between an S- and F-cell. These velocities are determined by solving the momentum equation and are called momentum velocities. The second class consists of the velocities between an S and an E-cell. The choice for these SE-velocities has a very large impact on the robustness and accuracy of the method. The last class consists of velocities between two E-cells that are sometimes needed to solve the momentum equation. These are determined using the tangential free surface condition. In this section, the SE and EE-velocity boundary conditions are discussed.



FF, FS, SS: momentum equation

SE: extrapolation

EE: tangential free surface condition

Figure 2.18: Different classes of velocities near the free surface

For the calculation of momentum velocities, velocities at the faces between surface and empty cells are needed. In the original Marker and Cell method [39], mass conservation is demanded in surface cells to determine the SE-velocity or velocities in the surface cell. This method has been widely used since. In [14] a thorough description of the free surface boundary conditions is given. In that paper also conservation of mass in surface cells is demanded, where obstacle cells are taken into account. In [13], the SUMMAC method is described, based on the MAC-method. It turned out that for simulation of waves the conservation-of-mass method to determine SE-velocities is not very accurate. An extrapolation of the velocity field has been used instead.

In the method described in this thesis, the advantages and disadvantages of both mass conservation and extrapolation have been combined in a new method. In this section, first the mass conservation method and the extrapolation method are described, after which the choice for the adopted new method is motivated. Some results are shown to illustrate the behaviour of the different methods.

Method 1: mass conservation in S-cells

The first method to determine velocities on faces between surface and empty cells is to demand mass conservation in a surface cell. This means that the total flux through the boundaries of a surface cell equals zero. Thereto, apertures have to be taken into account, as in the discretised continuity equation, Equation (2.6). In two dimensions, three different configurations of empty cells around a surface cell can be distinguished, as shown in Figure 2.19. In the left configuration, only one SE-velocity is present around the central S-cell. Conservation of mass can be applied immediately, where the SE-velocity follows from the other three velocities, taking into account the geometry apertures. In the second configuration, two SE-velocities are present that are not positioned opposite each other. In this case the net-flux through both SS-faces is divided over the two SE-faces. To divide this flux over the two SE-faces, first the SE-velocities are set equal to

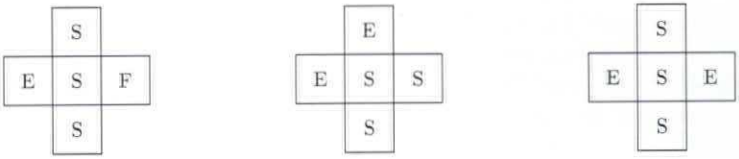


Figure 2.19: Three configurations of SE-velocities at the faces of a surface cell

the opposite SS-velocity, then the rest-flux is equally divided over the two SE-velocities. This way, the characteristics of the main direction of the flow are taken into account (in the first step), as well as the mass conservation in the S-cell (in the second step). The central S-cell in the right configuration also has two E-cell neighbours, but positioned opposite each other in this case. Now, the SE-velocities are determined by dividing the net flux through the SS-faces equally over both SE-faces.

There are two disadvantages of this method. First, instabilities can occur in the case of small cut cells which can even result in divergence of the computation. For example, in the configuration shown in Figure 2.20, the SE-velocity of the lower central S-cell is getting very large when applying conservation of mass in this cell. The SE-velocity is calculated using the flux over the other faces as

$$u_{SE} = \frac{1}{A^x_{SE}}(A^z_{SS}w_{SS} - A^x_{FS}u_{FS} - 0). \tag{2.22}$$

Because of division through the small aperture A^x_{SE}, the resulting SE-velocity gets very large. If this configuration stays the same for several time steps, the velocity is enlarged every time step with the factor $1/A^x_{SE}$. Due to the CFL-criterion for stability this increase in velocity results in a decrease of the time step, until infinitely small time steps. At the end, this causes the simulation to break down. Another disadvantage of this method is the observed inaccuracy in wave simulations. This will be explained in the next chapter in Section 3.4

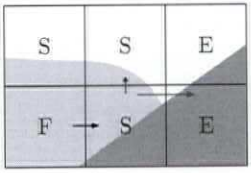


Figure 2.20: Configuration with small SE-face resulting in large SE-velocity; the dark-grey area is solid wall, the light-grey area fluid

Method 2: extrapolation from interior velocity field

To prevent the instabilities and inaccuracy in the first method described above, the principle of demanding mass conservation in surface cells is dropped. This does not violate mass conservation in the whole domain because in surface cells, which are only partly filled with fluid, no mass conservation is present. Instead it is proposed to determine the SE-velocities using extrapolation from the interior velocity field. Therefore, for every surface cell the direction in which the main body of the fluid is positioned is determined. Then, the SE-velocities of the surface cell are extrapolated from this direction. The direction of the main body of the fluid is found by examining the VOF-values of a 3 × 3 block of cells (in three dimensions a 3 × 3 × 3 block) as explained in Section 2.4.3. In Figure 2.21, an example is shown where the main direction of the fluid is in the negative z-direction (as is the case in most simulations with gravity). Then the horizontal SE-velocity u_{SE} is found by extrapolation using (at least) u_{FS}, and the vertical SE-velocities v_{SE_1} and v_{SE_2} are extrapolated using (at least) v_{FS_1} and v_{FS_2} respectively.

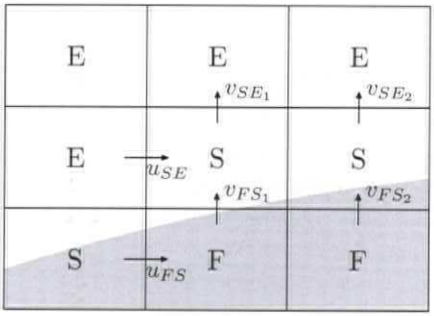


Figure 2.21: Determination of SE-velocities using extrapolation

The extrapolation can be done with different degrees of accuracy. In the current method a constant or linear extrapolation is used. In case of a wave that is propagating without disturbances linear extrapolation gives the best estimate for SE-velocities (as will be shown in Section 3.4). A disadvantage of linear extrapolation is that it can lead to instabilities when the velocity field is not smooth. In Figure 2.22 part of a velocity field is shown, where the SF and FF-velocities have opposite signs. When the SE-velocity is now calculated using linear extrapolation

$$u_{SE} = 2u_{FS} - u_{FF}$$

the velocity becomes unphysically large with an amplification factor of approximately three. If this configuration stays the same for several consecutive time steps, the velocity will 'blow up', leading to very small time steps due to the CFL-criterion for stability. In the end, this can cause the simulation to break down. To avoid this, constant extrapolation, where the SE-velocity is just copied from its lower neighbour $u_{SE} = u_{FS}$, should be chosen. Another disadvantage of the extrapolation method is the occurrence of numerical spikes in the pressure time series, as explained in the next paragraph.

Influence of SE-velocities on numerical spikes in the pressure

When looking at the pressure or force time series resulting from computations, often pressure spikes can be observed. An example is shown in the left of Figure 2.25, where

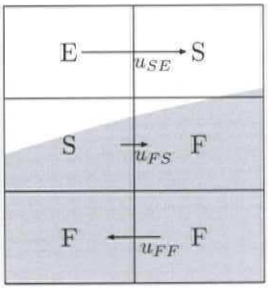


Figure 2.22: Very large SE-velocity due to linear extrapolation using u_{FS} and u_{FF}

the force signal of water smashing against a structure is plotted. In this simulation green water flow on the deck of a vessel is modelled by releasing a dam of water around the bow of the vessel. The vessel is fixed, and the water flows over the deck due to gravity. In Figure 2.23 a snapshot is shown of this simulation, three seconds after the water dam has been released. On the deck of the vessel a cylinder is placed, at which forces are calculated during the simulation. This force signal is shown in Figure 2.25.

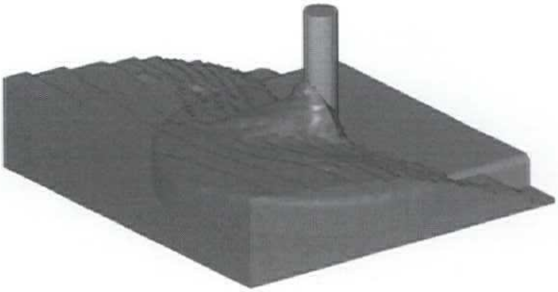


Figure 2.23: Snapshot of a simulation of green water flow on the deck of a fixed vessel

In this fixed-object case, the pressure spikes are originating from the free surface boundary conditions, especially the SE-velocities. The spikes occur because of changing cell labels: a surface cell changes to a fluid cell or an empty cell changes to a fluid cell. In Figure 2.24 an example of a configuration is shown, where a surface cell changes to a fluid cell, because the fluid has flowed to the right. The spike occurs when the velocity field is not divergence free in the S-cell, which means that the SE-velocities have not been chosen such that conservation of mass is applied in the S-cell. When such an S-cell becomes an F-cell, the pressure has to respond actively to create a divergence free velocity field in the F-cell. This response of the pressure is visible in a (numerical) pressure spike in this F-cell. As pressure pulses are travelling with infinite speed because of the incompressibility of the fluid, the pressure spike is noticed throughout the computational domain. So, these spikes will also be visible in the force, which is an integral of the pressure over the object surface.

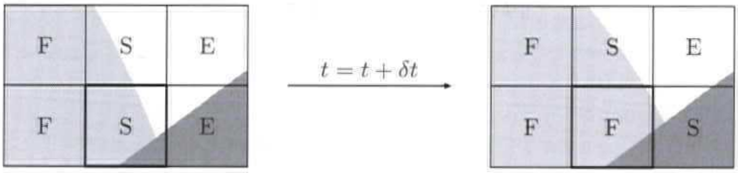


Figure 2.24: Label of the lower S-cell changes to an F-cell

The problem of pressure spikes due to changing labels is not present when using method 1 (conservation of mass) for SE-velocities. But the price to be paid when using this method, instabilities resulting in divergence of the simulations and inaccuracies in wave simulations, is very high. Nevertheless, the only way to suppress the pressure spikes is to demand conservation of mass in S-cells. When using method 2 (extrapolation) for SE-velocities, no conservation of mass is applied in S-cells, resulting in pressure spikes. This method has been used in the left of Figure 2.25, where the spikes in the pressure are clearly visible.

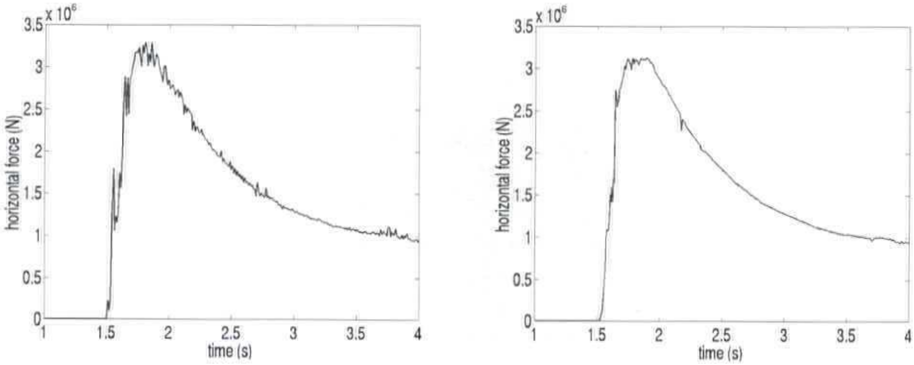


Figure 2.25: Horizontal force on the cylinder on the deck of a fixed vessel without (left) and with (right) changing former SE-velocities such that mass conservation is applied in the former S-cell

To avoid the pressure spikes without demanding conservation of mass in every S-cell, the following method is used. When the free surface velocities are calculated, all SE-velocities are determined using extrapolation. Then, at the start of the next time step, situations are identified where a surface cell has changed to a fluid cell at the end of the previous time step. Then, the velocities that were SE-velocity in the previous time step (they have changed to FS-velocities at the end of the time step) are changed, such that conservation of mass is applied in the former S-cell. Sometimes, also an E-cell changes to an F-cell in one time step. This can happen in the neighbourhood of the object, since the CFL-criterion is based on the uncut mesh size, implying that fluid can maximally travel one (uncut) cell in one time step. In those cases, the former E-cell will be made divergence free. Due to the change of the former SE-velocities at the start of the time step, the pressure sees an F-cell that already had conservation of mass in the previous time step. This results in a much smoother pressure time series. This method has been adopted in the simulation shown in the right of Figure 2.25: SE-velocities are calculated using extrapolation, but when an S- or E-cell changes to an F-cell, mass conservation is restored in that former S- or E-cell.

Unfortunately, conservation of mass can not be applied afterwards in all configurations. When, for example, an empty cell is completely surrounded by S- and F-cells and fluid enters the empty cell (as in Figure 2.26), the E-cell cannot be made divergence free afterwards. Since making the surrounding S-cells divergence free fixes all the velocities at the faces of the E-cell. This is for example the case in a simulation of a dambreak, from which pressure time series are shown in Figure 2.58. The large spikes at about 1.3 seconds are due to an E-cell that changes to an F-cell and was already surrounded completely by S- and F-cells.

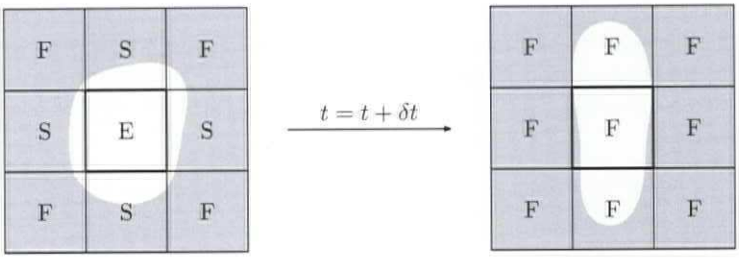


Figure 2.26: E-cell changes to an F-cell; mass conservation can not be applied to the E-cell, resulting in a pressure spike

Using the idea of mass conservation to calculate SE-velocities when an S-cell changes to an F-cell, still a very large SE-velocity can be introduced due to small edge apertures (as in Figure 2.20). Fortunately, this very large velocity will only be present in the time step when the repair has been established, but it can become so large that it still initiates an instability. To prevent the occurrence of these very large velocities a limiter has been introduced. There are two options for such a limiter: limit the size of the velocity or limit the size of the aperture such that the apertures will not become too small. Both options are a bit arbitrary, some threshold for the limiter should be chosen. To be on the safe side, thus to prevent instabilities that could lead to divergence of the method which is much worse than some pressure spikes, both limiters are used in the determination of SE-velocities. The limiter for the apertures is global, all apertures smaller than a certain threshold, in the order of 0.01, are closed. The error introduced by this limiter is very small and does not increase in time, since the geometry is moved every time step from its original position at time $t = 0$.

Adopted method for SE-velocities

To recapitulate the above discussion about SE-velocities, in this section a summary will be given of the chosen method. In principle method 2 is used, which means that SE-velocities are extrapolated from the interior velocity field. Therefore, first the direction of the main body of the fluid is determined, after which the velocities are extrapolated from that direction. At the start of a new time step it is checked if there are F-cells that were E- or S-cell in the previous time step. In that case, the former SE-velocities, as calculated at the end of the previous time step, are changed such that mass conservation holds in the former S- or E-cell. This procedure has been added to prevent pressure spikes. In Section 3.4 some more remarks are made about using linear or constant extrapolation and the influence of this choice on the accuracy of wave simulations.

Finally, for robustness, one more configuration of S- and E-cells should be noted that could lead to instabilities. This concerns a droplet that sticks to a sloping wall as in Figure 2.27. Using the above procedure for the SE-velocity, the SE-velocity is simply copied from the FS-velocity (constant extrapolation). In terms of mass fluxes, the mass flux through the SE-boundary is much larger than the mass flux through the FS-boundary. This leads to a larger FS-velocity in the next time step to compensate for the large mass flux that enters the cell through the SE-boundary. But this larger FS-velocity is copied again to the SE-velocity the next time step. If this configuration stays the same for several consecutive time steps, the FS-velocity will 'blow up', which finally results in divergence of the method. This problem only occurs when the aperture belonging to the velocity used for the extrapolation is smaller that the aperture of the SE-velocity. To prevent this problem, the aperture of the FS-cell should be taken into account. So, calculating the SE-velocity cannot simply be done by copying the FS-velocity, but by using

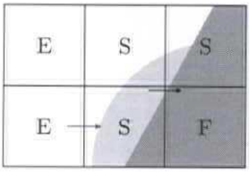


Figure 2.27: Configuration with small FS-face, adding energy when the aperture of the FS-face is not accounted for in the calculation of the SE-velocity

$$u_{SE} = \frac{A^x_{FS}}{A^x_{SE}} u_{FS}.$$

Then, the SE-velocity will not give rise to an increase of the FS-velocity. This has been incorporated in the method by accounting for apertures if the aperture belonging to the velocity used for extrapolation is smaller than the aperture of the SE-velocity. Note that accounting for apertures in the case that the factor A^x_{FS}/A^x_{SE} is larger than one, can cause instabilities as explained in the description of method 1 (Equation (2.22)). Using the above described method ensures a very robust treatment of the free surface boundary condition for SE-velocities.

2.3.11 Velocities at the free surface: EE-velocities

Between two surface cells a momentum equation is solved, which means that EE-velocities are needed as boundary condition (see Figure 2.28). The EE-velocities are determined using the tangential free surface condition given by Equation (2.4). Instead of discretising

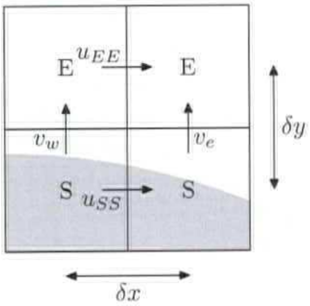


Figure 2.28: Calculation of EE-velocity boundary conditions

$\partial u_n/\partial t$ and $\partial u_t/\partial n$ in arbitrary directions, two Cartesian directions are chosen depending on the orientation of the EE-velocity. In the example of Figure 2.28, where the SS- and EE-velocity are in x-direction and the SE-velocities in y-direction, the normal direction is chosen to be the y-direction and the tangential direction is the x-direction. This leads to the following equation

$$\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)=0,$$

which is discretised using a central discretisation, giving for the required u_{EE}

$$u_{EE}=u_{SS}-\frac{\delta y}{\delta x}(v_e-v_w).$$

In three dimensions, dependent on the direction of the EE-velocity and the SE-velocities, one of the following equations is chosen for the calculation of the EE-velocity:

$$\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}=0,\quad \frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}=0,\quad \frac{\partial w}{\partial x}+\frac{\partial u}{\partial z}=0.$$

In the case that an EE-velocity is surrounded by more than one SS-velocity, the EE-velocity is calculated as an average using the SS-velocities. Exact details about which procedure is used in the different configurations can be found in [26].

2.4 Free surface displacement

Once the pressure and velocity field have been calculated, the free surface will be displaced using a method based on the Volume-of-Fluid (VOF) function. The VOF function is a discrete function with values between zero and one in each cell, indicating the fraction of the cell that is filled with fluid. When the VOF function $F^s=1$, the cell is completely filled with fluid, whereas when $F^s=0$, the cell is empty. When taking into account the geometry apertures of a cell, the following holds: $0\leq F^s\leq F^b\leq 1$.

Every time step the free surface is displaced using the advection Equation (1.1). In this thesis, two different methods are discussed: the original Hirt-Nichols method [44] and the method of Youngs [98]. Both methods are described, together with an enhancement:

Ref.	Reconstruction	Advection	2/3D
Biausser [4]	PLIC	segment-Lagrangian VOF	3D
Cerne [12]	PLIC (LVIRA)	direction split	2D
Guyffier [38]	PLIC	direction split	3D
Harvie [40]	PLIC	stream scheme	2D
Harvie [41]	PLIC	defined donating region (DDR)	2D
Hirt [44]	SLIC stair stepped	multidimensional (min/max limiters)	2D
Kothe [55]	SLIC stair stepped	Hirt-Nichols VOF	2D
Kothe [56]	PLIC	dir. split and multidimensional	3D
Muzaferija [67]	no reconstruction	high resolution interface capturing (HRIC)	2/3D
Noh [69]	SLIC	direction split	2D
Pilliod [75]	PLIC (ELVIRA)	dir. split and multidimensional	2D
Rider [78]	PLIC	dir. split and multidimensional	2D
Rudman [79]	no reconstruction	FCT-VOF (flux corrected transport)	2D
Scardovelli [81]	PLIC	direction split EI-LE (Eulerian implicit-Lagrangian explicit)	2D
Tank [88]	PLIC	direction split	2/3D
Youngs [98]	PLIC	direction split	2D

Table 2.1: Users of different VOF methods in literature

the introduction of a local height function. Results of some standard tests are shown: advection of a square and circle, rotation of a slotted disk, three-dimensional rotation of a notched brick and the deformation of a circle in a single vortex velocity field. Finally, the methods have been tested on a real case: a breaking dam in two dimensions. But first, an overview of the various VOF methods found in the literature is given.

2.4.1 Overview of VOF methods

Since the first introduction of VOF methods in the seventies and eighties of the last century, they have become very popular. In this overview a selection of the literature is made, which gives an overview of recent developments in VOF methods. The VOF methods are often classed by two features: the way the interface is reconstructed and the method for advecting the interface. Table 2.1 presents the reconstruction and advection features of some published volume tracking methods. A column has been added with information about the spatial dimension in which the methods are employed.

For the interface reconstruction two main methods are in use. Firstly, the simple line interface calculation (SLIC) by Noh and Woodward [69], where the interface is said to be parallel with one of the coordinate axes (see the left of Figure 2.29). Hirt and Nichols [44] also use a SLIC kind of reconstruction, which is not performed explicitly, but in this method within a cell a stair stepped interface form can be created. The second method was first used by Youngs [98], where a piecewise linear interface calculation (PLIC) is done (see the right of Figure 2.29). The piecewise linear reconstruction is much more accurate than piecewise constant, which is why it is used in most of the recent VOF publications (see Table 2.1). The disadvantage of the PLIC method is that it is more

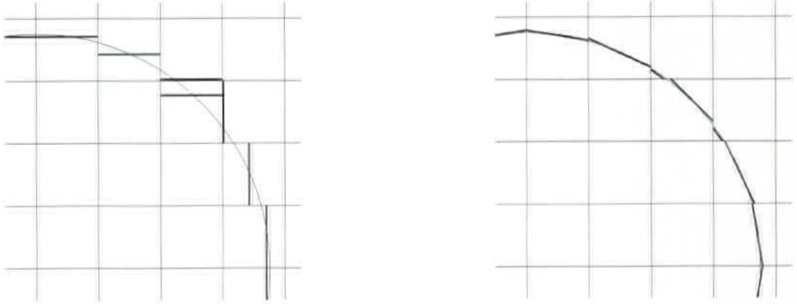


Figure 2.29: Reconstruction of a circular arc using SLIC (left) and PLIC (right)

complicated than SLIC, especially in three dimensions it is not easy to find a method for a piecewise planar reconstruction of the interface that is accurate and efficient. In the PLIC method the interface in a cell is represented by a plane (in two dimensions a line) given by

$$n_x x + n_y y + n_z z = c$$

To find the linear approximation of the interface in a cell, two steps are taken: first, the normal to the interface (n_x, n_y, n_z) is calculated using VOF data in surrounding cells; second, the plane constant c is determined, using the calculated normal and the VOF-value in a cell. The calculation of the normal has a large influence on the accuracy of the method. Several methods are in use, from which two are shortly described here.

- Youngs [98] calculates the normal in the four vertices of the cell (in 2D) using finite differences of the VOF-values ($\boldsymbol{n} = \nabla F / |\nabla F|$). Then the normal in the cell is taken as the mean of the normals in the four vertices. Also used by e.g. [4, 38, 40, 41, 56].

- Puckett [76] uses a least squares error minimisation technique (LVIRA). In a 3×3 block of cells a linear approximation of the interface is found with slope m by minimising the error between the VOF-values calculated using the linear approximation and the actual VOF-values in the 3×3 block of cells. The error is minimised by rotating the line under the constraint that the line exactly reproduces the volume fraction in the central S-cell. Pilliod [75] uses a same approach, but chooses the slope with minimal error from six candidate slopes (ELVIRA). The six candidates slopes are calculated using the backward, central and forward differences of the VOF-values in x- and y-direction. The error minimisation methods are also used by e.g. [2, 12, 40, 41].

In his paper Pilliod [75] determines the order of convergence of the different reconstruction methods. He concludes that only LVIRA and ELVIRA are truly second order methods (a line is reconstructed exactly), whereas Youngs' method and Hirt-Nichols VOF are first order.

After the interface has been reconstructed, it is advected. The advection can be done direction split or multidimensional. In the direction split approach the interface is reconstructed after which it is advected in the first coordinate direction. Then, a new reconstruction is made and the advection in the second coordinate direction is done based on the new reconstruction (e.g. [12, 38, 69, 98]). For symmetry, the order of the directions should be alternated. In multidimensional methods, there is only one reconstruction after which the advection is done in all directions simultaneously (e.g. [44, 78, 79]). In Table 2.1 it is indicated for some publications whether a direction split or a multidimensional method is used. Rudman [79] remarks that direction split gives more accurate results. However, Rider [78] states that the results are similar and argues that multidimensional advection could be a better choice for efficiency and symmetry considerations. In three dimensional computations direction split is mostly used, because it is very difficult to construct a multidimensional method for a proper flux calculation in three dimensions.

For the calculation of fluxes, many different methods are in use. Some are Lagrangian methods in the sense that markers are placed on the reconstructed interface (as in [4]). Most methods use a flux calculation, where the amount of fluid that should be fluxed is calculated geometrically by defining a region for every cell boundary from which the present fluid will be fluxed through the boundary. In many of the methods fluid can be fluxed twice or not fluxed at all, in which cases redistribution algorithms are used to conserve mass e.g. [40, 78]. In some two-dimensional methods, mass is rigorously conserved and no redistribution is necessary, e.g. [41, 60], but these methods can not be extended to three dimensions in a straightforward manner.

2.4.2 Hirt-Nichols VOF

In the original VOF method introduced by Hirt and Nichols [44] no explicit reconstruction of the interface is performed. To compute the VOF function at the new time level, fluxes are calculated over every cell face using a donor-acceptor method. Generally, the direction of the velocity defines whether a cell becomes a donor or acceptor cell. The flux through a cell face is calculated as the velocity times the area of the cell face A and the time step

$$\delta F^s = F^s \boldsymbol{u} \cdot \boldsymbol{n} A \delta t. \tag{2.23}$$

In the case of Figure 2.30, the flux through the cell face belonging to u_c becomes $\delta F^s_c = u_c \delta t A^x_c \delta y$. After all fluxes have been calculated (in two dimensions δF^s_e, δF^s_n, δF^s_w and δF^s_s) the VOF function is updated from time level n to $n+1$ using the explicit time integration

$$(F^s)^{n+1} = (F^s)^n + \frac{\delta F^s_e + \delta F^s_n - \delta F^s_w - \delta F^s_s}{\delta x \delta y}.$$

Away from the free surface this leads to a net flux of zero. In the neighbourhood of the free surface the calculation of the fluxes is somewhat more complicated. If the above procedure is used for fluxes at the free surface the newly calculated F^s may become larger than F^b or smaller than 0. This occurs because the cells near the free surface are not completely filled with fluid, such that the flux may exceed the amount of fluid present in the donor cell. Hence, in the original VOF method the flux near the free surface is computed as

$$\delta F^s_c = \min\{\frac{F^s_{AD}}{F^b_{AD}}|A^x_c u_c \delta t| + CF, F^s_D \delta x_D\}, \tag{2.24}$$

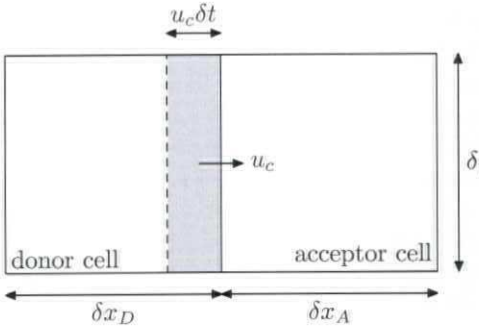


Figure 2.30: Displacement of the free surface using the donor-acceptor method

where

$$CF = \max\{\frac{F^b_{AD} - F^s_{AD}}{F^b_{AD}}|A^x_c u_c \delta t| - (F^b_D - F^s_D)\delta x_D, 0\}. \tag{2.25}$$

In these equations, F^s_D and F^s_A are the values of the VOF-function in the donor and acceptor cell respectively. The subscript AD stands for the donor or acceptor cell, depending on the orientation of the free surface. If the free surface is more or less advected normal to itself, or if either the acceptor cell or the cell upstream of the donor cell is empty, the acceptor cell is used, otherwise the donor cell is used. The min-operator in Equation (2.24) is present to avoid fluxing of more fluid to the acceptor cell than the donor cell contains. The max-operator in Equation (2.25) accounts for an additional flux CF if the amount of void in the donor cell exceeds the amount available. Despite these min- and max-limiters, the VOF-value can still exceed its ranges. In the original VOF-method, F^s is rounded off: when F^s exceeds F^b after fluxing the fluid, it is reset to F^b, when F^s is smaller than 0, it is reset to 0.

2.4.3 Local height function

The most well-known drawback of the VOF-method is the appearance of flotsam and jetsam, which are small droplets of fluid disconnecting from the free surface [40, 78]. In the left of Figure 2.32, a snapshot is shown of the free surface of a breaking dam flow at the end of the calculation (the water has gone through the domain back and forth). There are many small droplets, close to the free surface, which are due to the reconstruction and displacement of the free surface. Another drawback of the method is that mass is not conserved in the domain. Due to rounding off the VOF-values at the end of the displacement algorithm, water can be lost or gained. In the example of the dambreak about 7% of the water was lost. To prevent these problems of the original VOF-method, the displacement of the fluid in the neighbourhood of the free surface has been adapted. A local height function, which was already introduced for the calculation of the curvature in Section 2.3.9, is used to displace the fluid in surface cells. First, the direction of the local height function (horizontal or vertical) is determined using a 3 × 3 block of cells (in three dimensions a 3 × 3 × 3 block). Looking at the fluid distribution in the left of Figure 2.31, the local height function of the central S-cell is found by comparing the difference in VOF-values between the eastern and western cell with the difference between the northern and southern cell. In this case the vertical direction is chosen for the local height function, because the difference in vertical direction is larger (namely 0.9 compared to 0.7 in horizontal direction). Now, the local height function is defined by adding the VOF-values in a column of three cells, leading to $h_c = 1.1$ for the central S-cell (see Figure 2.31).

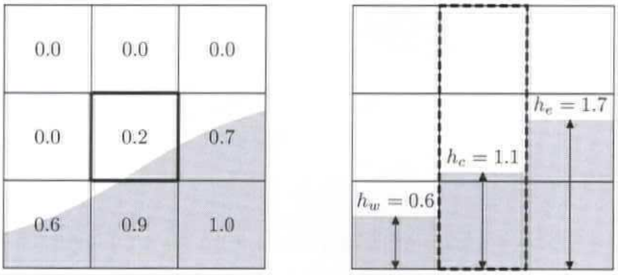


Figure 2.31: Definition of local height function for central S-cell

After calculating the fluxes across the cell boundaries of all three cells (the dashed-line region in Figure 2.31) as in classical VOF, not the individual VOF values of the three column cells are updated, but the height function is updated. The individual VOF values of the three cells are then calculated from the height of the fluid in the column.

When combining the original Hirt-Nichols VOF method with the local height function, the method is strictly mass conserving and almost no flotsam and jetsam appear. This combined method has been used in the dambreak simulation resulting in the snapshot in the right of Figure 2.32. The number of droplets is much smaller than in the original VOF-method. In Figure 2.33, the change of the total water volume during the dambreak simulations is shown. The loss of water using standard VOF is considerable, about 7 % after 6 seconds. In the adapted VOF method, the loss of water is only 0.02 %, so mass is almost perfectly conserved. Both methods will also be compared with each other in other tests, later in this section.

2.4.4 Youngs VOF

Another way to improve the accuracy of the VOF method, especially to get rid of the flotsam and jetsam, is a piecewise linear reconstruction of the interface. In the PLIC method in every cell the interface is represented by a plane, where the planes do not necessarily connect at a cell boundary between two cells. The plane is represented by

$$n_x x + n_y y + n_z z = c.$$

Basically, Youngs' method [98] is used for the reconstruction and advection of the interface, with some adjustments to simplify the calculations. First, the reconstruction is explained by describing the calculation of the normal and the plane constant. Then the advection is described. The adopted procedure has been explained in great detail in [26].

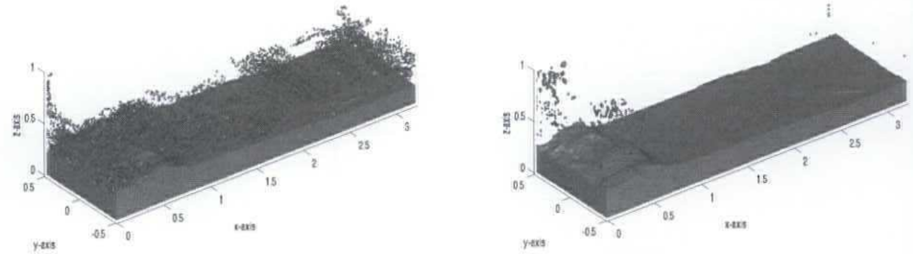


Figure 2.32: Snapshots at the end of dambreak flow simulations with different algorithms for the displacement of the free surface: original VOF (left) and VOF combined with a local height function (right)

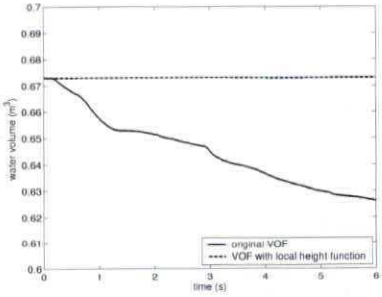


Figure 2.33: Change in water volume during a dambreak simulation using original VOF and VOF combined with a local height function

Calculation of the interface normal

Analytically, the unit normal is given by $\boldsymbol{n} = \nabla F^s / |\nabla F^s|$. The normal is calculated using finite differences after the idea of Rider and Kothe [78]. The equation for the (not yet normalised) normal $\tilde{\boldsymbol{n}} = \nabla F^s$ is discretised with respect to all neighbours, which results in two dimensions in 8 equations that can be written as

$$\begin{pmatrix} x_e - x_c & 0 \\ x_e - x_c & y_n - y_c \\ 0 & y_n - y_c \\ x_w - x_c & y_n - y_c \\ x_w - x_c & 0 \\ x_w - x_c & y_s - y_c \\ 0 & y_s - y_c \\ x_e - x_c & y_s - y_c \end{pmatrix} \begin{pmatrix} \tilde{n}_x \\ \tilde{n}_y \end{pmatrix} = \begin{pmatrix} F^s_e - F^s_c \\ F^s_{ne} - F^s_c \\ F^s_n - F^s_c \\ F^s_{nw} - F^s_c \\ F^s_w - F^s_c \\ F^s_{sw} - F^s_c \\ F^s_s - F^s_c \\ F^s_{se} - F^s_c \end{pmatrix}, \tag{2.26}$$

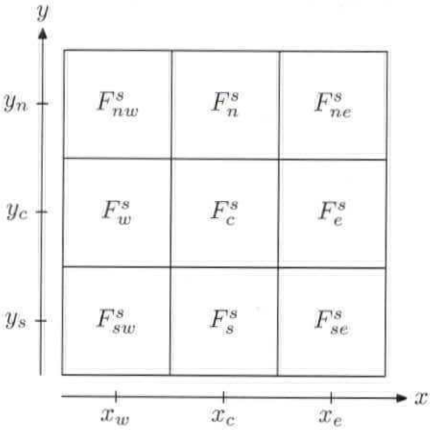


Figure 2.34: Explanation of the notation used in the computation of the interface normal

where the notation is explained in Figure 2.34 and $(\tilde{n}_x, \tilde{n}_y)$ is the non-normalised normal. In this system of equations, the first equation simply originates from a discretisation with respect to the eastern neighbour: $\tilde{n}_x = (F_e - F_c)/(x_e - x_c)$. The second equation originates from a discretisation with respect to the north-eastern neighbour. Therefore, four equations are formed:

$$\begin{aligned} (x_e - x_c)\,\tilde{n}_x &= F^s_e - F^s_c \\ (x_e - x_c)\,\tilde{n}_x &= F^s_{ne} - F^s_n \\ (y_n - y_c)\,\tilde{n}_y &= F^s_n - F^s_c \\ (y_n - y_c)\,\tilde{n}_y &= F^s_{ne} - F^s_e \end{aligned}$$

These four equations are summed and divided by 2, which results in the second equation in the system (2.26). The other equations are found in a similar way. This system of equations is written in matrix form as $A\tilde{\boldsymbol{n}} = \boldsymbol{b}$. The left and right hand side of this equation are multiplied by A^{T} giving a 2×2 (in three dimensions a 3×3) system of equations for $\tilde{\boldsymbol{n}}$. From this system $\tilde{\boldsymbol{n}}$ is solved using Gaussian elimination, after which $\tilde{\boldsymbol{n}}$ is normalised to find $\boldsymbol{n}$.

Calculation of the plane constant

After the normal is determined, the plane constant c is calculated using the constraint that the fractional volume of the cell for which $n_x x + n_y y + n z_z > c$, equals the VOF value in that cell. The correct value for c is found using the bisection method. The upper and lower bounds of the first bisection iteration are found by calculating the VOF values belonging to lines through the four vertices of the cell parallel to the interface (for an example, see Figure 2.35). The calculated VOF values are compared with the actual VOF value, after which an upper and lower bound are chosen. Then, in every iteration, the bisection domain is halved such that the actual VOF value is still contained in the bisection domain. Therefore, in every iteration, the VOF value belonging to the upper or lower bound value c has to be determined. The calculation of this VOF value is explained in the next paragraph. It is recommended to approximate the actual VOF value up to machine precision when determining the plane constant. Otherwise, when displacing the fluid using the calculated plane constant, small holes can occur in the fluid, causing loss of fluid.

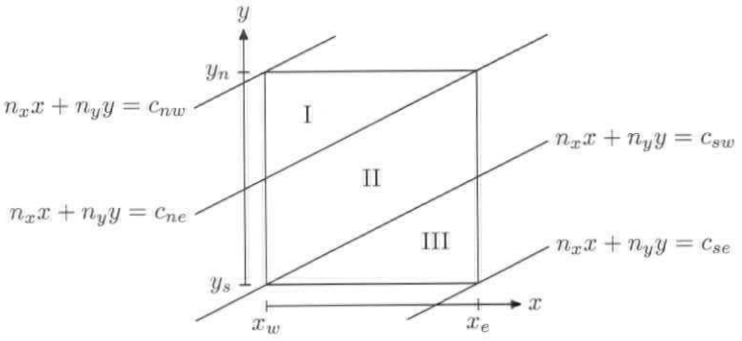


Figure 2.35: Determining an upper and lower bound in the bisection method used to find the plane constant c

Calculation of the VOF value for given normal and plane constant

In the bisection method used to find the plane constant c, the VOF value for a given normal and plane constant is needed. In [26, 56] a very elegant way of determining the VOF value in a cell, given the linear reconstruction of the interface, is described. The VOF value is determined by calculating the volume of the fluid present in that cell. In two dimensions the fluid volume in a cell (see Figure 2.36, where the shaded area is fluid) can be written as

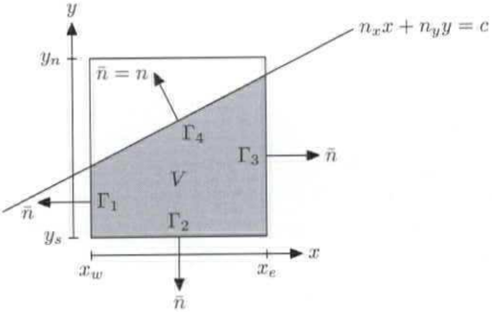


Figure 2.36: The VOF value of a cell with given interface reconstruction can be computed using a boundary integral over Γ_1, Γ_2, Γ_3 and Γ_4

$$V = \int_V dV = \frac{1}{2}\int_V \nabla \cdot \begin{pmatrix} x \\ y \end{pmatrix} dV = \frac{1}{2}\int_V \nabla \cdot \begin{pmatrix} x - cn_x \\ y - cn_y \end{pmatrix} dV,$$

where the term $\boldsymbol{cn}$ can be subtracted because it is a constant. Using the divergence theorem, this integral can be written as a boundary integral

$$V = \frac{1}{2}\oint_{\partial V} \begin{pmatrix} x - cn_x \\ x - cn_y \end{pmatrix} \cdot \bar{\boldsymbol{n}} dS,$$

where $\bar{\boldsymbol{n}}$ is the outward pointing normal at ∂V. The boundary ∂V consists of four parts, Γ_1, Γ_2, Γ_3 and Γ_4, so the boundary integral can be written as

$$V = -\frac{x_w - cn_x}{2}\int_{\Gamma_1} dy - \frac{y_s - cn_y}{2}\int_{\Gamma_2} dx + \frac{x_e - cn_x}{2}\int_{\Gamma_3} dy + \frac{1}{2}\int_{\Gamma_4} (n_x x + n_y y - c) d\Gamma_4.$$

The integral over the last boundary Γ_4 vanishes, because on the interface $n_x x + n_y y - c = 0$. The remaining integrals are rewritten as integrals over the cell faces

$$\begin{aligned} V \;=\; & -\frac{x_w - cn_x}{2}\int_{y_s}^{y_n} 1_{\{n_x x_w + n_y y > c\}} dy - \frac{y_s - cn_y}{2}\int_{x_w}^{x_e} 1_{\{n_x x + n_y y_s > c\}} dx \\ & + \frac{x_e - cn_x}{2}\int_{y_s}^{y_n} 1_{\{n_x x_e + n_y y > c\}} dy + \frac{y_n - cn_y}{2}\int_{x_w}^{x_e} 1_{\{n_x x + n_y y_n > c\}} dx. \end{aligned} \tag{2.27}$$

The last integral, which is zero in the case of Figure 2.36, is added to obtain a formula that is valid in all mixed cells. The integrals in Equation (2.27) are easily computed.

In three dimensions the same procedure is followed, resulting in a boundary integral over the six cell faces. Each of these six integrals can be calculated using the two-dimensional method described above, resulting in six equations that are similar to Equation (2.27). The method described in this section is very elegant, because a case-by-case study is not needed (as used in [98]). The method can be applied in every mixed cell.

Advection of the fluid interface

After the interface is reconstructed using the method described above, the interface is advected. Therefore, fluxes must be calculated over the cell faces. In three dimensions, it is very difficult to develop a multidimensional advection method. Most authors use a direction split method. The problem with a multidimensional method is that in a naive approach some fluid can be fluxed twice [78]. This is explained in Figure 2.37 where four grid cells are shown with a reconstructed interface (dotted line). The fluid contained in the medium-grey rectangles is fluxed, resulting in overlapping areas (the dark-grey regions). The fluid in these areas is fluxed twice giving problems with mass conservation.

In a direction split method also first a reconstruction of the interface is made. Then the fluid in the flux areas belonging to the horizontal velocity is advected. Then the interface is reconstructed again, after which the fluid is fluxed in vertical direction. And the same

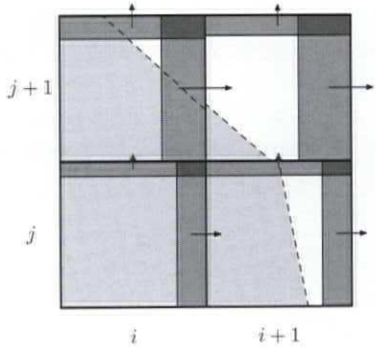


Figure 2.37: Problem when using a multidimensional advection scheme together with the flux calculation: the fluid in the dark shaded areas (if present) will be fluxed twice

procedure is repeated in the third direction. The fluxes are calculated in a straightforward manner. For the x-direction this is shown in Figure 2.38. The amount that is fluxed over the cell face is determined by the volume of the fluid inside the rectangle defined by $u_c \delta t$ times δy (in two dimensions). This amount is calculated using the procedure explained in the previous paragraph.

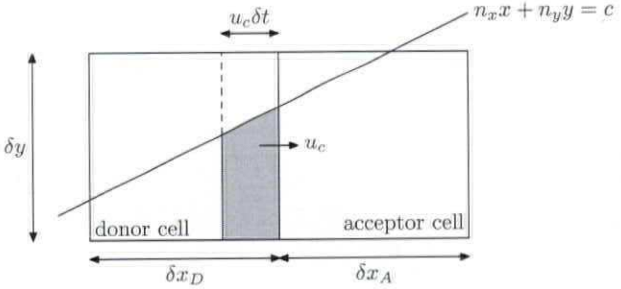


Figure 2.38: Computation of the flux for advection in x-direction: the light grey area is the amount of fluid to be fluxed over the cell face belonging to u_c

After the fluxes in x-direction are calculated (for cell i, j, k the fluxes δF^s_{i-1} and δF^s_i), the VOF function F^s is updated. The direction split method that is used is taken from Youngs [98]. First the complete cell is moved in a Lagrangian manner. The volume of the cell changes from V^0 to V^L in the following way

$$\begin{aligned} V^0 &= \delta x \; \delta y \; \delta z, \\ V^L &= V^0 - \delta V_{i-1} + \delta V_i - \delta V_{j-1} + \delta V_j + \delta V_{k-1} + \delta V_k. \end{aligned}$$

Here, δV_{i-1}, ... ,δV_k are the volume fluxes of the cell, which are calculated as the area of each cell face of which the fluid will be fluxed (so δV_i is the area of the region within the dashed line in Figure 2.38):

$$\begin{aligned}
\delta V_{i-1} &= A^x_{i-1jk} u_{i-1jk}\ \delta t\ \delta y\ \delta z, & \delta V_i &= A^x_{ijk} u_{ijk}\ \delta t\ \delta y\ \delta z,\\
\delta V_{j-1} &= A^y_{ij-1k} v_{ij-1k}\ \delta t\ \delta x\ \delta z, & \delta V_j &= A^y_{ijk} v_{ijk}\ \delta t\ \delta x\ \delta z,\\
\delta V_{k-1} &= A^z_{ijk-1} w_{ijk-1}\ \delta t\ \delta x\ \delta y, & \delta V_k &= A^z_{ijk} w_{ijk}\ \delta t\ \delta x\ \delta y.
\end{aligned}$$

The VOF function is unaltered during this step. In the advection of the VOF function the moved volume V^L is re-mapped to its original volume V^0 as described by the next Equations (2.28), (2.30) and (2.32). This is needed because the net flux in one direction, which is considered in a direction split method, does not equal zero. When taking into account the change in volume, the net flux will be zero and no problems with mass conservation occur. For the advection in x-direction the cell volume and VOF function are changed according to

$$V^x = V^L + \delta V_{i-1} - \delta V_i, \tag{2.28}$$

$$F^{s,x} = \frac{1}{V^x}(F^{s,n}V^L + \delta F^s_{i-1} - \delta F^s_i). \tag{2.29}$$

Here, δF^s_{i-1} and δF^s_i are the fluid volumes to be fluxed calculated earlier (for example δF^s_i is the light grey area in Figure 2.38) and $F^{s,n}$ is the VOF value at the beginning of the advection procedure. After the new VOF function is calculated based on the horizontal advection, a new reconstruction of the interface is made. Then the advection in y-direction is done using

$$V^y = V^x + \delta V_{j-1} - \delta V_j, \tag{2.30}$$

$$F^{s,y} = \frac{1}{V^y}(F^{s,x}V^x + \delta F^s_{j-1} - \delta F^s_j). \tag{2.31}$$

Then, the interface is reconstructed again and, finally, for the z-direction the following steps are taken

$$V^z = V^y + \delta V_{k-1} - \delta V_k, \tag{2.32}$$

$$F^{s,z} = \frac{1}{V^z}(F^{s,y}V^y + \delta F^s_{k-1} - \delta F^s_k). \tag{2.33}$$

Now, $F^{s,z}$ is the new VOF value on time level $n+1$ and $V^z = V^0$. Combining Equations (2.28) to (2.33), effectively the following advection equation is solved

$$\frac{F^{s,n+1} - F^{s,n}}{\delta t} + \frac{\partial(F^{s,n}u)}{\partial x} + \frac{\partial(F^{s,y}v)}{\partial y} + \frac{\partial(F^{s,z}w)}{\partial z} = (\nabla \cdot \boldsymbol{u})F^{s,n}. \tag{2.34}$$

which is the standard advection equation $DF^s/Dt = 0$ with a correction term $(\nabla \cdot \boldsymbol{u})F^{s,n}$ added on both sides of the equation. This correction term in the right-hand-side of the equation is needed because a direction split method is used. Otherwise, fluid will be lost or gained in the procedure. In the combined Equation (2.34), this term vanishes, because mass is conserved in the computational cells.

Of course, this correction procedure can only be applied in cells where mass conservation holds. When this is not the case, as in the surface cells in our method, the correction term $\nabla \cdot \boldsymbol{u}F^{s,n}$ can not be applied. In the current method no correction is made in the surface cells, so the equation

$$\frac{F^{s,n+1} - F^{s,n}}{\delta t} + \frac{\partial(F^{s,n}u)}{\partial x} + \frac{\partial(F^{s,y}v)}{\partial y} + \frac{\partial(F^{s,z}w)}{\partial z} = 0$$

is solved in surface cells in a direction split manner. This can lead to problems in mass conservation. To solve that, the local height function can be applied in mixed cells as explained in Section 2.4.3.

The direction split method can cause problems with symmetry. To prevent that, the sweep order is changed every time step: in the odd time cycles, the $x-y-z$ order is used and in the even time cycles the order is reversed to $z-y-x$. Another disadvantage of the direction split method compared to the multidimensional method is the calculation time. Every time step three reconstructions have to be performed. However, the calculation time for the displacement of the free surface does not increase with a factor 3, because in three dimensions the reconstruction of the interface is only done in a two-dimensional subset (only in cells with $F^s < F^b$).

2.4.5 Test of simple translation

The VOF methods of Hirt and Nichols and of Youngs are tested on various cases. First the performance of the methods is tested using standard kinematic tests. In these tests the velocity field in the whole domain is prescribed and only the free surface calculations are performed. So only the influence of the errors made in the free surface calculation is present. As a first test certain two-dimensional bodies of fluid formed as a circle or square are translated through a prescribed velocity field. The characteristics of this test are taken from [79] and it has also been performed by [40, 41]. Three fluid configurations are used: a hollow square, a hollow square with an angle of 26.57 degrees (arctan(1/2)) to the x-axis and a hollow circle. The fluid is translated in a velocity field of $(u, v) = (1, 0)$ and $(u, v) = (2, 1)$ for 500 time steps with a CFL-number of 0.25. This results in a translation of approximately 3 times the diameter of the fluid. The mesh size is 200 × 200 grid cells. The diameter of the body of fluid is covered by 40 grid cells and the distance between the outer and inner interface is 10 grid cells.

In Figure 2.39, the results for the translation in x-direction are shown. Four different methods are used for the reconstruction and advection: the standard Hirt-Nichols VOF method, Hirt-Nichols VOF with a local height function (as described in Section 2.4.3), Youngs' method (as described in Section 2.4.4), and Youngs' method combined with a local height function. In all figures the contour lines of 0.025, 0.5 and 0.975 are shown. For the translation of $(u, v) = (1, 0)$ the results are pretty good. No severe deformations are observed and all methods qualitatively give the same results.

Figure 2.40 shows the translation according to $(u, v) = (2, 1)$. Youngs' PLIC method clearly gives superior results compared to Hirt-Nichols VOF. The fluid bodies advected using Hirt-Nichols show some small holes in the fluid. Besides that, small droplets have been observed in the trace of the translation, which are not visible in this zoomed picture. The results improve considerably when using a local height function. Also the small droplets, which were observed in the results using the original Hirt-Nichols' method, are not present. In both results of Hirt-Nichols (with and without a local height function) the

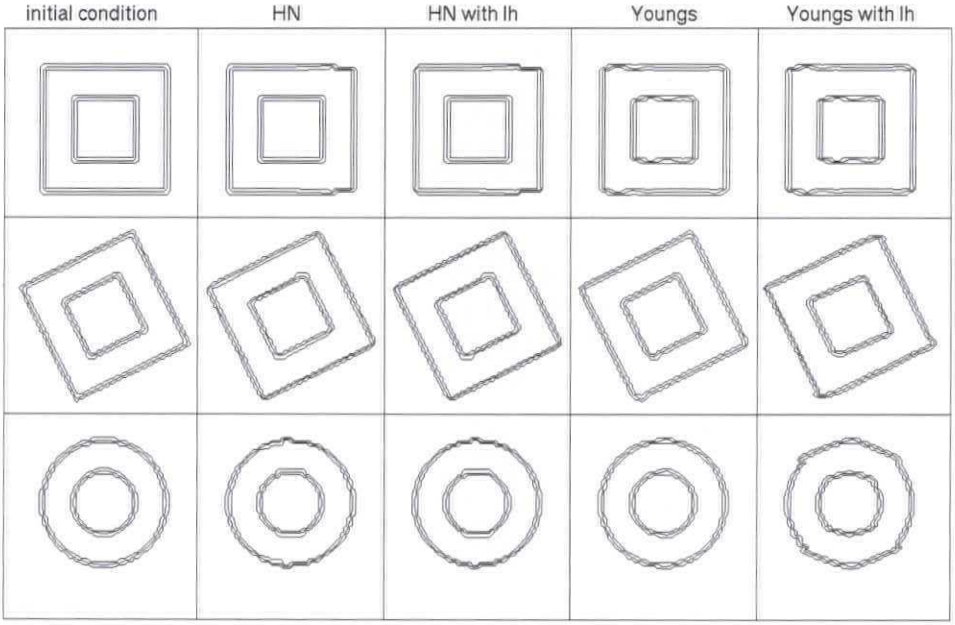


Figure 2.39: Translation of a square, rotated square and circle in a velocity field of $(u, v) = (1, 0)$: initial condition, Hirt-Nichols VOF, Hirt-Nichols VOF with a local height function, Youngs' method, and Youngs' method with a local height function

interface is distorted, a stairstepped interface is created. This originates from the interface reconstruction that is aligned with one of the coordinate axes. This same behaviour is observed by Rudman [79] who also shows results using original Hirt-Nichols. The PLIC method of Youngs shows good results, the shape of the fluid remains intact. The use of the local height function does not have much effect. The circle is a bit deformed when using the local height function together with Youngs' method, whereas in the case of the square adding the local height function gives slightly better results.

2.4.6 Rotation of a slotted disk

Zalesak [99] introduced the rotation of a slotted disk as a good validation case for interface advection algorithms. It has been used by many authors since, e.g. [1, 40, 41, 60, 75, 79, 81, 96]. In a domain of 4 by 4 meter a disk of fluid is initially placed at position (2,2.75). In the domain a rotation velocity field is prescribed, whereby the disk is rotated. The diameter of the disk is 1 meter. From the disk a slot has been removed with a width of 12 cm and the top of the slot is at the disk center. On a grid of 200×200 grid cells this results in 50 mesh cells in the diameter of the disk and 6 mesh cells in the slot width. The time

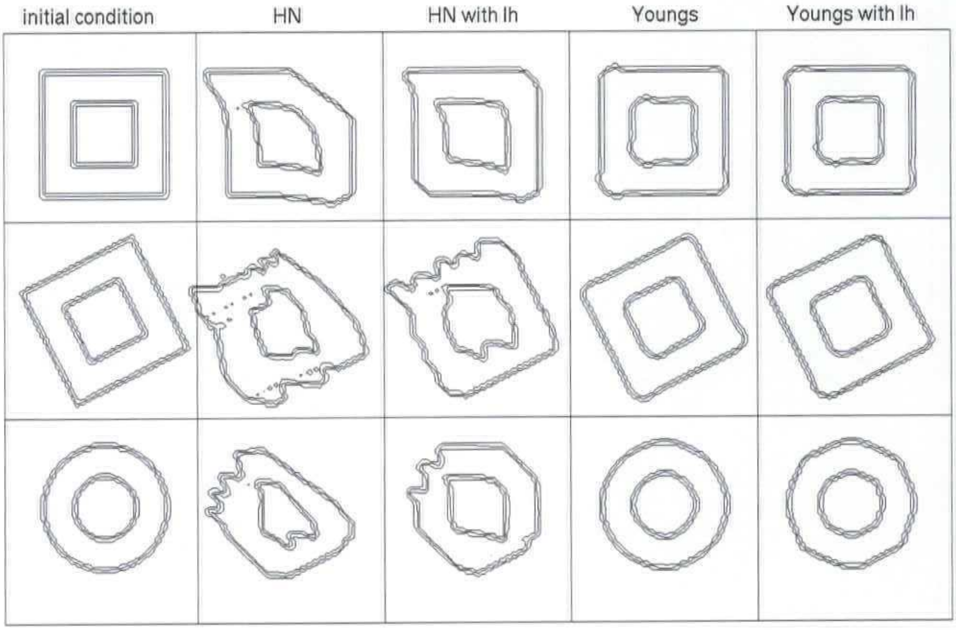


Figure 2.40: Translation of a square, rotated square and circle in a velocity field of $(u, v) = (2, 1)$: initial condition, Hirt-Nichols VOF, Hirt-Nichols VOF with a local height function, Youngs' method, and Youngs' method with a local height function

step is 0.0048 s and one rotation corresponds to approximately 2600 time cycles. Figure 2.41 shows the free surface profile of the slotted disk after one rotation using Hirt-Nichols' method with a local height function (left) and Youngs' method (right). The simulation has been performed on four different grids: 50, 100, 200 and 400 cells per direction. In the figure only the results of 100, 200 and 400 cells are shown. Hirt-Nichols combined with a local height function gives irregularities in the circular shape and the corners at the slot are rounded. The results improve on finer grids. Youngs' method is superior, but still has a rounding at the corners of the slot. Also, finer grids give better results.

To investigate the convergence of the methods, the error after one rotation has been computed using the formula

$$E = \frac{\sum_{i,j} |F^s(i,j) - F^{s,0}(i,j)|}{\sum_{i,j} F^{s,0}},$$

where $F^{s,0}$ is the initial solution. Also a quantification of the gain or loss of fluid is given

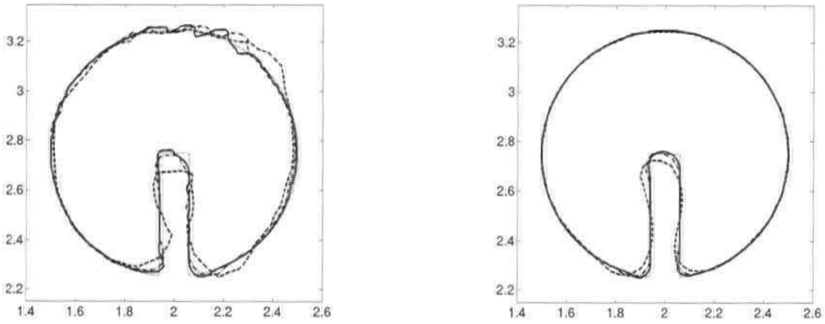


Figure 2.41: Slotted disk after one rotation using Hirt-Nichols VOF with local height function (left) and Youngs VOF (right); dotted line - exact solution, dashed line 100 × 100 grid cells, dashed-dotted line 200 × 200 grid cells; solid line 400 × 400 grid cells

using the formula

$$E_m = \frac{\sum_{i,j} F^s(i,j) - \sum_{i,j} F^{s,0}(i,j)}{\sum_{i,j} F^{s,0}(i,j)}.$$

The error is positive when the amount of fluid is increased and negative otherwise. In Table 2.2 the error in the interface reconstruction and advection, and the mass error are given for the four methods that are used on four different grids. In the upper part the error after one rotation is given together with the convergence rate under grid refinement. The errors in all methods decrease under grid refinement. Youngs' method without local height function gives the smallest error. Not a clear conclusion can be drawn about the order of the methods. As stated by Pilliod [75] none of the methods is second order, all are first order. But the refinement from 200 to 400 grid cells per direction shows an order smaller than one. In the lower part of the table the mass error is shown. Hirt-Nichols'

N	HN-	rate	HN+	rate	Y-	rate	Y+	rate
50	1.3851		1.3377		0.1046		0.1264	
100	0.1467	3.2	0.0971	3.8	0.0337	1.6	0.0586	1.1
200	0.1254	0.2	0.0458	1.1	0.0143	1.2	0.0294	1.0
400	0.1222	0.0	0.0285	0.7	0.0103	0.5	0.0169	0.8
	Gain or loss in fluid mass							
50	+0.49		-0.48e-4		-0.17e-2		-0.24e-4	
100	+0.36e-1		-0.77e-4		-0.66e-3		-0.16e-4	
200	+0.39e-1		-0.11e-3		-0.56e-3		-0.65e-5	
400	+0.45e-1		-0.88e-4		-0.20e-3		-0.24e-4	

Table 2.2: Top: error E and convergence rate of the slotted disk rotation using four different methods: Hirt-Nichols and Youngs, without and with a local height function (HN-, HN+, Youngs- and Youngs+); bottom: error in mass E_m using the same methods

method shows a large increase of mass, even by 50% on the coarsest grid. The mass error in both Youngs and Hirt-Nichols combined with the local height function is very small.

For mass conservation in Youngs' method on the finest grid of 400×400 grid cells double precision (i.e. 64 bit) computations are necessary. On a single precision 32 bit machine the amount of fluid using Youngs' method is decreased with 3%. It is not clear at the moment, why double precision is needed in computations on fine grids.

2.4.7 Rotation of a three-dimensional notched brick

To test the performance of the method in three dimensions, a notched brick is rotated as suggested by [56, 77]. The rotation vector is given by $\omega = (i, j, k)\omega_0$, with ω_0 a constant, thus the rotation is about the diagonal of a unit cube from corner $(0, 0, 0)$ to $(1, 1, 1)$. This results in a velocity field given by $\boldsymbol{u} = 1/\sqrt{3}\,(\mathbf{1} \times \boldsymbol{x})$ that is prescribed in the whole domain, where one full rotation is established in 2π seconds. The domain of 4 by 4 meter is covered with grids of $40 \times 40 \times 40$ and $80 \times 80 \times 80$ grid cells. The dimensions of the brick are $25 \times 20 \times 15$ grid cells on the coarse grid and $50 \times 40 \times 30$ cells on the fine grid. The center of the brick is positioned in the center of the domain, such that the axis of rotation points diagonally through the brick. From the right side of the brick (with largest x-value) 4 smaller cubes are removed with dimension of 5 and 10 grid cells on the coarse and fine grid, respectively. (See the left of Figure 2.42 where the initial condition on the coarse and fine grid are shown.) The time step is 0.0025 s resulting in a CFL-number of approximately 0.25 on the fine grid (based on velocities in the whole domain). Figure 2.42 shows the results of the VOF field after one rotation using two methods: Hirt-Nichols VOF with local height function (central column) and Youngs' method (right column). The notches in the brick are not very well resolved on the coarse grid. 5 cells per notch is clearly not enough. On the fine grid, the notches stay much sharper, especially using Youngs' method gives a quite good result. Furthermore, the resulting interface using Youngs' method is much smoother than when using Hirt-Nichols' method on both grids. Although the resulting interface using Hirt-Nichols' method is pretty much deformed, the brick is still recognisable in contrast to the SLIC method used in [77].

2.4.8 Single vortex

The translation and rotation tests shown so far are relatively simple in the sense that the fluid topology does not change. A more demanding test was developed by Rider and Kothe [78], where the body of fluid is deformed and thus the volume tracking method is tested more severely. Initially a circle of fluid with radius 0.15 is positioned in a unit square domain with its center at (0.5,0.75). The fluid body is stretched out by a single vortex velocity field given by the stream function

$$\Psi(x, y) = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y),$$

from which fluid velocities are deduced using $(u, v) = (-\partial\Psi/\partial y, \partial\Psi/\partial x)$. The circular body stretches and spirals about the center of the domain when put in the single-vortex field. To be able to quantify the results, the velocity field is reversed after $t = T/2$ seconds, with T the maximum simulation time. Then, in case of a perfect volume tracking

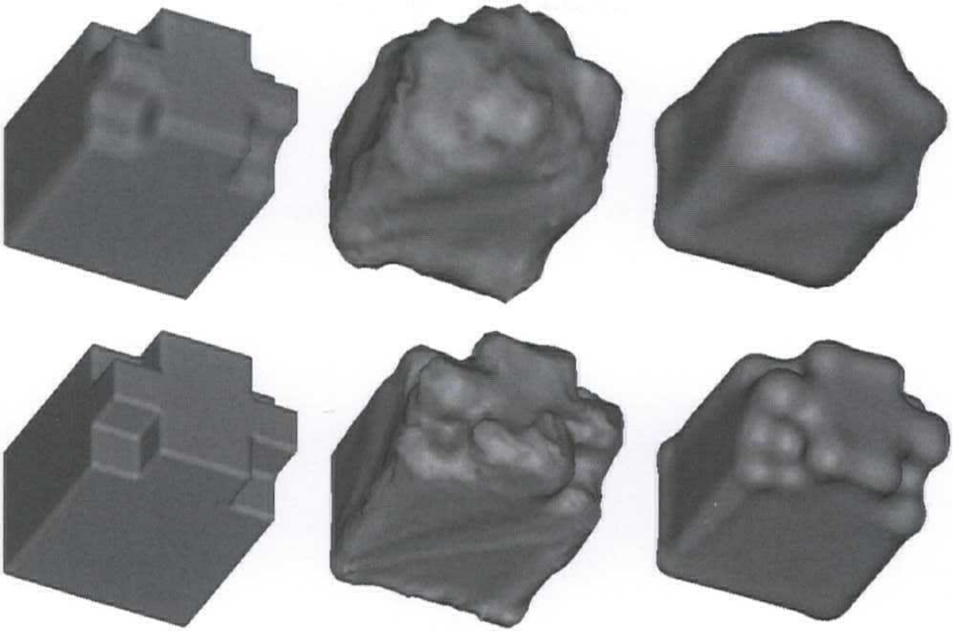


Figure 2.42: Rotation of a notched brick on a 40 × 40 × 40 grid (upper row) and a 80 × 80 × 80 grid (lower row); the first column is the initial condition, the second column Hirt-Nichols with a local height function and the third column Youngs' method

algorithm, the fluid will return to its initial state at the end of the computation. In this thesis results of four different computations are shown using the four displacement methods as before. The first computations are performed on a 64×64 grid with T=2 and T=8 respectively. The same computations are repeated at a 128×128 grid. Some of the results are shown in Figure 2.43. It is clear that Youngs' method performs best in all cases. Hirt-Nichols VOF shows a lot of small droplets and is not able to resolve the circle after the velocity field is reversed. Also with local height function the results do not improve much, although the number of droplets is decreased significantly. From the result with T=8 seconds on the fine grid using Youngs (in the last two rows and columns) it can be seen that a problem occurs when the fluid does not stick together, but chunks are being formed. In that case, the advection method is not able to reverse these chunks towards the circle, but small 'tails' are formed at the circle. This is also noticed by Cerne [12], who uses adaptive grid refinement to prevent the fluid from breaking up in chunks, which gives much better results.

2.4.9 Two-dimensional dambreak simulation

In this section the interface displacement methods are tested in a computation of a dambreak flow. So now also the velocity and pressure field are calculated using the methods described in Section 2.3. The configuration of the dambreak is chosen the same as in an experiment performed at MARIN, such that the results can be compared with

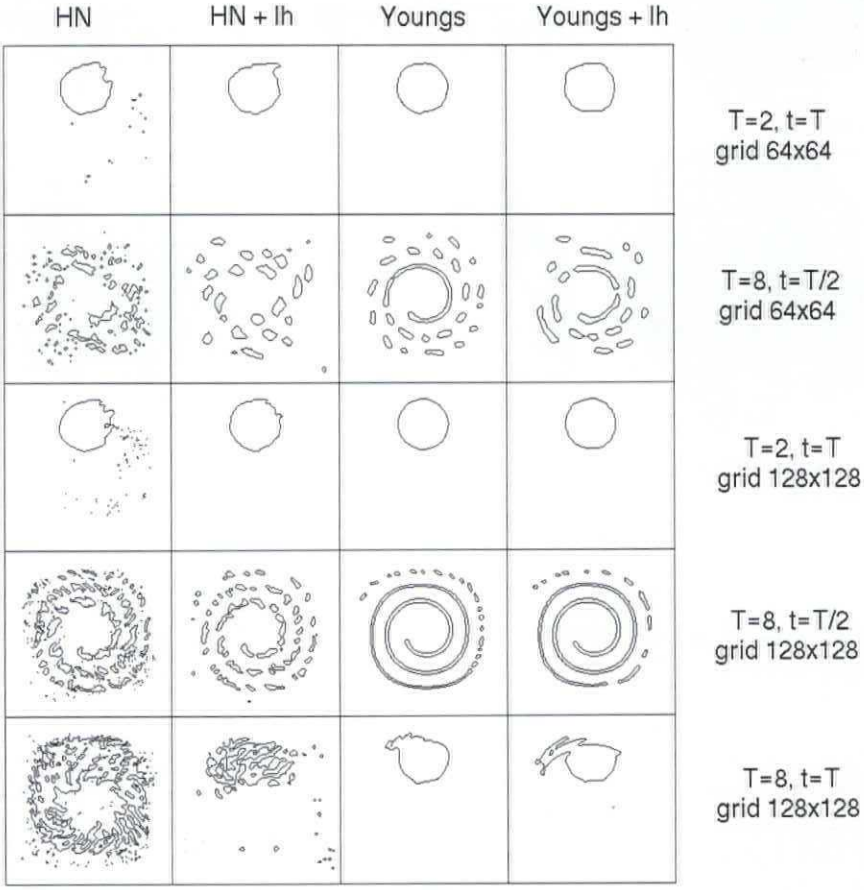


Figure 2.43: Single vortex computations with T=2 (row 1 and 3) and T=8 seconds (row 2, 4 and 5) on a grid of 64^2 cells (row 1 and 2) and 128^2 cells (row 3, 4 and 5); four different methods are used, from left to right: Hirt-Nichols without and with local height function, Youngs without and with local height function

measurements. In Section 2.6 the same configuration has been used, but there three-dimensional flow is studied with an object in the container. The container has a length of 3.23 m and is 1 m high with an open roof. In the right part of the tank, from $x = 2$ m to $x = 3.23$ m water of 0.55 m high is waiting to flow into the empty part of the tank. The domain is covered with a grid of 118 × 34 grid cells. As interface advection methods Hirt-Nichols' and Youngs' method are used, both with and without a local height function. To investigate the extra calculation time needed for Youngs' method in a real calculation, the calculation times of Hirt-Nichols' and Youngs' methods are measured and compared. It turned out that Hirt-Nichols' method needs 2 minutes and 20 seconds, whereas Youngs' method needs 3 minutes and 6 seconds to complete the simulation. So, the calculation time increases a bit, but not too much. In the left of Figure 2.44 a snapshot of the simulation has been shown after a simulation time of 0.8 seconds. From this snapshot it can be seen that the moment the water hits the left wall of the container is different for the four interface tracking methods. This is due to the amount of liquid in the container, which is not constant in all simulations. The right plot in Figure 2.44 shows a time history of the amount of liquid in the four simulations. Both Youngs' and Hirt-Nichols' methods loose fluid, especially in the early stage of the simulation, when the fluid jet is moving over the bottom of the container. The reason for the loss of water can be found when looking at the left of Figure 2.45, where a zoomed snapshot of the jet flowing over the bottom of the container is given in case of Youngs' method. The tongue of the jet consists of a surface cell with an empty cell at the left and above it. Both the SE-velocities, determined from extrapolation (see Section 2.3.10), are equal to 0, whereas the FS-velocity is quite large and pointing inwards the cell. Therefore, only fluid will be fluxed into the cell and nothing out, which can result in a VOF value larger than 1. At the end of the displacement algorithm, the VOF value is rounded to 1 if it is larger than 1, causing loss of fluid. To see the effect of the non-mass-conserving method for SE-velocities, a mass conserving way of determining SE-velocities is adopted (method 1 in Section 2.3.10). The amount of fluid in the container using both methods for

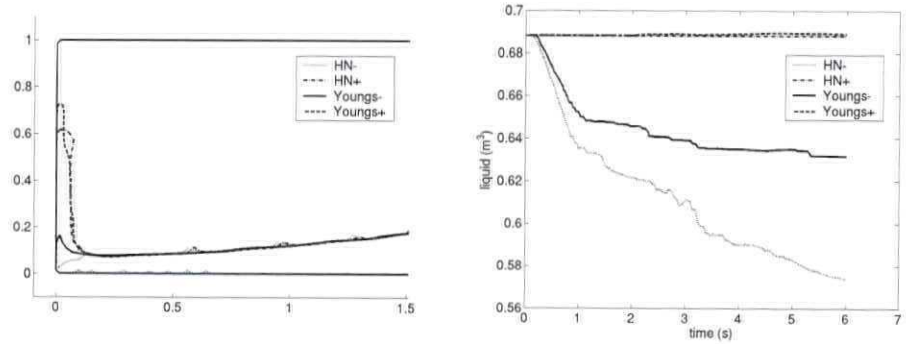


Figure 2.44: Left: snapshot showing the free surface at the left wall of the container in a dambreak simulation after 0.8 seconds using four different interface advection methods (Hirt-Nichols and Youngs with (+) and without (-) local height function); right: time history of the amount of liquid in the container

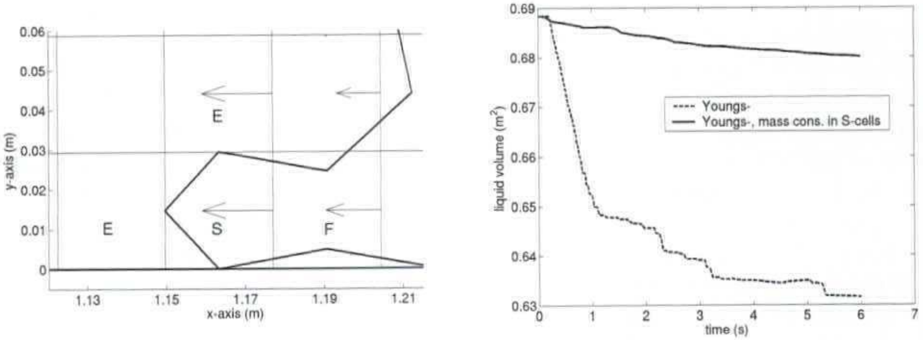


Figure 2.45: Left: zoomed snapshot showing velocities in the water jet over the bottom of the container; right: time history of the amount of liquid in the container using Youngs' method with the standard way and a mass conserving way of determining SE-velocities

SE-velocities is shown in the right of Figure 2.45. Using the mass conserving method for SE-velocities shows much less loss of water, and the loss is more gradually. Unfortunately, the mass conserving method for SE-velocities can not always be used (for robustness and accuracy of the method, see Section 2.3.10). Therefore, to conserve mass when using Youngs' method the local height function should be added, which solves the problem.

In the experiments the water height has been measured at different positions in the container. The top row in Figure 2.46 shows the water height in the reservoir that is initially filled with water, at $x = 2.66$ m. The bottom row shows the water height close to the left wall at $x = 0.5$ m. The four different methods for the interface calculation are shown in the different columns. From the results it can be seen that the methods without the local height function (first and third column) are a bit lagging behind in time. The best agreement between the calculation and experiment can be found when using Youngs' method in combination with a local height function (last column). The first bump in the top row and even the second bump in the bottom row are very well predicted.

2.4.10 Concluding remarks

Concluding this section about the different methods for free surface displacement, the following remarks can be made. First, when doing simulations with a prescribed velocity field as translation, rotation and the single vortex simulation, Youngs' method is most accurate in all cases. Combining Youngs' method with a local height function does not violate the results a lot and still gives very acceptable results. Hirt-Nichols' method performs less good in these kind of simulations, especially when no extra measures are taken to prevent flotsam and jetsam. The introduction of a local height function significantly improves the results.

Second, when doing a simulation of real fluid flow, like the dambreak shown in this section, the conclusions are not that clear any more. Youngs' method results in a smoother free surface profile than Hirt-Nichols' method. But when examining water heights in the tank for the dambreak simulations Hirt-Nichols and Youngs give comparable results, a

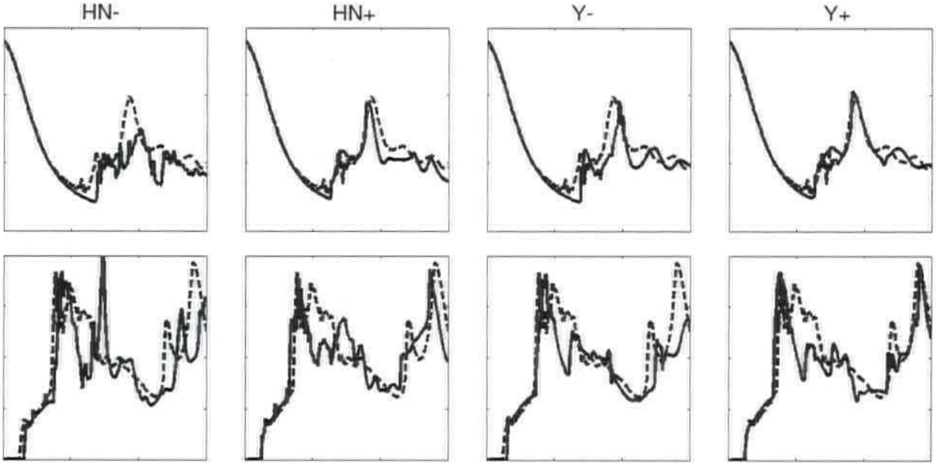


Figure 2.46: Water heights in the container of the dambreak simulation simulation (solid line) compared with measurements (dashed line); measurement position in reservoir (top row) and close to left wall (bottom row); four different methods are used, from left to right: Hirt-Nichols without and with local height function and Youngs without and with local height function

bit in favour of Youngs. In both methods mass is not conserved in the domain when the methods are not combined with a local height function.

So, in a simulation with real physical conditions Youngs' method with local height function seems to produce the best results. Hirt-Nichols' method combined with local height function also produces satisfying results. More different situations should be studied to confirm this conclusion. In the next chapter wave simulations will be performed where also attention will be paid to the free surface displacement algorithm.

2.4.11 Free surface displacement and moving objects

The displacement of fluid in the neighbourhood of a moving object is not straightforward. Problems with mass conservation occur when, for example, a cell is partly open for fluid at one time step, but the object is covering the complete cell after moving (as in Figure 2.47). Then, in one time step, the open edge of a cell is closed and a fluid cell has changed into a boundary cell. If the fluid is then moved according to the donor-acceptor method using fluxes as calculated in Equation (2.23), no fluid will be moved because the area of the cell face A equals zero. But there is still fluid inside the cell, which will be lost if nothing special will be done. Also, small holes can be created near the moving object when the object is moving, and the available space is not filled by fluid.

There are two ways to prevent the loss of mass. First, the local height function, introduced earlier to diminish flotsam and jetsam and to take care of mass conservation (see Section 2.4.3), can also be used to prevent loss of mass in this case. The extra fluid

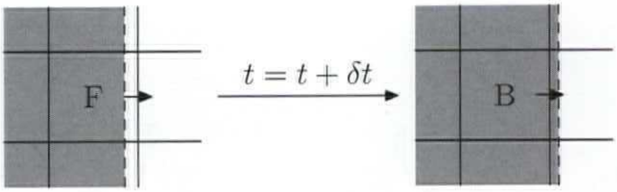


Figure 2.47: A fluid cell becomes a boundary cell, giving mass conservation problems when the fluid is displaced in a straightforward manner

in cells that have F^s larger than F^b is redistributed over the neighbours. This option is to be used with care, since the local height function is not designed to be used away from the free surface. The 'redistribution' of fluid can cause other neighbouring cells to become too full.

Another way is to displace the fluid by not using the calculated velocity field, but using a slightly different velocity field. To determine this velocity field, a Poisson equation is solved, that is modified from the standard pressure Poisson equation, resulting in an auxiliary pressure field Q. In the right-hand-side of this Poisson equation an extra term is added, which is the difference between the geometry apertures on the old and new time level. By adding this extra divergence term, a pressure field is created that forces the fluid to respond on the motion of the object. When the object is entering the cell, extra space is created, whereas space is filled when the object is leaving the cell. The velocity field based on the gradient of this auxiliary pressure field $\boldsymbol{u}_Q$ is used for the displacement of the free surface (and only for that). Further, some apertures are modified indicated by $\tilde{A}$, such that apertures of cells that have become boundary cells, but are not yet empty, are not closed and permit fluid to flow out. The fluxes used in the VOF-method are now calculated using

$$\delta F^s = \boldsymbol{u}_Q \cdot \boldsymbol{n}\tilde{A}^x \delta t. \tag{2.35}$$

This method has been adopted in the simulation of a small moving box, completely submerged by water in a closed domain. In Figure 2.48 a time series of the total amount of fluid in the domain is plotted. In the simulation indicated with a solid line, the standard method of Hirt-Nichols is used without local height function or adaption of the velocity field. A gradual loss of fluid occurs. When using the local height function, the result is much improved and the amount of liquid in the domain is almost constant (except for a drop of the fluid level in the first time step). The same accounts for the simulation with the adaption in the velocity field using the auxiliary pressure field: the fluid level is almost constant.

2.5 Pressure spikes due to moving objects

In Section 2.3.10 the occurrence of spikes in the pressure signal due to the determination of SE-velocities at the free surface has been discussed. Such numerical pressure spikes do also occur as a consequence of the motion of an object. Because a cut cell method is used on a fixed Cartesian grid, the objects are moving through the grid, resulting in changing

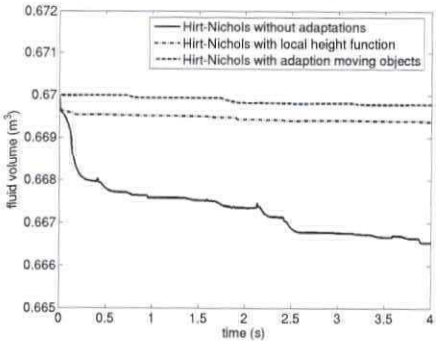


Figure 2.48: The total amount of liquid in the domain when moving a small box, with and without an adapted procedure for moving objects in the free surface displacement

geometry apertures and labels. Fekken describes in [23] some possible causes for pressure spikes due to moving objects. The first reason for spikes originates from the fact that the moving object possesses sharp corners. These pressure spikes can be prevented as will be described below. The second important reason Fekken mentions is the position of the pressure in a cut cell.

Objects with sharp corners

The first reason for pressure spikes due to moving objects mentioned by Fekken [23] is that the moving object possesses sharp corners. A test case has been designed to investigate this problem. A square is moved orthogonally to one of its sides through the grid without the presence of a free surface as shown in Figure 2.49. The geometry and edge apertures are calculated exactly to eliminate any possible influence of an inaccurate apertures calculation. At a position below the box, the pressure has been monitored (in point P). In the left of Figure 2.50, the resulting pressure signal at this point is shown. A very spiky behaviour can be observed: each time the square is entering a new cell and the edge aperture changes from one time step to another, the pressure produces a spike. This spike has a direct relation with the time step: if the time step is twice as large, the spike becomes twice as small.

In [23] a one-dimensional analysis is performed to discover the origin of the pressure spikes. Consider an object with a sharp corner that is moving through a channel where only one cell is used in vertical direction (see Figure 2.51). The object is moving with velocity $u_b = 2$, while the fluid in the channel is moving with a constant flow rate with $u = 1$. Because of mass conservation in the computational cell, the velocity u_w in the left of Figure 2.51 that is a velocity at a wall half covered by the object equals zero. When omitting convection, diffusion and external forces, the pressure gradient Gp_h equals zero when the front of the moving body does not cross a cell face, because all velocities keep the same value:

$$u^{n+1} = u^n - \frac{\delta t}{\Omega^{n+1}\rho} G^{n+1} p^{n+1}.$$

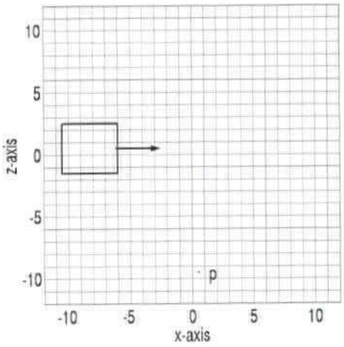


Figure 2.49: Simulation of a moving square without a free surface to investigate the occurrence of numerical spikes due to sharp corners

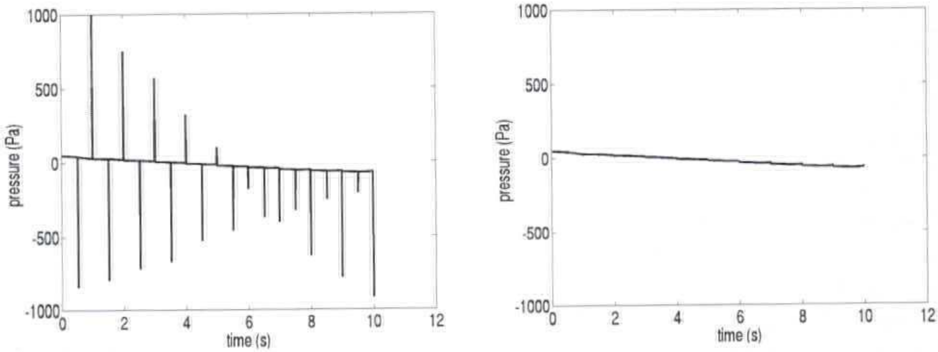


Figure 2.50: Pressure signal of point P in the moving box simulation: without (left) and with (right) a splitting procedure for the pressure to prevent spikes

When the front of the body crosses a cell face from time step n to $n+1$ as in Figure 2.51, the pressure gradient across cell face c is given by

$$G^{n+1}p^{n+1} = \frac{\Omega^{n+1}\rho}{\delta t},$$

since $u_c^n = 1$ and $u_c^{n+1} = 0$. According to the discretisation of the pressure gradient, Equation (2.10), the difference between the pressure in the eastern and western cell is computed as

$$(p_e - p_w)^{n+1} = \frac{\Omega^{n+1}\rho}{A_x^c \delta y \delta t} = \frac{2\Omega^{n+1}\rho}{\delta y \delta t}.$$

A pressure spike is created, caused by the jump in the edge aperture A_c^x from time step n to $n+1$. From the above equation it can be seen that decreasing the time step enlarges the spike: when the time step is halved, the magnitude of the spike is doubled. The spikes are purely numerically created as a consequence of the aperture 'jump' in combination

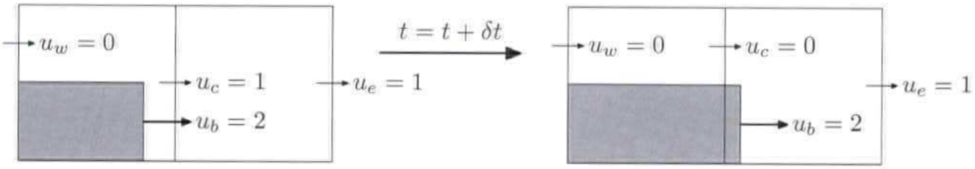


Figure 2.51: Illustration of a sharp corner crossing a cell face from time step n to $n+1$

with the no-slip condition at the object that forces the fluid to move according to the moving body.

To prevent the occurrence of these spikes, the computed pressure will be split into two parts: the physical pressure and a non-physical part of the pressure

$$p(x, y, z) = p_{phys}(x, y, z) + \phi(x, y, z).$$

The idea is that the spike is contained in the non-physical part of the pressure $\phi(x, y, z)$, such that only the physical pressure part is used in the computations. This idea is related to the use of Lagrange multipliers in the fictitious domain method [32]. There the moving bodies are filled with the surrounding fluid, and Lagrange multipliers are defined over the moving body to force the rigid body motion inside the moving bodies. The potential ϕ has to be calculated such that the velocity field at the new time level can be made divergence free. To accomplish the split of the computed pressure, first the crossing of the sharp corner of the square with a cell has to be detected. Second, a temporary velocity and pressure field are calculated based on the new apertures as usual, but using old apertures in cells where the sharp corners have been detected. The aperture 'jump' has now been prevented and thus no pressure spikes are present in the temporary pressure field. Now, the velocity field has to be made divergence free by calculating the total pressure, including the peaks in the potential ϕ. To accomplish this, a second Poisson equation is solved based on the temporary velocity field and new apertures everywhere. This splitting procedure for the pressure has been adopted in the test case of a moving square, where the geometry apertures are calculated exactly again. The resulting pressure signal has been plot in the right of Figure 2.50. The signal does not contain spikes any more.

When the geometry apertures are not calculated exactly, but approximated as in the computations shown in this thesis (see Section 2.3.2), the resulting pressure signal does not show very large spikes. In Figure 2.52 the pressure signal of point P in Figure 2.49 is shown for the cases with and without the splitting procedure of the pressure. Already in the case without the splitting procedure (the left plot) the large spikes are not present, because the aperture jumps are much more gradually: due to the approximate aperture calculations, the apertures change more smoothly from one time step to another, preventing the large pressure spikes that are present in Figure 2.50. So in this case, the splitting procedure for the pressure does not have a very large influence any more.

To examine the influence of the splitting procedure in the case of a realistic situation with a free surface, a cylinder entry case has been studied. The cylinder with a diameter of 1 m is positioned above the free surface at the start and then entering the free surface

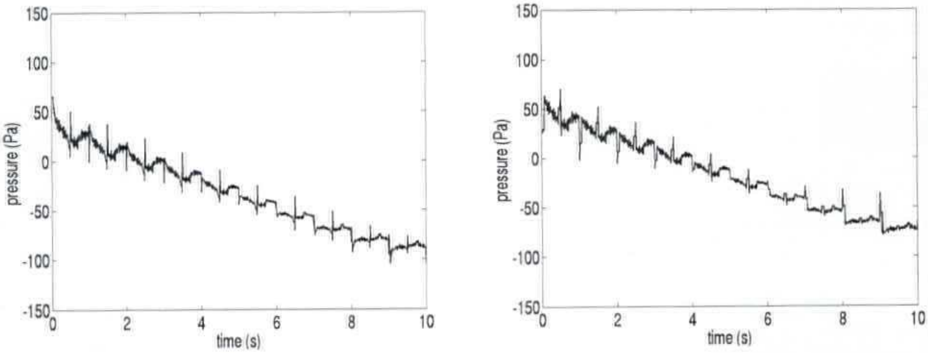


Figure 2.52: Pressure signal of point P in the moving box simulation where apertures are calculated approximately: without (left) and with (right) a splitting procedure for the pressure to prevent spikes

with a constant velocity of -1 m/s. Only the initial stage of impact is simulated. A uniform grid of 400 × 400 grid cells is used. The resulting vertical slamming force on the cylinder is shown in Figure 2.53. Although the force is an integral over the pressure that should smoothen a pressure spike occurring in one cell, the spikes are also present in the force signal. This is caused by the incompressibility of the fluid, which forces a pressure pulse to travel over the whole domain in infinitely small time. So a pressure spike originating from one cell in the domain is visible in all cells in the domain. From Figure 2.53 it can be concluded that the spikes are not that large in such a simulation. Also, the splitting procedure for the pressure does decrease the number of spikes somewhat, but the influence is not very large. No clear conclusion can be drawn about which method gives best results. One of the spikes, occurring in the initial stage of impact, is even larger when using the splitting procedure. There are several causes for the relatively small influence. First, the edge apertures are not calculated exactly, which takes care of a more smooth change in the apertures from one time step to another. Second, the

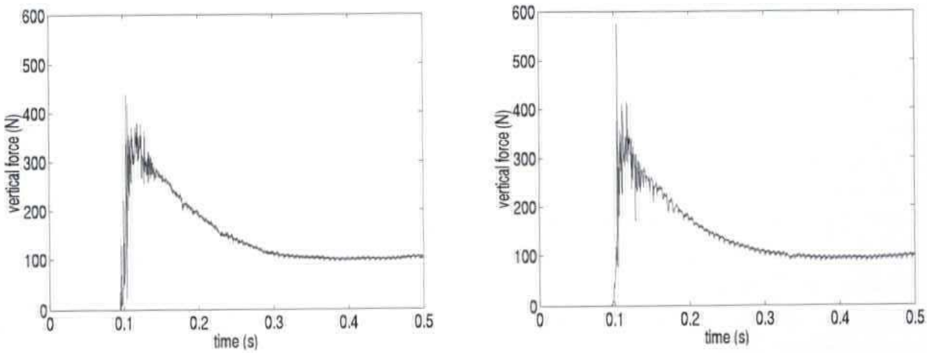


Figure 2.53: Vertical force on a cylinder entering the free surface; simulation without (left) and with (right) a splitting procedure for the pressure to prevent spikes

jumps in the edge apertures are not that large, because no corner is present that is parallel with the coordinate axes as is the case in the moving box. So, in this case the splitting procedure for the pressure does not improve the results much, whereas it is costly in computational time. Every time a simulation is set up, a trade off should be made between the occurrence of the spikes and the computational time it takes to determine ϕ by solving a second Poisson equation.

2.6 Validation: flow of a breaking dam

At the Maritime Research Institute Netherlands (MARIN) experiments have been performed for breaking dam flows. These experiments can be seen as a simple model of green water flow on the deck of a ship. The dambreak is a very popular validation case, because the set-up is easy: no special in- or outflow conditions are needed. A large tank of 3.22 by 1 by 1 meter is used with an open roof. The right part of the tank, from $x = 2$ m, is first closed by a door. Behind the door, 0.55 meter of water is waiting to flow into the tank when the door is opened. This is done by releasing a weight, which almost instantaneously pulls the door up. In the tank a box has been placed that represents a scale model of a container on the deck of a ship. The box is 16 cm long, 40 cm wide and 16 cm high, with the left side of the box positioned at $x = 0.67$ (see the left of Figure 2.54).

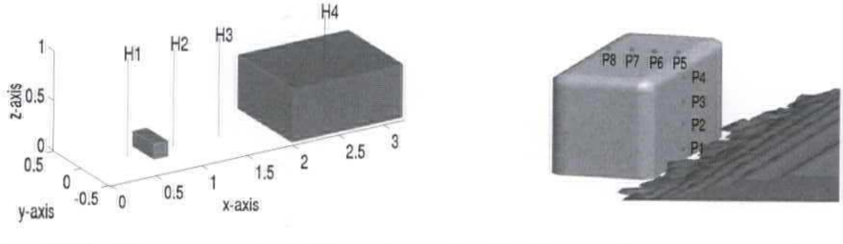


Figure 2.54: Measurement positions for water heights and pressures in the dambreak experiment

During the experiment measurements have been performed of water heights, pressures, and forces. In Figure 2.54, the positions of the measured quantities are shown. Four vertical height probes have been used; one in the reservoir and the other three in the tank at positions $x = 0.5$, 1.0, 1.5 and 2.66. The box was covered by eight pressure sensors, four on the front of the box at height $z = 0.025$, 0.063, 0.099 and 0.136, and four on the top of the box at $x = 0.806$, 0.769, 0.733 and 0.696. The sensors on the front of the box are positioned 0.026 m left of the center line $y = 0$ and the sensors on the top of the box 0.026 m right of the center line. The forces on the box were also measured. To determine the velocity of the water when entering the tank, a horizontal wave probe is used near the side wall of the tank.

As initial configuration of the simulation, the water in the right part of the domain is at rest. When the simulation is started, due to gravity the water starts to flow into the empty part of the tank. A fine grid of 236 × 76 × 68 grid cells has been used with some stretching towards the bottom of the tank. The simulation is continued for 6 seconds with an automatically adapted time step using maximum CFL-numbers around 0.75, resulting in a time step of the order of 0.001 seconds. In Figure 2.55 two snapshots of the early stages of the simulation are shown together with some images of the video from the experiment (at the same instants of time). The smaller pictures inside the snapshots show the water in the reservoir. There is a very good agreement between the snapshots of simulation and experiment. The time instant when the water is first hitting the box is the same. The shape of the free surface, bending a bit forward in the second picture, is seen in both experiment and simulation. In the simulation, the free surface has some ripples, which can be suppressed by using a piecewise linear reconstruction of the free surface [98] instead of the reconstruction aligned with the coordinate axes used in this simulation.

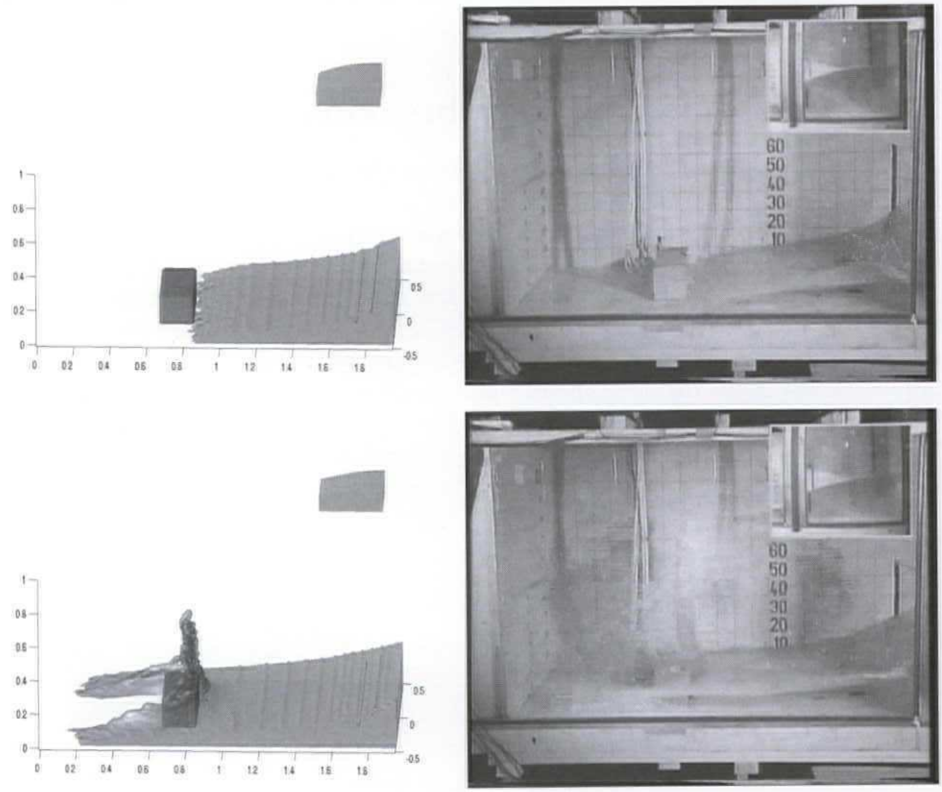


Figure 2.55: Snapshots of a dambreak simulation with a box in the flow compared with experiment at time 0.4 and 0.56 seconds

In Figure 2.56 the time history of the horizontal wave probe is shown, compared with the simulation. The velocity with which the water is flowing into the tank is predicted very well by the simulation. From the measurement it seems that the probe is not completely covered with water from time 0.5 to 2 seconds, but this is not observed in the video.

In Figure 2.57 time histories of the water height at two locations are shown: in the reservoir, and in the tank just in front of the box. The agreement in both pictures is very

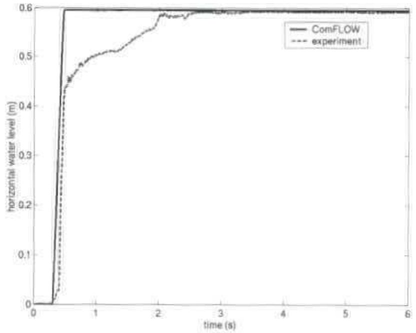


Figure 2.56: Horizontal wave probe along the side of the tank

good until the water has returned from the back wall (after about 1.8 seconds). After that some differences occur, but the global behaviour is still the same. After the water has returned from the wall, the fluid height at probe H2 is the largest. The water flows back to the reservoir, where it turns over again after about 4 seconds. The moment that this second wave meets the height probe at H2 again (after about 5 seconds) is almost exactly the same in simulation and experiment.

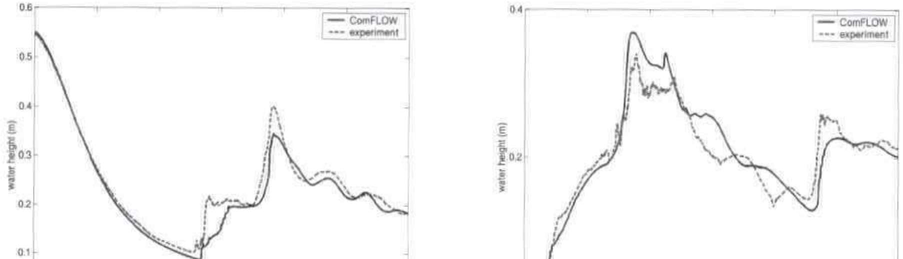


Figure 2.57: Vertical water heights in the reservoir H4 (left) and the tank H2 (right)

The instant when the wave hits the box is perfectly captured by the simulation as can be seen from Figure 2.58. Here the pressure at point P1 and P3 at the front of the box and at the top of the box, P5 and P7 (see Figure 2.54), are shown. The magnitude of the impact pressure is the same for simulation and experiment at pressure point P1 (the lowest on the box), but is underpredicted by the simulation at point P3. The moment the return wave hits the box again (at about 4.7 seconds) is again visible in the graphs. In the bottom graphs of Figure 2.58, where the time history of pressure transducers at the top of the box are shown, a difference is observed between simulation and experiment. After about 1.3 seconds, there is a wiggle in the simulation with a duration of 0.5 seconds,

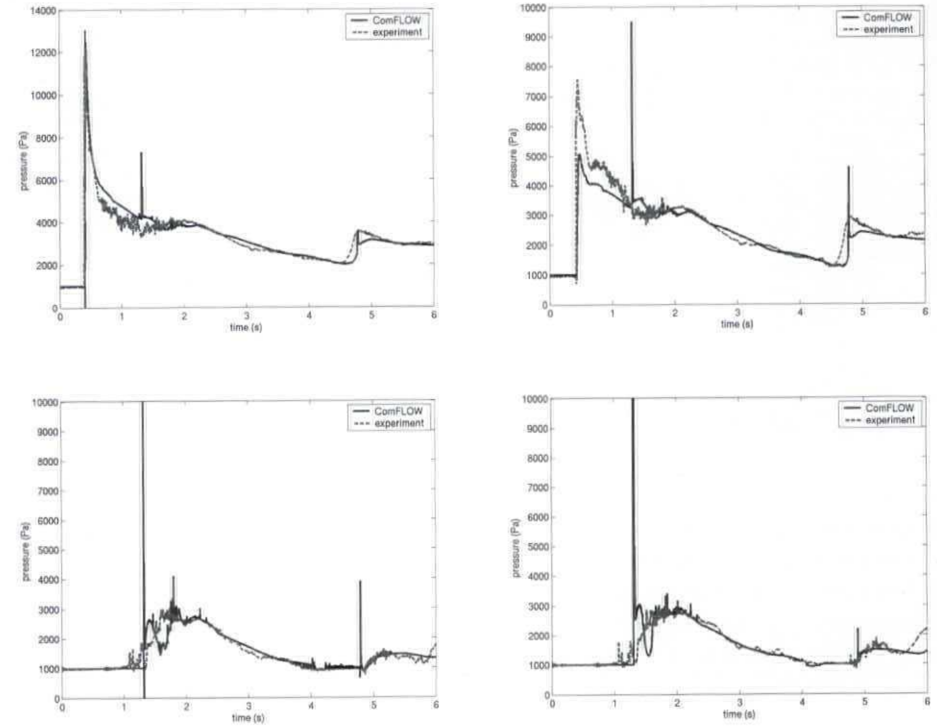


Figure 2.58: Pressure time histories at P1 (picture upper left), P3 (upper right), P5 (lower left), and P7 (lower right)

which is not present in the experiment. Before this point the water hits the top of the box when the wave coming back from the wall is overturning.

Several spikes appear in the pressure signals that are visible in all graphs at the same moment (for example at 1.3 seconds). These spikes occur, because some water enters an empty cell that is completely surrounded by cells with fluid. When the water enters the E-cell, there is no empty cell left in the neighbourhood, so this cell changes to a fluid cell in one time step without being a surface cell in between. This discontinuous change in label and the corresponding restoration of $\nabla \cdot \boldsymbol{u} = 0$, results in a pressure peak over the whole pressure field.

In Figure 2.59 a grid refinement study of the dambreak simulation is shown. Three different grids have been used with in increasing order 59×19×17 grid points, 118×38×34 grid points and 236×76×68 grid points. The finest grid has also been used in the previous figures. In the figure the pressure along the lower part of the front side of the box is shown. The overall flow of the water is pretty much the same in all three grids, but when zooming in on the pressure peak (in the right of the figure) differences occur. The coarsest grid is clearly not good enough. The pressure peak is overpredicted and the water reaches the box too late. Although the water reaches the box earlier in the finer grids, there is still a small difference between simulation and experiment. The magnitude of the impact is better predicted on the finer grids.

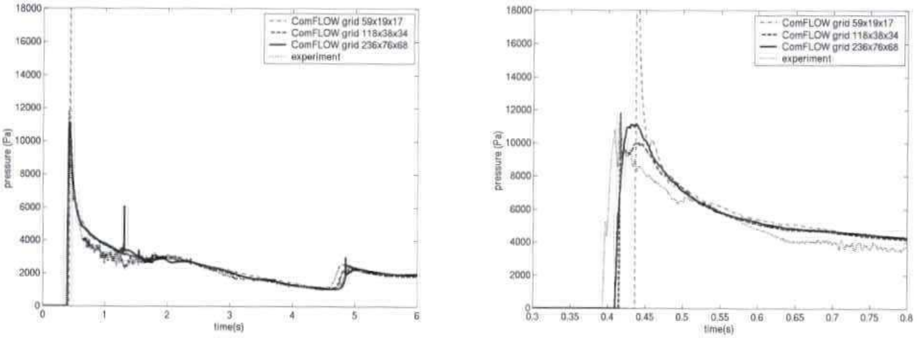


Figure 2.59: Grid refinement in the dambreak simulation: pressure at the lower part of the box, with on the right a zoomed picture

The results of the dambreak simulation are in good agreement with the experiment. The global behaviour of the fluid is the same and the impact peak of the pressure agrees well, especially at the lower part of the box.

2.7 Validation: water entry

In this section, results are presented from water entry of two-dimensional wedges, circular cylinders and of a three-dimensional cone. In [23] already some cases of water entry have been shown, also with free falling objects. The tests in this thesis have been performed with prescribed constant entry velocities.

2.7.1 Wedge entry

Figure 2.60 presents free surface profiles for the entry of two wedges. The wedges have deadrise angles of 30 and 45 degrees, respectively. The simulation results are compared with photographs of experiments by Greenhow and Lin [37]. The visual comparison between experiments and simulations is relatively good. However, not all the droplets and small details are resolved by the simulation, because of the displacement algorithm and the grid resolution. The simulations have been performed on a grid of 300×280 computational cells. Using such a fine grid, the simulation method is able to resolve the jets at the side of the wedges. The angle under which the jets are formed and bend away is well predicted in the simulation. However, these jets do not have a large influence on the impact loads during the penetration [36]. The flow is not perfectly symmetric, because the numerical algorithm is not symmetric (for example the marching direction in the iteration process of the Poisson equation has some influence on the symmetry).

In Fig. 2.61 the results of a grid refinement have been shown. Three different grids have been used: 75×70, 150×140 and 300×280 grid cells. The large difference between the results of the free surface profiles is in the formation of the jets: the finer the grid, the better the jets are resolved.

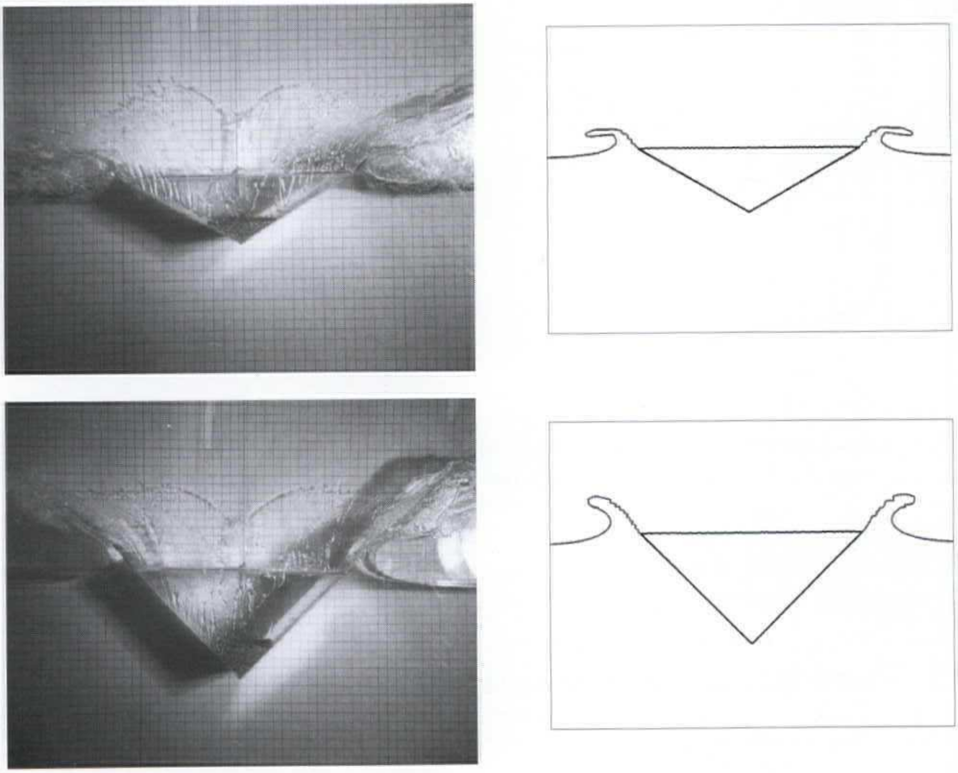


Figure 2.60: Snapshots of wedge entry with deadrise angles 30 degrees (up) and 45 degrees (down), experimental photographs of Greenhow and Lin, 1983

2.7.2 Cone entry

As the three-dimensional equivalent of the wedge, the entry of a cone has also been studied. The cone has a deadrise angle of $\beta = 30$ degrees. The slamming coefficient, as a non-dimensional measure of the total hydrodynamic force, has been compared with the theoretical prediction by Schiffman and Spencer made in 1951 [82]:

$$\frac{F}{\rho} = 3k(\beta)\tan^3(\frac{\pi}{2} - \beta)V^4t^2,$$

where $t = 0$ is the moment the cone hits the free surface first. The non-dimensional parameter parameter $k(\beta)$ is considered most accurate at value 1.6 (see [3]). In the top of Figure 2.62 the slamming coefficient as calculated in the simulation has been plotted against the penetration depth Vt. Three different entry velocities have been chosen, leading to the same results in a suitable set of scaled variables.

The results of the slamming coefficients for the impact during the entry of a cone are in very good agreement with the theory of Schiffman and Spencer [82]. In the bottom of Figure 2.62 a cross-section is shown of the free surface profile during the cone entry. From

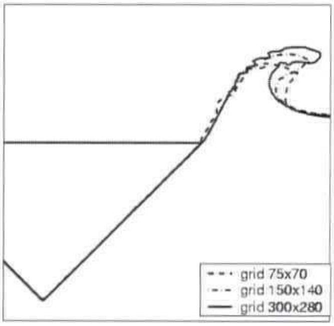


Figure 2.61: Free-surface profile of a falling wedge simulation with dead-rise angle 45 degrees using three different grids

this it can be seen that the jets at the side of the cone, which are clearly present in the entry of a wedge (see Figure 2.60) are not well resolved. This is caused by the difference in two and three dimensions and due to the much coarser grid used in the cross-section of the cone entry. However, it does not have a large influence on the total slamming force, because pressures in the jets are very small and do not contribute much.

2.7.3 Entry of a circular cylinder

The entry of a two-dimensional circular cylinder has also been studied. Snapshots of two different instants in time are shown in Figure 2.63 and are compared with photographs of experiments by Greenhow and Lin [37]. The free-surface shape observed in the experiment is very well resolved by the simulation method. Not all the details of the droplets of the splash are captured by the simulation, but the jets that appear at the sides of the cylinder are well predicted.

The total vertical hydrodynamic force on the cylinder during the first stage of the impact has been calculated and compared with experimental results of Campbell and Weynberg [10], also reported in [3]. In Figure 2.64 the slamming coefficients of the cylinder entry with different entry velocities have been plotted versus the non-dimensional penetration depth. The slamming coefficient is given by $Cs = F/\rho R V^2$, with F the total vertical hydrodynamic force, R the radius of the circular cylinder and V the entry velocity. Besides the experimental result of Campbell and Weynberg, also the theory of Von Karman (1929), reported by Faltinsen in [22], has been included. This theory is based on potential flow theory. For the very initial state of the entry of a circular cylinder, the hydrodynamic slamming force can be estimated by

$$F = V\rho\frac{\pi}{2}(2VR - 2V^2t),$$

where t denotes time with $t = 0$ the moment of first impact.

The comparison between the experiments of Campbell and Weynberg and the simulations is relatively good. It can be seen that the initial impact is a bit underpredicted in

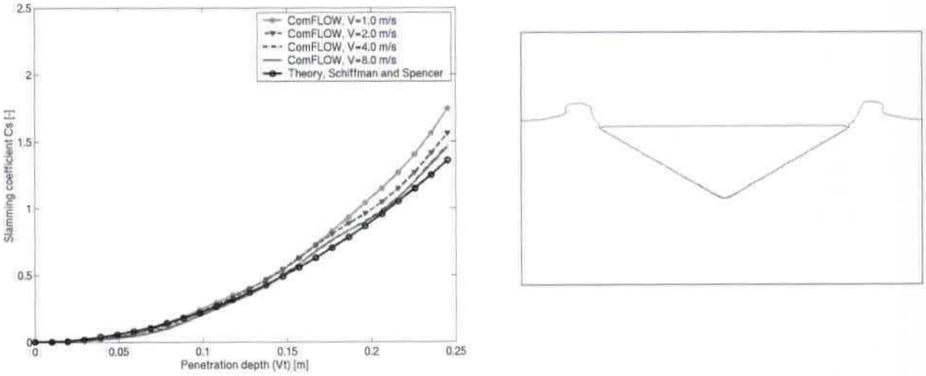


Figure 2.62: Left: slamming coefficient for the impact of a cone, simulation compared with theory of Schiffman and Spencer (1951); right: cross-section during the entry of a cone showing the free surface profile

the simulations for all entry velocities. The initial impact is more in agreement with the theory of Von Karman. In a later stage the results are in very good agreement with the experiments of Campbell and Weynberg. The simulation results are similar for different entry velocities, confirming near-perfect scaling with V^2.

To investigate convergence of the method under grid refinement, the circular cylinder entry simulations have also been run with different grids. The results are presented in Figure 2.65. It can be seen that the coarseness of the grid has a very large influence on the formation of the jets aside the cylinder. This was also concluded from the simulation of the cone entry. A very fine grid is needed to capture the jets. However, the formation of the jets does not have a large influence on the total hydrodynamic force. The force can be predicted quite accurately on the coarsest grid.

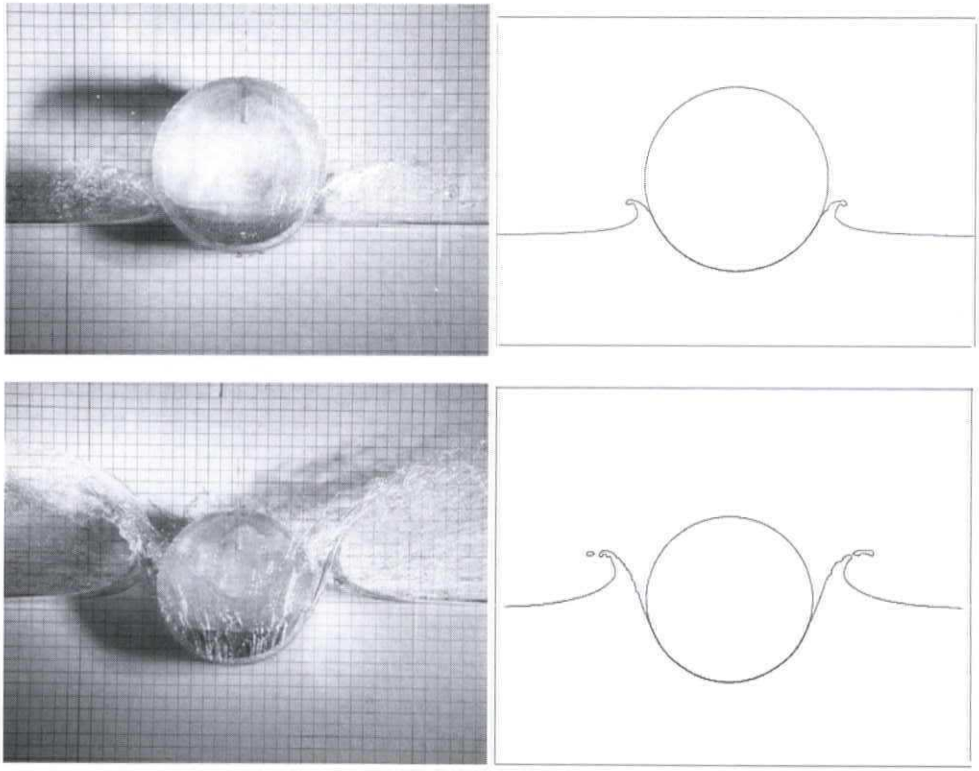


Figure 2.63: Snapshots of cylinder entry, experimental photographs of Greenhow and Lin, 1983

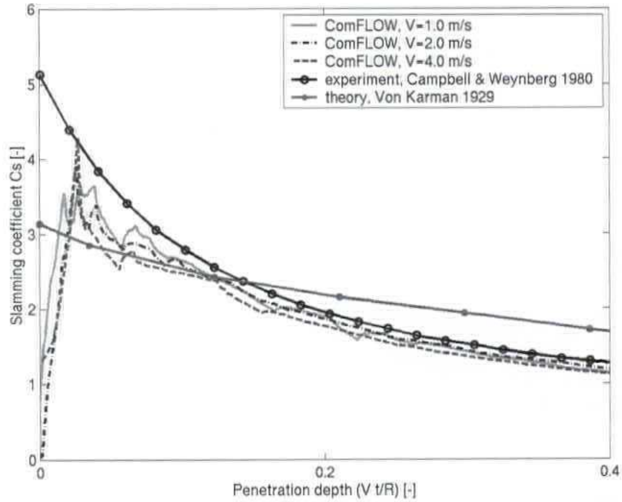


Figure 2.64: Slamming coefficient of the entry of a circular cylinder compared to the experiments of Campbell & Weynberg and the theory of Von Karman

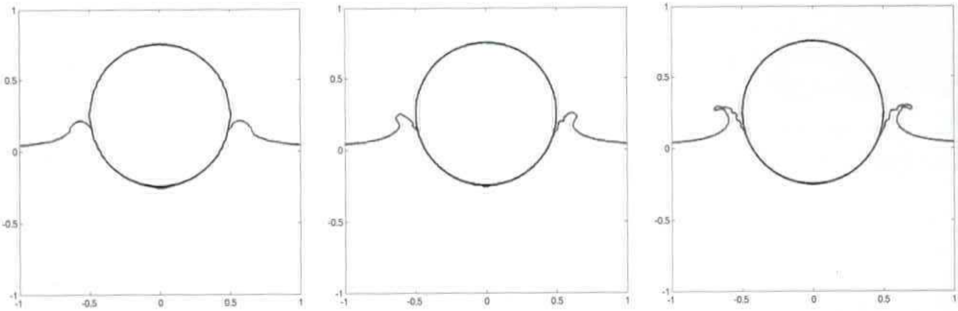


Figure 2.65: Effect of grid refinement on the jets formed during water entry of a circular cylinder: from left to right 100 × 100, 200 × 200 and 400 × 400 grid cells

Chapter 3

Wave generation and propagation

3.1 Introduction

To perform realistic simulations of green water and bow impact loading, the method has to be extended with wave generation options. There are different ways to create a wave field in a computational program:

- Prescribe a wave at the inflow boundary using theory, for example Stokes theory or a superposition of linear waves.
- Use a wave maker, which can be modelled as a moving object.
- Use another calculation method, which calculates kinematics away from an object that will be prescribed at the domain boundaries.

In this chapter the implementation and validation of the first option will be described, whereas the third option is used in the next chapter. The waves considered are long crested waves, so the incoming wave is two-dimensional of shape. Of course, when an object is present, the wave will be disturbed and three-dimensional aspects become important. At the boundary of the domains appropriate conditions are needed to let the wave travel in and out of the domain in an undisturbed way. Therefore, at the inflow boundary fluid wave kinematics will be prescribed according to wave description theories. ? The undisturbed wave will be prescribed, so the disturbance due to wave diffraction on an object is not taken into account. At the outflow boundary conditions should be imposed, such that the wave can leave the domain undisturbedly. Determining these conditions is ? difficult, because no information about the wave is present near the outflow boundary.

After describing the wave theories and conditions at the outflow boundary, attention will be paid to possible energy dissipation due to the upwind discretisation of the convective terms. Also the influence of the treatment of the free surface on wave propagation, like the choice for free surface velocities and the displacement algorithm, is investigated.

To validate the numerical model, simulations have been performed, which can be checked against theory or experiments. First, results will be shown of two-dimensional wave simulations without the presence of an object. Further, a spar platform has been put into the flow. Forces and wave heights have been calculated, which can be compared with experimental results. Finally, as the most demanding test case, green water on the bow of a moving FPSO is simulated.

[illegible]

3.2 Wave definition

At the inflow boundary the wave will be initiated using a theoretical wave description. In this thesis the wave is usually travelling in positive x-direction, making the left domain wall the inflow boundary (see Figure 3.1). The wave is prescribed as head wave, but of course by rotating the body in the domain every wave direction can be achieved. For the simulation of a wave in the domain, the velocities and wave height given by wave theory are prescribed. The waves that can be formed in ComFLOW are based on Airy wave theory describing linear waves, and 5th order Stokes theory describing nonlinear waves. Also a superposition of linear wave components can be used to generate an irregular wave.

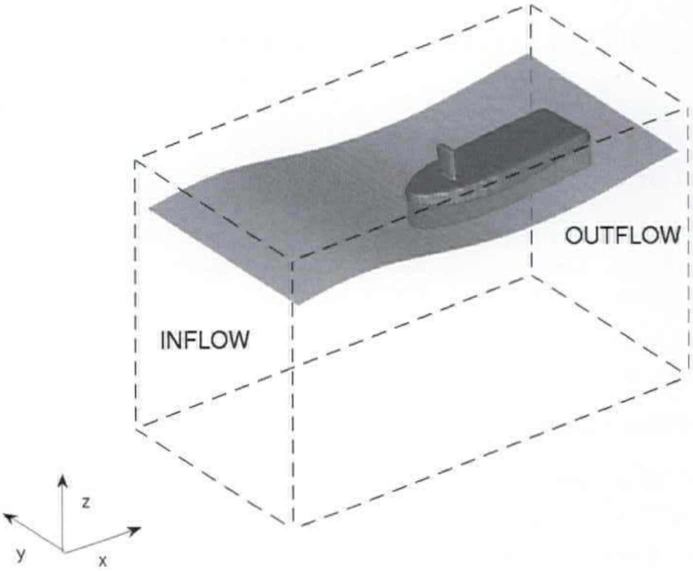


Figure 3.1: Coordinate system used in the simulation: the wave travels in positive x-direction

3.2.1 Airy wave theory

The theory for the generation of a linear wave can be derived in the following way. Mass conservation in a volume filled with an incompressible fluid leads to the continuity equation

$$\nabla \cdot \boldsymbol{u} = 0,$$

with $\boldsymbol{u} = (u, w)$ the velocity vector in two dimensions. The velocity potential Φ is defined by the equations

$$\frac{\partial \Phi}{\partial x} = u, \ \frac{\partial \Phi}{\partial z} = w, \qquad (3.1)$$

with x and z the horizontal and vertical coordinate, respectively. Substituting Φ into the continuity equation leads to a Laplace equation for the velocity potential:

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

When assuming the water surface slope very small, the potential is written as a sine with an amplitude depending on the water depth

$$\Phi(x, z, t) = P(z)\sin(\omega t - kx + \phi), \tag{3.2}$$

where t is the time, k the wave number, and ϕ the phase angle. $P(z)$ can be determined by substituting Equation (3.2) into the Laplace equation and solving the differential equation for $P(z)$, which finally results in

$$\Phi(x, z, t) = (C_1 e^{kz} + C_2 e^{-kz})\sin(\omega t - kx + \phi). \tag{3.3}$$

The constants C_1 and C_2 can be determined using the boundary conditions at the sea bed and the free surface. At the sea bed a no-leak condition holds, which is given by

$$\frac{\partial\Phi}{\partial z} = 0 \qquad \text{at } z = -h.$$

Substituting this boundary condition into Equation (3.3) reduces the two unknowns to one:

$$\Phi(x, z, t) = C\cosh(k(z+h))\sin(\omega t - kx + \phi). \tag{3.4}$$

The unknown constant C can be determined by using the free surface dynamic boundary condition that is deduced from the Bernoulli equation, taking into account the small wave steepness. The condition at the free surface is linearised around the calm water level ($z = 0$), which leads to

$$\frac{\partial\Phi}{\partial t} + g\zeta = 0 \qquad \text{for } z = 0.$$

The free surface elevation ζ can be derived from this equation after substitution of the velocity potential, resulting in

$$\zeta(x, t) = \zeta_a\cos(\omega t - kx) \quad \text{with} \quad \zeta_a = -\frac{\omega}{g}C\cosh(kh).$$

Rewriting the expression for ζ_a, the constant C can be determined as $C = -\frac{\zeta_a g}{\omega}\frac{1}{\cosh(kh)}$. Substituting this into Equation (3.4) leads to the final expression for the velocity potential

$$\Phi(x, z, t) = -\frac{\zeta_a g}{\omega}\frac{\cosh(k(z+h))}{\cosh(kh)}\sin(\omega t - kx + \phi). \tag{3.5}$$

The linearised free surface kinematic boundary condition given by

$$\frac{\partial z}{\partial t} + \frac{1}{g}\frac{\partial^2\Phi}{\partial t^2} = 0 \quad \text{for } z = 0$$

leads to the dispersion relation, which connects the wave number and frequency in the following way:

$$k \tanh(kh) = \frac{\omega^2}{g}.$$

For deep water, the dispersion relation reduces to $k = \omega^2/g$.

The velocity vector $\boldsymbol{u} = (u, w)$ can now be determined using the definition of the velocity potential given in Equation (3.1)

$$u(t, x, z) = \frac{\zeta_a g k}{\omega} \cos(\omega t - kx + \phi) \frac{\cosh(k(z+h))}{\cosh(kh)}, \tag{3.6}$$

$$w(t, x, z) = -\frac{\zeta_a g k}{\omega} \sin(\omega t - kx + \phi) \frac{\sinh(k(z+h))}{\cosh(kh)}. \tag{3.7}$$

As stated before, linear theory can only be applied to very low waves. According to Le Méhauté [61], the range of suitability of linear theory in deep water is $H/\lambda < 0.0062$, with H the wave height and λ the wavelength.

Linear wave theory can also be used to generate irregular seas, because the superposition principle can be applied. Given vectors of frequencies, amplitudes and phases, a linear wave can be built as the sum of the individual components.

3.2.2 Wave kinematics above the calm water level

Since linear wave theory is only valid up to the calm water level and velocities are needed up to the free surface, some kind of stretching technique has to be used above the calm water level. In this method Wheeler stretching has been used as a commonly accepted method, which is easy to implement. In the Wheeler stretching technique the negative z-axis has been extended from the actual instantaneous free surface elevation to the sea bed. This has been done by replacing z in the right-hand-side of Equations (3.6) and (3.7) by

$$z' = \frac{h}{h+\zeta}\, z + h\, (\frac{h}{h+\zeta} - 1)$$

$z = \zeta \Rightarrow z' = 0$
$z = -h \Rightarrow z' = -h$

where

- z' is the computational vertical coordinate $-h \leq z' \leq 0$;
- z is the actual vertical coordinate $-h \leq z \leq \zeta$.

3.2.3 5th order Stokes theory

For steeper waves, where in general the crests become higher and the troughs flatter, linear theory does no longer hold. To describe nonlinear waves, a solution to potential theory is used. The solution is represented by Fourier series, and the coefficients in these series can be written as perturbation expansions with parameter Ak. Here, k is the wave number, which can be written in terms of the wave length $k = 2\pi/\lambda$, and A is the amplitude of the wave at lowest order. The terms in the perturbation expansion can be found by satisfying boundary conditions on the free surface, and solving the resulting set of ordered equations. The expansion of the series can be done, in theory, infinitely far, but in practice, the 5th order solution is already very complicated.

Details about how to implement the 5th order Stokes solution can be found in [83], where the sign correction described by [25] should be taken into account.

3.3 Treatment of open boundaries

Domain walls in wave simulations at open sea need to permit fluid flow in and out. Therefore, as pictured in Figure 3.1, at least an inflow boundary is needed where the wave is generated, and an outflow boundary opposite to the inflow boundary where the wave leaves the domain. In the current method the waves, which are long crested, usually travel in the positive x-direction. Because of the two-dimensional shape, the side boundaries can often be closed walls; when positioned far enough from the structure they do not influence the wave field in the close surroundings of the structure. When the waves are highly distorted, such that side walls are severely influencing the flow, also outflow boundary conditions can be used at the side walls.

In Figure 3.2 a cross-section of one row of cells is shown with the positions of the velocities at the inflow and outflow boundaries. At the inflow boundary the velocities and VOF function in the column of cells directly left of the domain boundary are prescribed (a Dirichlet boundary condition). This way, the horizontal inflow velocity is positioned at the domain boundary, and no pressure is needed in the inflow cells. For the outflow boundary the horizontal velocity is shifted to one cell width away of the actual domain boundary. Therefore, also a pressure is needed in the outflow cells, so velocities, pressure and VOF function need to be determined at the outflow boundary (using a Neumann-type outflow condition). In the previous section the wave descriptions have been given, from which velocities and wave height at the inflow boundary are derived. The remainder of this section deals with the outflow boundary.

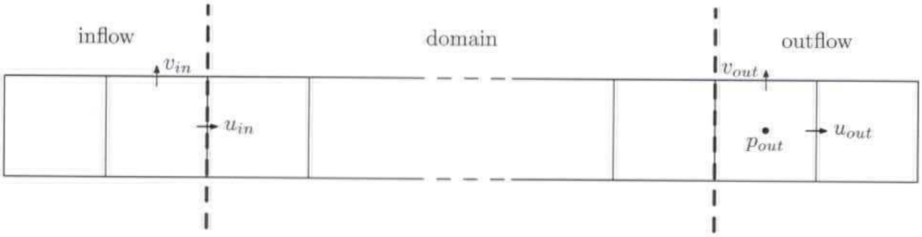


Figure 3.2: Position of velocities at the inflow boundary, and velocities and pressure at the outflow boundary

3.3.1 Overview outflow boundary conditions

A very important aspect of wave simulation is to determine the conditions at the outflow boundaries. If the wave is developing inside the domain, it should flow out of the domain as if there was no boundary. Otherwise, the wave will reflect from the boundaries into the domain, which disturbs the simulation.

An overview of outflow boundary conditions that prevent wave reflections is given by Givoli [30]. He describes three classes of outflow boundary conditions.

- The first class consists of special procedures for the numerical solution of wave problems in unbounded domains, that involve an artificial boundary but not the direct use of a non-reflecting boundary condition. Cerjan et al. [11] and others presented what can be termed a 'filtering scheme'. In this scheme, the amplitudes of the displacements are gradually reduced in a strip of nodes adjacent to the boundary. Thus, the solution is artificially damped in the vicinity of the boundary. Another kind of dissipation zone was used by [95], where an extra damping pressure was added to the free surface, which opposes the vertical wave velocity. A disadvantage of such damping zones is the increase of computational cells out of which the damping zone exists. Especially, in three dimensions many computational cells have to be added outside the real computational domain.

- In the second class, which is the largest one, local non-reflecting boundary conditions (NRBCs) based on the scalar wave equation are used. There exist many variations in local NRBCs, every problem in different types of fields (acoustics, gas dynamics, hydrodynamics, ...) uses NRBCs that works best for that particular problem. A comparison of different boundary conditions derived from the discretisation of the multi-dimensional wave equation is given by [42].
  The widely used NRBC of Sommerfeld [84] is a discretisation of the one-dimensional scalar wave equation with an a priori chosen wave velocity. A variation hereof was introduced by Orlanski [71], who calculated the wave velocity for every grid point and used that velocity in the discretised wave equation. The advantage of this method is that no information is needed of the flow beforehand.

- Finally, non-reflecting boundary conditions are considered that are non-local in time or space or both. These NRBCs have the disadvantage that many time levels should be stored in memory.

In the following sections a discussion of the Sommerfeld and Orlanski boundary conditions is given. The implementation of a general non-reflecting boundary condition is described, and the damping zone used by [95] is introduced.

3.3.2 Non-reflecting boundary conditions

Boundary conditions based on the wave equation

One way to prevent waves reflecting from the outflow boundary is to use a non-reflecting boundary condition. In case of waves most of the existing boundary conditions are based on the wave equation [42]

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = 0, \tag{3.8}$$

where ϕ is any quantity that travels wave-like as velocity or pressure, and c the wave velocity. In Equation (3.8) the direction of the wave is in the x-direction. In the Sommerfeld boundary condition [84] the wave velocity is chosen a priori, which is easy when regular waves are simulated. In that case, the Sommerfeld condition gives very good results as will be shown later. When irregular waves are considered, or when the regular waves are deformed due to the presence of an object, the method does not give very accurate results, since only one wave velocity can be chosen. For those situations, Orlanski [71] developed a method where the wave velocity is not chosen a priori, but is calculated every time step based on the local wave kinematics near the outflow boundary. The problem with this method is to determine the wave velocity accurately. To calculate c, first the wave equation is discretised using finite differences in cells close to the outflow boundary where the condition will be imposed. The wave equation is discretised for the horizontal velocity in every level of cells in the water depth. The c is then calculated as the mean of the approximated wave velocities in every level in the water depth. In Figure 3.3 the calculated wave velocity is shown for a regular wave with an actual wave velocity of 22.6 m/s during one wave period. The calculated wave velocity is not constant at all, but oscillates around the theoretical value. The jump that occurs around 7.2 s is present because there the crest of the wave travels through the outflow boundary resulting in $\partial u/\partial x = 0$. This can cause the wave velocity to jump from minus infinity to plus infinity. In this region, the wave velocity is adapted, such that these extreme values are not used in the outflow boundary conditions. In an investigation of wave propagation using Orlanski's method at the outflow boundary, it turned out that the results were not very accurate, so this method is not used in the simulations shown in this thesis.

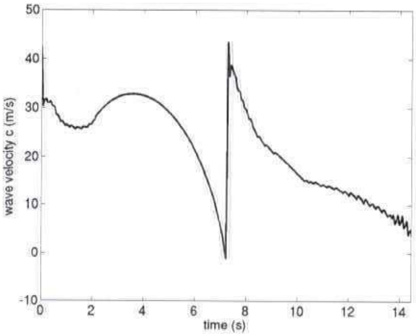


Figure 3.3: Calculated wave velocity using a finite difference approximation of the wave equation

The implementation of a non-reflecting boundary condition

At the outflow boundary conditions for pressure and velocity are needed. When using a Sommerfeld boundary condition, for the velocities at the outflow boundary, the wave equation Equation (3.8) is discretised using finite differences in the following way

$$\frac{\phi_e^{n+1} - \phi_e^n}{\delta t} + c\frac{\phi_e^n - \phi_c^n}{\delta x_e} = 0,$$

where the notation is explained in Figure 3.4 and ϕ is used for the velocity components. From this equation the velocities at the outflow boundary u_e^{n+1}, v_e^{n+1} and w_e^{n+1} follow.

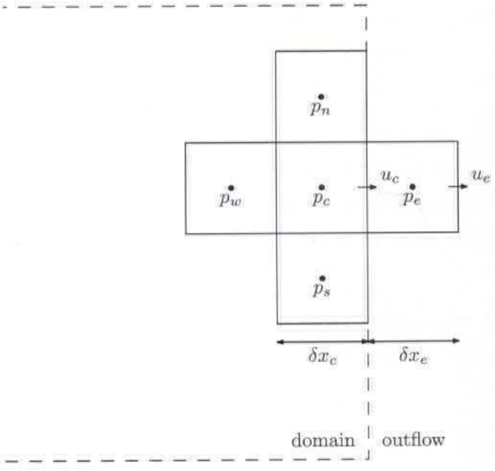


Figure 3.4: Configuration of cells near the outflow boundary

For the calculation of the pressure in the outflow cell the procedure is somewhat more complicated, since the pressure appears in the Poisson equation that is solved using an iterative method. To avoid an iteration in the outflow cell, the boundary condition is substituted in the equation for the interior pressure cell p_e, such that the equation for the pressure in the outflow cell is not needed. At the end of the iterations the pressure in the outflow cell is updated using the pressure in the interior cell. This procedure is elaborated for a general outflow boundary condition

$$\alpha\frac{\partial p}{\partial x} + \beta p = \alpha A + \beta p_0.$$

Here, the Neumann condition can be recognised for $(\alpha, \beta) = (1, 0)$ and the Dirichlet condition for $(\alpha, \beta) = (0, 1)$. The Sommerfeld condition is obtained when taking $\alpha = c$, $\beta = 1/\delta t$, $A = 0$ and $p_0 = p_e^n$. This equation is discretised using finite differences

$$\alpha\frac{p_e - p_c}{\delta x_{pe}} + \beta p_e = \alpha A + \beta p_0,$$

with $\delta x_{pe} = (\delta x_e + \delta x_c)/2$. From this equation the pressure in the outflow cell p_e can be solved

$$p_e = \frac{\alpha}{\alpha + \beta\delta x_{pe}} p_c + \frac{\alpha A\delta x_{pe} + \beta p_0 \delta x_{pe}}{\alpha + \beta\delta x_{pe}}. \tag{3.9}$$

The discretised Poisson equation for the interior pressure p_c can be written as (for notation see Figure 3.4)

$$C_c p_c + C_w p_w + C_e p_e + C_n p_n + C_s p_s = RHS.$$

The coefficients C_c to C_s contain squared grid sizes and the right-hand-side RHS contains the divergence of the contributions from convection, diffusion and external forces

(Equation (2.18)). In this equation the outflow pressure given by Equation (3.9) can be substituted resulting in

$$(C_c + \frac{\alpha}{\alpha + \beta \delta x_{pe}} C_e) p_c + C_w p_w + C_n p_n + C_s p_s = RHS - \frac{\alpha A \delta x_{pe} + \beta p_0 \delta x_{pe}}{\alpha + \beta \delta x_{pe}} C_e.$$

The matrix containing the coefficients remains diagonal dominant when $\alpha, \beta \geq 0$. After the pressure in the interior of the domain is solved from the Poisson equation, the pressure in the outflow cell is updated using Equation (3.9).

3.3.3 Dissipation zone: pressure damping at the free surface

Instead of using a non-reflecting boundary condition, a dissipation zone can be used where the wave is damped. In the current method, for the damping inside the dissipation zone a pressure term has been added to the free surface pressure. Physically, this can be interpreted as the air acting like a damper on the wave. The pressure added to the atmospheric pressure term at the free surface is chosen as a function of the vertical velocity at the free surface:

$$p_{damp}(t, x, \zeta) = \alpha(x)\ w(t, x, \zeta).$$

The damping function $\alpha(x)$ should be chosen such that the wave is damped completely, and the wave should not reflect at the start of the damping zone. A polynomial form for the function $\alpha(x)$ is used by [95], and [62] concludes in his report that a linear function is suitable for the purposes of wave simulation. If a linear damping function $\alpha(x) = ax + b$ is used, two constants a and b have to be chosen. Here, a is the slope of the damping function, which determines the rate of the damping. The constant b has to be chosen such that the damping function is zero at the start of the dissipation zone. The slope of the damping function is determined by the characteristics of the wave that is to be damped. Meskers [62] shows in his report how the slope and the length of the dissipation zone can be determined after choosing the total reflection that is allowed. In Figure 3.5 the result is shown. For a number of allowed reflection factors the slope and length of the dissipation zone are plotted. For regular waves, the following steps have to be taken to determine the slope and length of the dissipation zone. First, choose the amount of reflection (r_{tot}) that can be permitted. This amount of reflection is calculated theoretically, and is only valid for a perfectly regular wave [62]. Then, given the wave frequency and reflection coefficient, determine the length of the numerical beach using the curved lines. At the left coordinate axis, the accompanying values are placed. Finally, determine the slope of the numerical beach, using the straight lines with the values of the right coordinate axis.

Besides the choice of the function $\alpha(x)$, there also are some different possibilities for the closing wall of the domain, by which we mean the wall opposite of the inflow boundary.

- In the first option the wave is damped completely, the closing wall of the domain is a solid wall, no water is flowing out. 1
- In the second option the wave is damped towards its analytical form. So a 'perfect' wave is formed at the end of the domain. The closing wall is an outflow boundary. 2

What is its "analytical form" ?

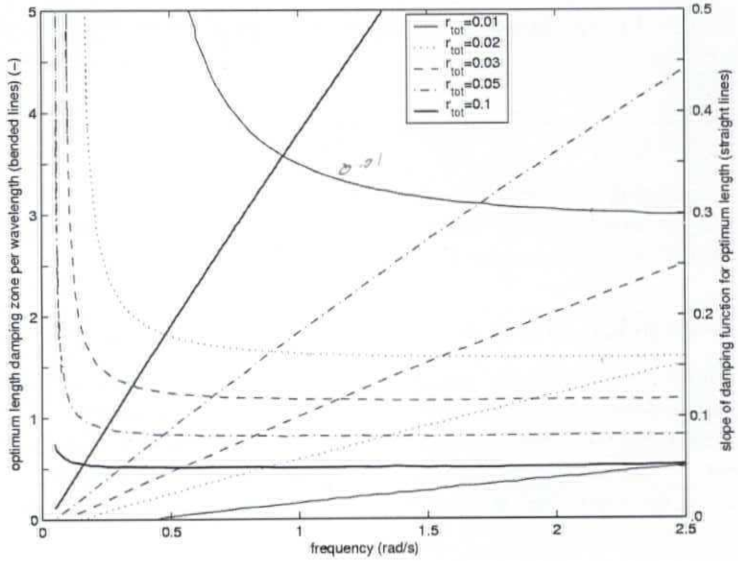


Figure 3.5: Combination of required length and slope of the dissipation zone for a certain allowed amount of reflection (from [62])

at which the analytical velocities are prescribed. This works very nicely in the case of wave propagation simulations without an object that disturbs the wave.

- In [95] and [62] a combination of the dissipation zone with a Sommerfeld boundary condition is used. In this case, the closing wall is an outflow boundary where a Sommerfeld boundary condition has been applied. The Sommerfeld boundary condition is tuned for the smallest frequencies of the wave in the domain, whereas the dissipation function is tuned for the larger frequencies.

In the current model we aim for the most general solution, so damping towards zero (the first option) is used. The wave is only damped in the travelling direction, whereas the tangential sides of the domain are solid walls, which have been placed at such a distance that the simulation results are not influenced by them.

When applying this to a rather high wave, some problems arise. The water level in the domain increases linearly in time, which is due to the fact that the net amount of water flowing into the domain at the inflow boundary is positive over one period and not zero, the so-called Stokes drift [22]. This is shown in Figure 3.6, where the results of a wave simulation are shown with a period of 14.44 s, a wave height of 32.6 m, a wavelength of 325 m and the water depth is 600 m. In the left of the figure the amount of water flowing into the domain is shown. Clearly, the integral over one period is not zero but positive. Because there is a solid wall at the end of the domain, the total amount of water is determined by the in- and outlet at the inflow boundary only, which causes the water level to increase linearly in time as is shown in the right of the figure. Because the wave is quite high, there is a very large increase of the water level, which is directly reflected

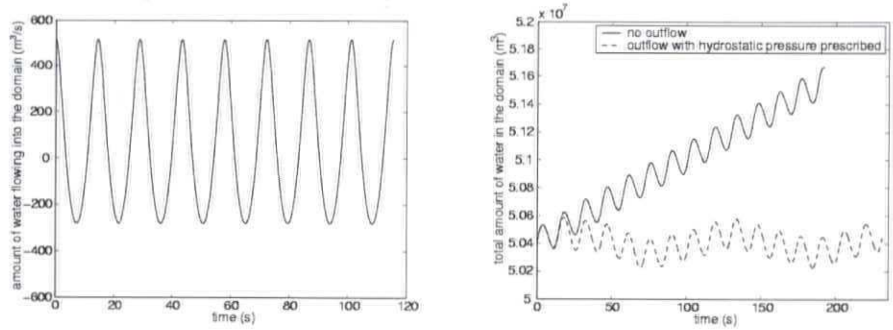


Figure 3.6: Left: Amount of water flowing into the domain; right: total amount of water in the domain

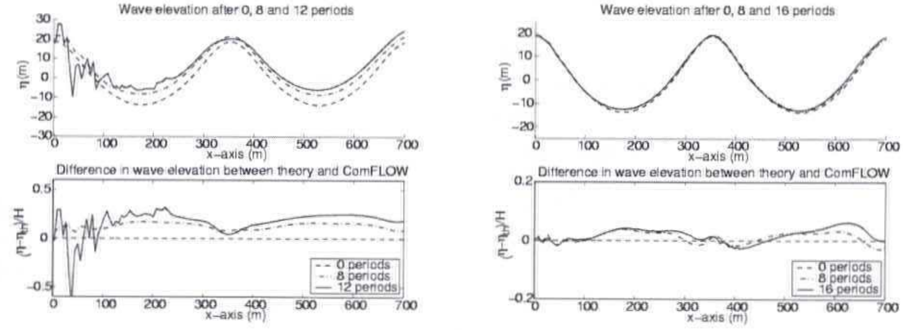


Figure 3.7: Wave elevation (top) and difference in wave elevation between theory and simulation (bottom) without outflow boundary (left) and with an outflow boundary where hydrostatic pressure is prescribed (right)

in the mean of the wave elevation as can be seen in the left of Figure 3.7. There are also some problems near the inflow boundary, where the wave is disturbed. Due to the reflections from the outflow boundary and the rise of the water level, the velocities and water height in the interior of the computational domain do not fit to the analytically prescribed velocities at the inflow boundary any more.

There are several ways to solve this increase of water problem. One way is to determine the net inflow and add an outflow boundary where this amount of fluid is forced out of the domain. A disadvantage of this method is, that it can only be applied to regular waves in a nice way, in which case it is known how much fluid should flow out of the domain at what time, due to the regularity.

A more flexible method has been found by changing the solid wall at the end of the domain into an outflow boundary, where hydrostatic pressure is prescribed. In a perfect situation, the water height at the end of the domain always equals the calm water level because of the damping of the wave. The fluid simulation will always try to maintain that level when the hydrostatic pressure for a water level equal to the water depth is prescribed at the outflow boundary. This method has been adopted in the dashed line in the right of Figure 3.6 and in the right of Figure 3.7. Instead of the linear increase of water, a fluctuation of the water volume around the initial amount of water can be observed. The fluctuation is due to some reflections, which can be concluded from the fact that the period of the fluctuation is 8 wave periods, which means that the reflected wave travels 8 wavelengths before it reaches the starting position, the outflow boundary, again. In a domain consisting of 4 wavelengths, this is exactly 8 wavelengths (back and forth in the domain). The increase and decrease of the total water level during 8 periods is less than 0.5%.

3.4 Free surface velocities

In Section 2.3.10 the treatment of the velocities in the neighbourhood of the free surface is explained. It was concluded that special care has to be taken for the determination of SE-velocities, which are velocities at the cell face between a surface cell and an empty cell. In Section 2.3.10 two methods are described, of which a combination is used in ComFLOW. In the first method SE-velocities are defined by demanding conservation of mass in an S-cell. This means that the total flux through the cell faces of the S-cell should be zero. One problem of this method, which is the origin of instabilities, has already been described in Section 2.3.10.

Another disadvantage of this method is the inaccuracy in wave simulations as will be shown here. This was already noticed by Chan et al. [13], who used an extrapolation of the velocity field instead. The inaccurate prediction of an SE-velocity in case of a wave simulation can be understood from Figure 3.8. In the right of the figure the horizontal velocity has been shown as function of the vertical coordinate. The theoretical values of the horizontal velocity in the neighbourhood of the free surface are indicated by the solid line. To satisfy div$(u) = 0$ in the central S-cell, the SE-velocity u_{SE} is copied from the left neighbour velocity (and the vertical velocity of this S-cell v_{SE} is copied from the lower cell face). Due to the coarseness of the grid, this neighbour velocity is about one or more meters left of the SE-velocity resulting in an inaccurate prediction of u_{SE} sketched by the dashed arrow in Figure 3.8.

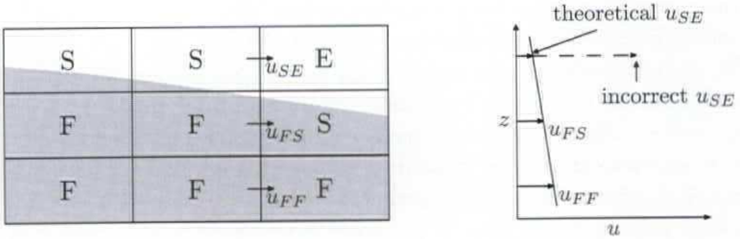


Figure 3.8: Inaccurate prediction of u_{SE} in a wave simulation when method 1 is used

In the second method for SE-velocities it is proposed to determine the velocities by choosing a direction, from which all SE-velocities in an S-cell are extrapolated. This direction is chosen as the direction where most fluid is present, so the coordinate direction that is 'most normal' to the free surface. In the case of waves, this is mostly the negative

z-direction. As stated before, constant and linear extrapolation can be used. In wave simulations linear extrapolation gives superior results, since then the velocities are estimated very accurately. The impact of the different methods for SE-velocities in a wave simulation can be seen in Figure 3.9. Here, an irregular steep wave propagates through an empty domain. In the figure a snapshot of the wave elevation is shown at the time point that the wave is most steep. The asterisks show measurements of this wave. The mass conservation method does not predict the wave elevation accurately for this steep wave. Both constant and linear extrapolation produce a much better resemblance with the measurements. The simulation using linear extrapolation is most accurate. Based on the simulation of waves in this section, the extrapolation method should be chosen for determining the SE-velocities. However, in Section 2.3.10 it was concluded that linear extrapolation cannot always be used for stability reasons. The method that is adopted here is a combination of linear and constant extrapolation as explained in Section 2.3.10.

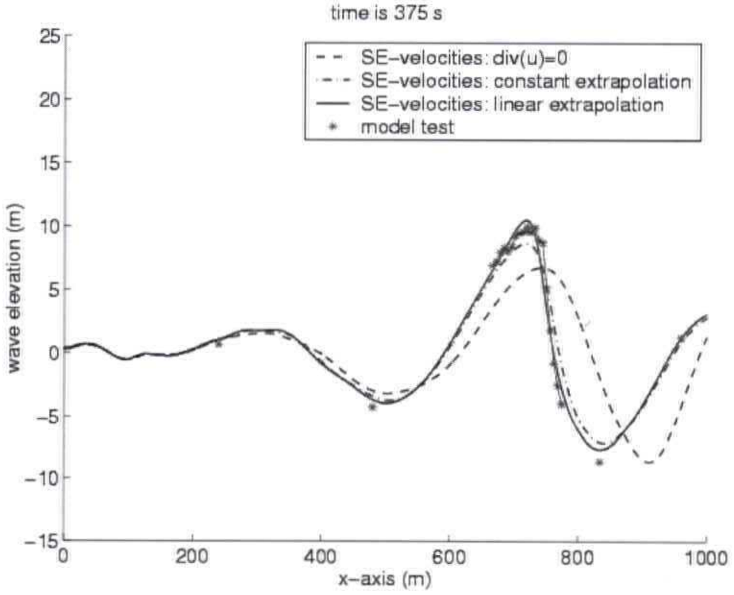


Figure 3.9: Wave elevation of an irregular wave where different SE-velocity treatments have been compared: mass conservation in S-cells, constant extrapolation and linear extrapolation

3.5 Wave propagation using different VOF methods

In Section 2.4 a few different VOF methods have been described and their performance has been tested using standard kinematic tests and a dambreak simulation. Four different methods have been used. First, the original Hirt-Nichols method has been used, where the interface is implicitly reconstructed in a piecewise constant manner. To avoid flotsam and jetsam and to take care of mass conservation, a local height function has been introduced. A more sophisticated method for the displacement of the free surface is the method of Youngs [98]. The free surface is reconstructed using piecewise linear elements and the displacement is based on this reconstruction. Youngs' method has also been used in combination with the local height function to ensure perfect mass conservation. In all simulations in this section, the extrapolation method for free surface velocities has been used.

First, the different VOF methods are tested on the propagation of a regular wave. The wave has a period of 14.44 seconds and a length of 325 meter. The wave height is 10.14 meter and the water has a depth of 600 meter. At the outflow boundary a Sommerfeld condition is used. Figure 3.10 shows the resulting free surface profile after a simulation time of four wave periods. Also the error in the calculated free surface profile compared to the theoretical solution is shown, calculated by

$$E_w(x) = |\eta(x) - \eta_{th}(x)|/H,$$

where η is the calculated wave elevation, η_{th} the theoretical wave elevation, and H the wave height. From the resulting error E_w, plotted in the bottom of Figure 3.10, it can be

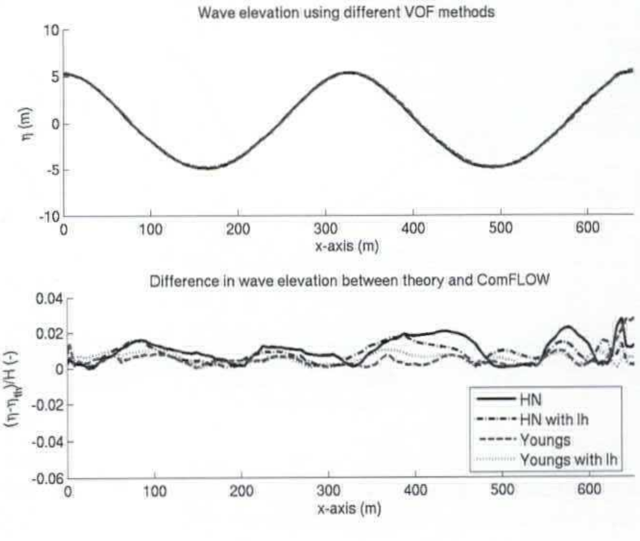


Figure 3.10: Wave elevation (up) and difference in wave elevation between Stokes 5th order theory and simulation (down) for different VOF methods: Hirt-Nichols and Youngs without and with a local height function

concluded that Youngs' method gives more accurate results in a regular wave simulation than Hirt-Nichols' method. Youngs without a local height function is best in the interior of the domain, but has a larger error at the outflow boundary than Youngs with a local height function. No reason has been found for this behaviour. In all four simulations relatively little mass was lost. When examining the total amount of water in an area of 30 m high around the calm water level, all four methods have mass loss within 0.5% of the amount of water in that area. The calculation times of the four simulations are comparable. The factor between the calculation times of Youngs and Hirt-Nichols is 1.2. Youngs method does not take much extra calculation time in this simulation, since the interface reconstruction is only performed in a small percentage of the total number of cells (in average in 130 of the 6000 cells a reconstruction is made).

The four different VOF methods are also used in the propagation of a steep wave event. The wave event is taken from an experiment at MARIN, to be more precise experiment 114002, which is described in Section 3.6.2. The wave is generated at the inflow boundary by prescribing a superposition of linear wave components derived from Fourier analysis of the measured wave elevation in the experiment. Figure 3.11 shows the wave elevation at the position and time point where the wave is high and steep. Youngs' method without a local height function results in the highest wave and gives the best agreement with the measurement. Hirt-Nichols with and without local height function give similar results, both are pretty good, but the wave crest is a bit flattened compared to the experiment. Youngs with a local height function performs worst, which is remarkable considering the good performance of this method in Section 2.4. The explanation for this result has not yet been discovered.

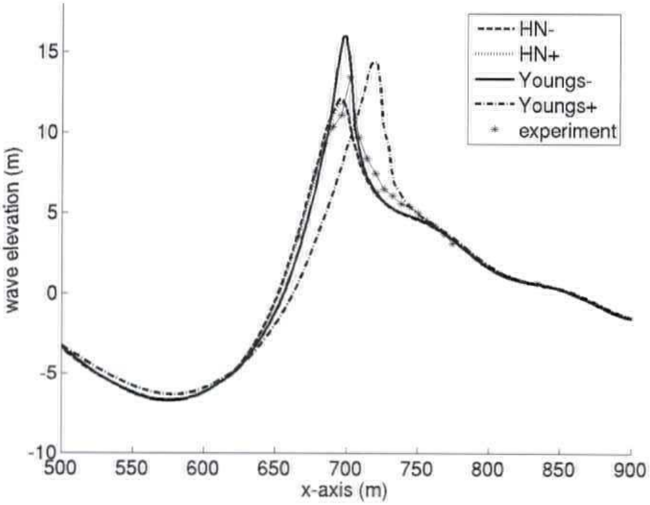


Figure 3.11: Wave elevation of a steep wave event using the four different VOF methods: Hirt-Nichols' VOF and Youngs' VOF with and without a local height function

To conclude, Youngs' method performs best in wave simulations, its results are most accurate. A problem arises when combining Youngs' method with a local height function in the steep wave event shown in this section. So, this combination should only be used with care until this problem is solved. Unfortunately, due to the choice of free surface velocities that are extrapolated, Youngs' method without local height function can cause loss of mass as shown in the dambreak simulation in Section 2.4.9. But in a simulation with a very smooth free surface as in the wave simulations, only a very small amount of water is lost using Youngs' method. So, for wave simulations, Youngs' method without local height function can be used without problem. But when performing a more violent simulation with a much distorted free surface, mass can be lost as shown in the section just mentioned. Mass can be perfectly conserved using Youngs' method, when it is combined with $\text{div}(u) = 0$ in surface cells. But using $\text{div}(u) = 0$ for the determination of free surface velocities results in inaccurate wave simulations as shown in Section 3.4 and can cause instabilities in the computations. Therefore, this option should not be used. The safest way is to use Hirt-Nichols' method with local height function, but this can be a bit less accurate.

In the remainder of this chapter Hirt-Nichols' method combined with the local height function is used for the displacement of the free surface.

3.6 Validation of wave propagation

For the validation of wave propagation in ComFLOW several tests have been performed. Firstly, two-dimensional wave propagation without an object in the flow has been investigated. Attention has been paid to reflections at the outflow boundary, the influence of the artificial viscosity present due to the upwind discretisation and the size of the grid and time step necessary for an accurate simulation of waves. Especially, steep and high waves, which are the most important waves in green water and wave impact calculations, have been studied.

Secondly, simulations have been performed of wave loading on a spar platform. The waves are regular, very long and quite low. The simulation results have been compared with experimental results that have been provided by the Maritime Research Institute Netherlands (MARIN).

3.6.1 Two-dimensional wave propagation: regular waves

A very extensive study of two-dimensional wave propagation without an object in the flow has been performed by Meskers [62]. Some of the most important results will be shown here also. In this section attention will be paid to the number of cells and the time step that are needed for an accurate description of the wave. The simulations have been performed and compared with both the linear and the 5th order Stokes theory. Also, the influence of the artificial viscosity has been studied by performing a simulation of many wavelengths during many periods. The next section is devoted to higher and steeper waves. These waves have been studied by simulating a design wave, which is a kind of mean shape of a wave in a linear random sea-state of which the power spectrum is given.

In this section a study of the necessary number of grid points and time steps for an accurate wave simulation is presented. Also a comparison of the use of a Sommerfeld boundary condition with a damping zone is made. The simulations have been performed for one example of a wave, which is one of the characteristic waves in an FPSO field with

Fig.	H (m)	# cells/λ	# cells/H	# dt/T	theory	outflow
3.12	20.28	60	3	250	Airy/Stokes5	damping
3.13	10.14	60	3	250	Stokes5	Sommerfeld
3.14	10.14	60	3	65/125/250/500	Stokes5	damping
3.15	10.14	30/60/90	3	250	Stokes5	damping
3.16	10.14	60	3/6/9	250	Stokes5	damping

Table 3.1: Characteristics of simulations run to investigate the propagation of two-dimensional regular waves

a water depth of 600 meter. This wave has a period of 14.44 seconds, resulting in a wave length of 325 meter. The wave height is varied between 10.14 and 20.28 meter. From the conclusions of Meskers in [62] we see that the characteristic parameter settings that have shown to give an accurate simulation of this example wave can also be used for deep water waves with different periods and wavelengths.

In the simulations of the wave shown in this thesis, a few numerical parameters have been varied, namely the number of time steps per period, the number of cells per wavelength, and the number of cells in the wave height. The results have been presented as the resulting wave elevation after a simulation time of four periods. The domain consists of four wavelengths in case of using a damping zone, of which two wavelengths are used as damping zone. The domain consists of two wavelengths when the Sommerfeld boundary condition is used.

In Table 3.1 an overview is given of the simulation characteristics, of which the results are shown in Figures 3.12 to 3.16.

In Figure 3.12 Airy wave theory and 5th order Stokes theory have been used for the initial condition and the inflow boundary condition. In the lower picture the difference between the simulation and linear theory and Stokes 5th order theory, respectively, has been shown. Clearly, the 5th order Stokes results are much better than the linear Airy wave results, which is consistent with the fact that H/λ equals 0.06, outside the validity region of linear theory [61].

Figure 3.13 shows results of two simulations with different outflow methods. First, the Sommerfeld outflow boundary condition is used that is based on the wave equation. Second, a damping zone is added as in the other simulations in this section. Both methods give similar results. The Sommerfeld boundary condition lets the wave flow out of the domain properly without much disturbance in the domain. The advantage of using the Sommerfeld condition over a damping zone is the number of grid cells that need to be used. In this simulation, the number of grid cells in the simulation using a damping zone is twice as large as the number of grid cells in the Sommerfeld simulation. On the other hand, the Sommerfeld condition can only be used in case of regular waves that are not too much disturbed. Further, the wave velocity should be given a priori, which is only possible when the wave characteristics are known.

In Figure 3.14 the number of time steps per period has been altered. From the figure it can be seen that more time steps per period results in a more accurate wave simulation. The difference between 250 and 500 time steps per period is not very large. From this it is concluded that 250 time steps per period is enough for an accurate simulation result.

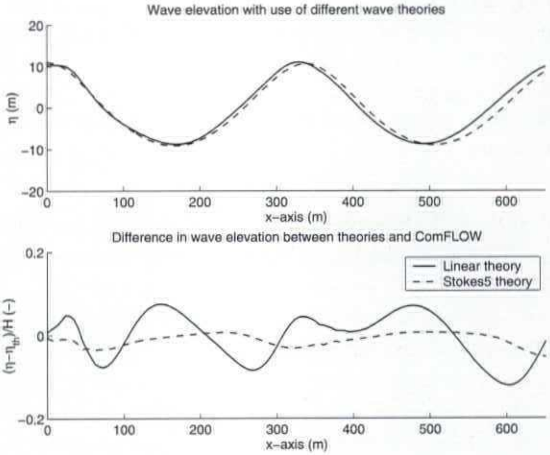


Figure 3.12: Wave elevation (top) and difference in wave elevation between theory and simulation (bottom) with initialisation using linear or Stokes 5th order theory

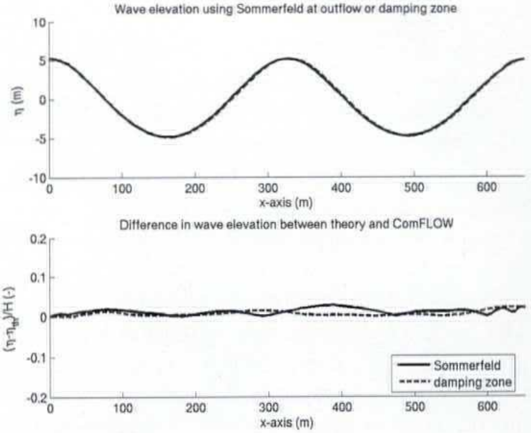


Figure 3.13: Wave elevation (top) and difference in wave elevation between theory and simulation (bottom) with initialisation using linear or Stokes 5th order theory

An investigation of the number of cells per wavelength has been summarised in Figure 3.15. The accuracy of the wave simulation increases with a larger number of cells per wavelength. There is no difference between 60 and 90 cells per wavelength, so 60 cells per wavelength is enough to capture the wave.

Figure 3.16 shows the results of altering the number of cells in the wave height. The grid was equally stretched towards the calm water surface in the three different simulations. There is not a large difference between the results. For all three simulations, the difference between computation and theory is about 5% of the wave height. In the case of 9 cells in the wave height, the simulation shows some small peaks, especially in the crest and trough of the wave. This is due to the ratio between dx and dz, which are the distances between two grid lines in x and z direction, respectively. Meskers gives in his report an estimate for the minimum required aspect ratio to prevent these wiggles, based on a series of simulations. In the future, this point should be investigated further to understand the nature of the wiggles.

Dissipation of energy in wave simulations

For the discretisation of the convective terms in the Navier-Stokes equations a first order upwind scheme has been used as explained in Section 2.3.5. Compared to a central discretisation the upwind discretisation can be interpreted as a central discretisation plus an extra diffusive term. This term adds extra viscosity to the physical viscosity μ/ρ. The amount of extra viscosity is equal to $uh/2$ where u is the velocity and h the mesh size. So this term is dependent on the position in space.

When simulating waves, we have to investigate the influence of the artificial diffusion on the wave propagation. From the results in the previous section it can be seen that the influence is definitely not very large over a few periods, because no real damping is visible after the four periods that have been simulated. To get a better idea of the influence of the artificial viscosity, a wave has been simulated for many periods in a domain of 25 wavelengths. The wave has a period of 1.9 seconds and a wave height of 0.16 meter.

In Figure 3.17 the wave elevation as function of time and as function of distance to the inflow boundary are shown. In the left of this figure the wave elevation as function of time at a position 14 wavelengths from the inflow boundary is shown. The simulation is started with an undisturbed (and thus undamped) wave field. During the simulation the amplitude of the wave decreases, until the stationary damping is reached after about 58 seconds. At this position, 14 wavelengths from the inflow boundary, the wave height has decreased from 0.16 meter to 0.125 meter, which is a decrease of 22% of the wave height. From the right picture it can be seen that the wave height has decreased 28% at 20 wavelengths behind the inflow boundary after a simulation time of 100 periods. An estimate of the maximum artificial viscosity that is added due to the upwind discretisation can be found by calculating $uh/2$. For the velocity u the maximum of the mean velocity over one period is taken, which according to linear theory is equal to $u = gAk/\omega \approx 0.26$ m/s. So the approximate maximum artificial viscosity is given by

$$k_{art} = uh/2 \approx 0.26 * 140/1500 \ / \ 2 = 0.012 \text{ m}^2/\text{s},$$

whereas the physical kinematic viscosity ν is equal to 10^{-6} m^2/s. To check whether the damping in Figure 3.17 is really caused by the artificial viscosity, simulations have

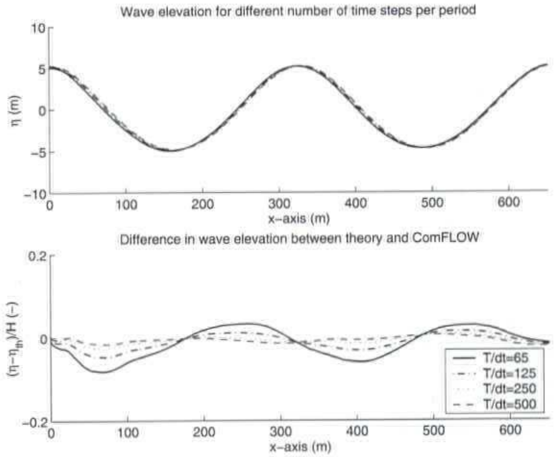


Figure 3.14: Wave elevation (top) and difference in wave elevation between theory and simulation (bottom) varying the number of time steps per period

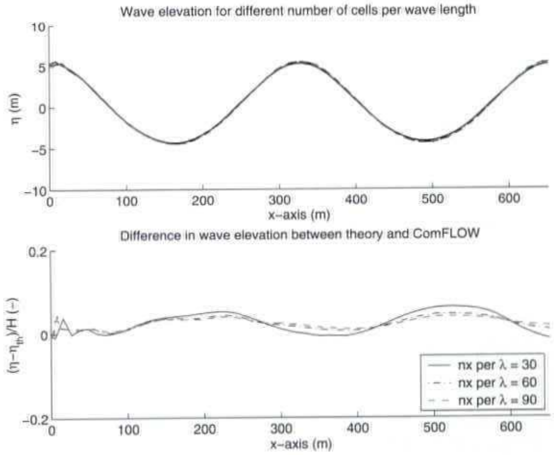


Figure 3.15: Wave elevation (top) and difference in wave elevation between Stokes 5th order theory and simulation (bottom) for different number of cells per wavelength

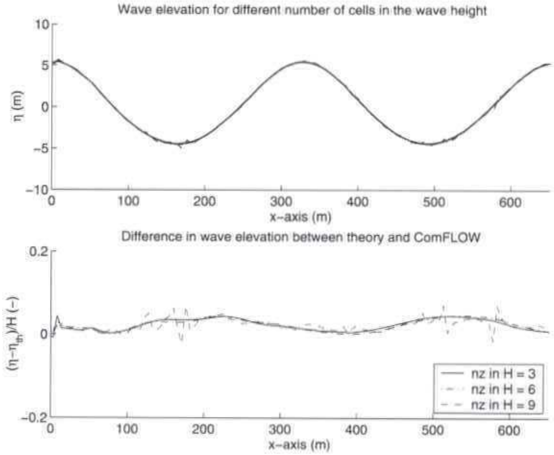


Figure 3.16: Wave elevation (top) and difference in wave elevation between Stokes 5th order theory and simulation (bottom) for different number of cells in the wave height

been performed with less artificial viscosity (only 10% of the artificial viscosity is added) and with a central discretisation. The simulation with the central discretisation has been performed for 58 periods in a domain of 12 wavelengths, which is because the grid sizes and time step have to be very small in a calculation with a central discretisation. In the left part of Figure 3.18 the wave elevation at 9 wavelengths behind the inflow boundary has been given for the different amounts of added viscosity. It can be clearly seen, that the amplitude of the wave is decreasing more in time when the added viscosity becomes larger. This is also shown in the right part of Figure 3.18, where the wave height decrease is shown after 100 periods of simulation at the different locations behind the inflow boundary. We can conclude from this figure that there is a large influence of the artificial viscosity, which has a damping effect on the wave. But clearly, this is not the

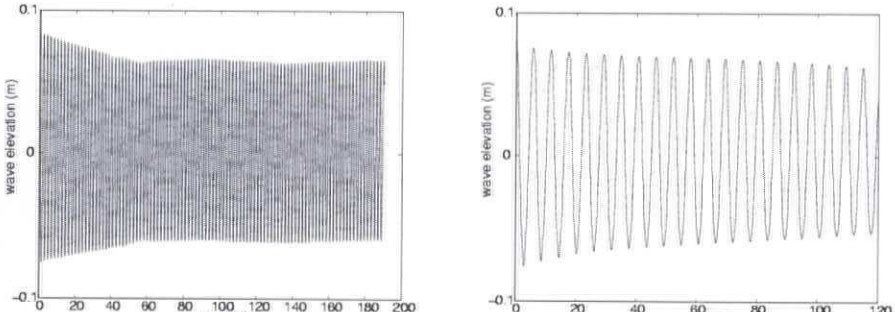


Figure 3.17: Wave elevation as function of time 14 wavelengths behind the inflow boundary (left) and as function of place after a simulation time of 100 periods (right)

which?

only reason for dissipation of energy in the wave simulations. Although in the central discretisation no artificial viscosity is present, the wave has been damped much more than it should be according to the physics. This is due to the boundary conditions and the displacement of the free surface that sometimes are chosen to be a bit dissipative to ? get a stable solution.

Summarising, for wave simulation in a very long domain, where many periods are simulated, a clear damping is visible. This dissipation of energy is for a great deal due to the artificial viscosity that is added when using an upwind discretisation. But another part of the energy dissipation is coming from the treatment of the boundaries and the free surface. Although an energy preserving discretisation is used (see [26]), some energy is lost in other parts of the algorithm. The loss of energy is only a few percent in a small domain when not that many periods are simulated. So for the applications of wave loading on ships, the influence will not be very significant.

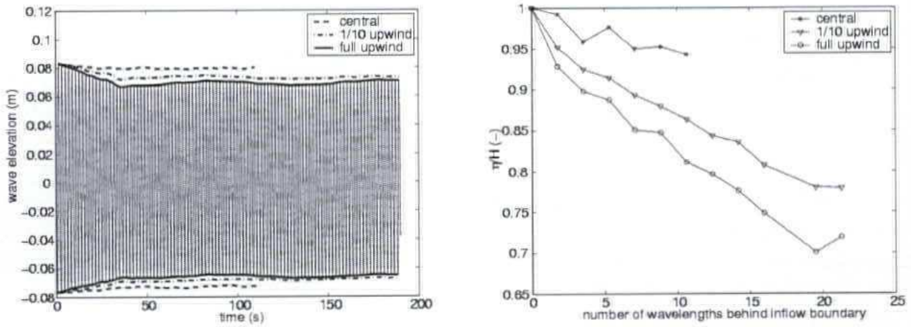


Figure 3.18: Left: wave elevation at 10 wavelengths behind the inflow boundary; right: decrease of wave height after a simulation time of 100 periods at different distances from the inflow boundary

3.6.2 Two-dimensional wave propagation: steep wave events

For the validation of irregular steep or high waves, waves generated on basis of Newwave theory have been used [90]. Such a wave event has been developed as a mean wave shape of a high wave in a given linear random sea-state. The wave is modelled by a superposition of linear waves, and thus is a linear wave. Of course high waves are not linear of form, and therefore, mostly some nonlinear corrections are being made to the waves. The waves are sometimes used as a design wave, because they have the nice feature that the position and time of the largest impact can be predicted beforehand.

At MARIN these design waves have been used in the experimental program of the SafeFLOW project to get a better understanding of wave slamming. Before the experiments with a vessel in the waves is performed, the waves have been calibrated in the basin without the presence of a vessel. The measurements of this undisturbed wave elevation have been used to compare our numerical method with. The characteristics of the wave events are described in Table 3.2 with H_{m_0} the significant wave height and T_0 the peak period.

Test no.	single wave event description
112003	1/14 steep wave in sea state steepness for 100 year return period
113001	mean highest wave for 100 year return period
114002	1/16 steep wave in sea state steepness for 100 year return period
119001	1/18 steep wave in sea state steepness for 100 year return period
120001	highest wave in design wave spectrum with $H_{m_0} = 12m,\ T_0 = 12s\ , \gamma = 2.5$
121001	steepest wave in design wave spectrum with $H_{m_0} = 12m,\ T_0 = 12s\ , \gamma = 2.5$

Table 3.2: Characteristics of the wave events, for which the simulations are compared with measurements

The waves have been generated at the inflow boundary of the numerical wave tank by using a Fourier transform of the time trace of a measured wave height. The calculated wave elevation in the domain is compared to the measurements at positions of 240, 480, 720, and 960 m behind the inflow boundary. Detailed measurements, using 19 wave probes, have been performed at the position where the wave is expected to be highest between 666 and 774 m behind the inflow boundary. In the left of Figure 3.19 the calculated wave height of test 112003 at 720 m behind the inflow boundary, where the wave is expected to be highest, is compared to theory and experiment.

The comparison with the experiment is very good. The amplitude of the simulation is a bit smaller than in the experiment, which has been observed before as a consequence of using Hirt-Nichols' method for the free surface displacement. Also the dissipation of energy due to the upwind discretisation could have an influence on the wave height. The simulation has been performed on two different grids, using 600 × 50 grid cells and 1200 × 100 grid cells. The grid refinement does not have a large influence, almost no difference can be observed between the two grids. In the left of Figure 3.19 also the wave elevation predicted from linear theory, extrapolated from the FFT of the measured wave elevation, is shown, which is very different from the calculations and measurement. This means that ComFLOW has correctly dealt with the nonlinearities that are present in reality. The simulations of the other events have all been performed with the fine grid of 1200x100 grid points. The results of tests 113001, 114002, and 119001, which are shown in Figures 3.20 to 3.22, are much the same as the results of the first test. From the left of Figure 3.20 it can be noticed that the wave does not completely reach the height of the experiment in the first peak. This is confirmed by the wave profile at 430 s, where the wave has collapsed a little. In Figure 3.23 and 3.24 the results of tests 120001 and 121001 have been shown. Again, the simulations do not reproduce the exact height of the first peak at about 395 seconds in both tests. Nevertheless, the other parts of the simulation are fairly close to the experiment.

The simulations in this section show that using FFTs to initiate the wave could well be used to reproduce a wave of the basin. To be sure that the wave is well reproduced in the simulation the wave must be compared with the undisturbed wave in the basin first. When it has been confirmed that the wave is the same as in the basin, it can be used for a simulation of wave impact. Further, the results show that ComFLOW reproduces nonlinearities from the experiment, but does not always reproduce the height of the wave exactly.

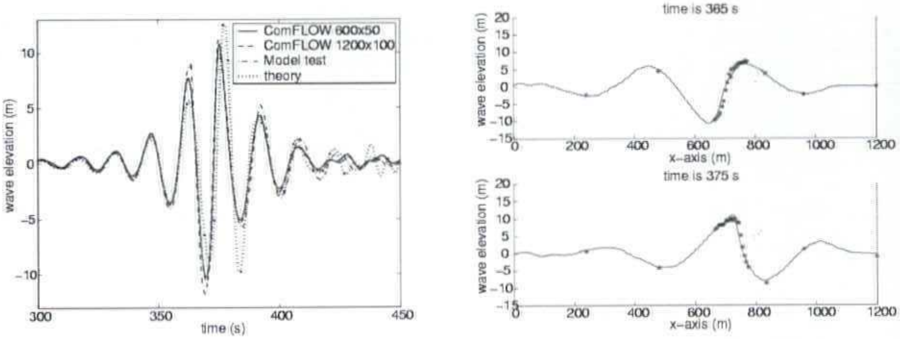


Figure 3.19: Test 112003: wave elevation at wave probe 4, 720 m behind the inflow boundary (left) and wave elevation at two different time points (right)

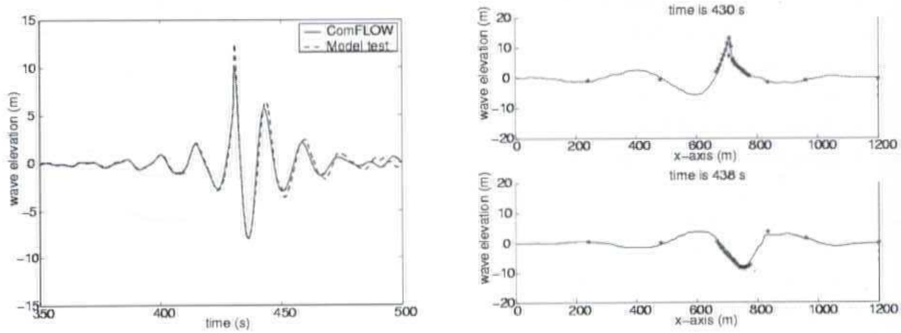


Figure 3.20: Test 113001: wave elevation at wave probe 4, 720 m behind the inflow boundary (left) and wave elevation at two different time points (right)

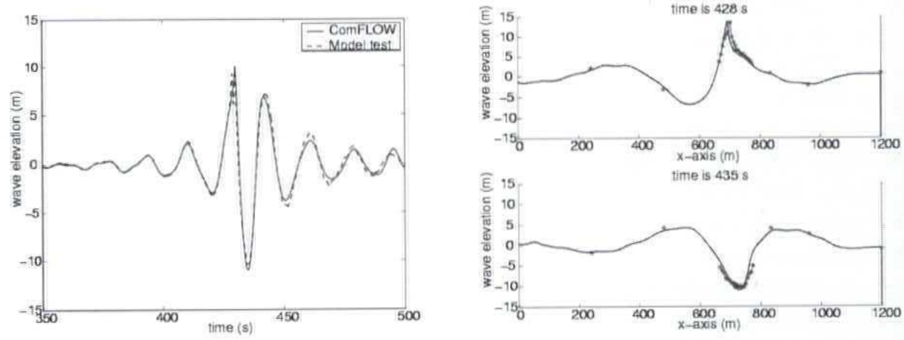


Figure 3.21: Test 114002: wave elevation at wave probe 4, 720 m behind the inflow boundary (left) and wave elevation at two different time points (right)

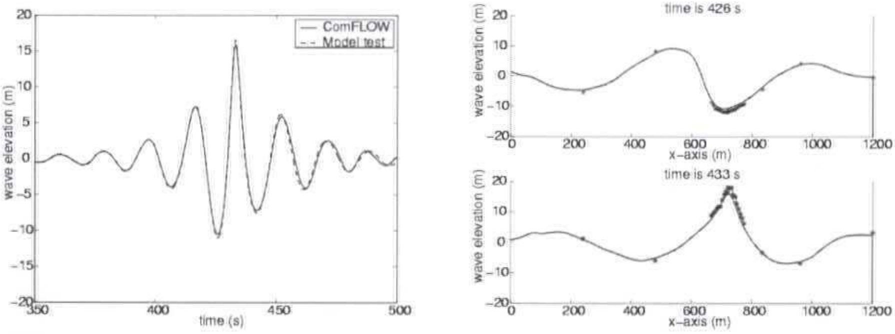


Figure 3.22: Test 119001: wave elevation at wave probe 4, 720 m behind the inflow boundary (left) and wave elevation at two different time points (right)

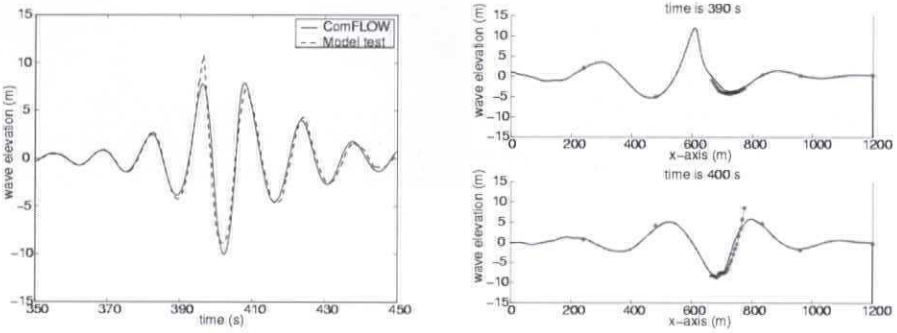


Figure 3.23: Test 120001: wave elevation at wave probe 4, 720 m behind the inflow boundary (left) and wave elevation at two different time points (right)

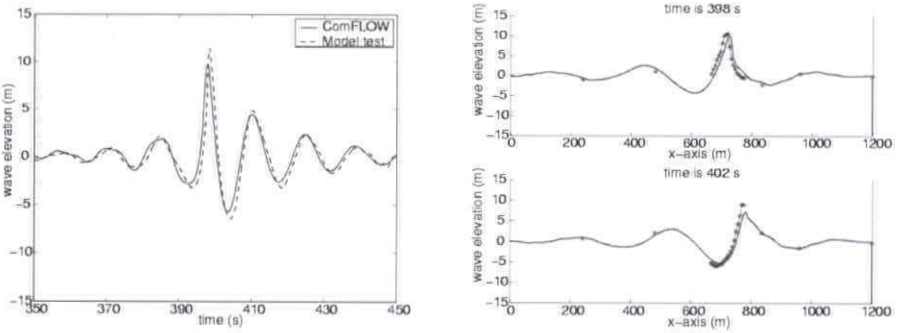


Figure 3.24: Test 121001: wave elevation at wave probe 4, 720 m behind the inflow boundary (left) and wave elevation at two different time points (right)

3.6.3 Regular wave loading on a spar platform

For the validation of wave loading, simulations have been performed of regular wave loading on a spar platform. A spar platform is a floating structure for drilling and the production of crude oil in the ocean. Typically, a spar is a long cylindrical steel structure with 30-50 meters in diameter and 200 meters in length. In Figure 3.25 a schematic picture and a photo of the Genesis spar platform are shown.

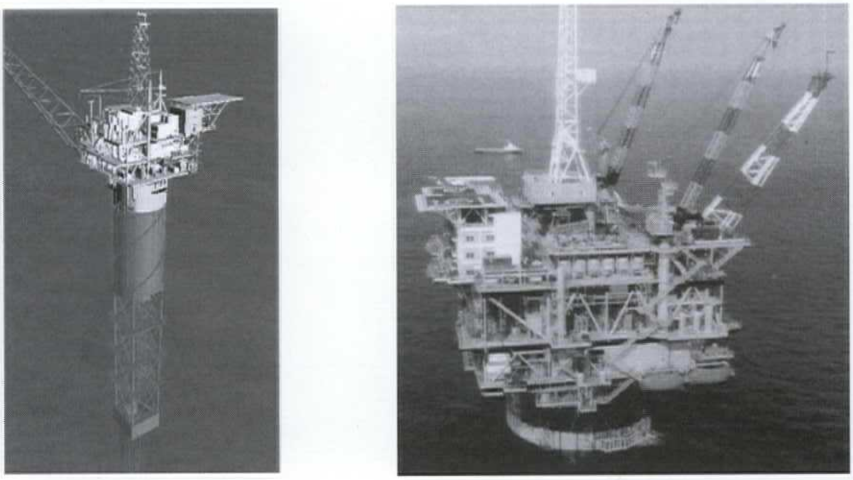


Figure 3.25: Left: schematic picture of a spar buoy, taken from http://www.ae.ic.ac.uk/; right: photo of the Genesis platform, taken from http://www.offshore-technology.com/

At MARIN experiments with a spar have been performed, where the spar was fixed while regular waves hit the structure. In full scale the spar has a total length of 220 m and a diameter of 35 m, the draft is 200 m in a water depth of 290.35 m. The spar has been divided into three horizontal segments, on which forces have been measured (see Figure 3.26).

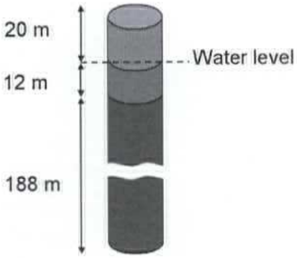


Figure 3.26: Configuration of the spar buoy, which is divided into 3 segments: the part above the calm water surface, the part 12 meter below the calm water surface and the remainder

The characteristics of the waves used in the simulations that have been performed are given in Table 3.3. The spar buoy has been placed at a distance of one wavelength behind the inflow boundary. About 100 m behind the cylinder, a dissipation zone of one wavelength has been added to prevent the wave from reflecting into the domain.

Test No.	frequency	period	wave height	wavelength
101	0.26 rad/s	24.17 s	10.77 m	866 m
201	0.26 rad/s	24.17 s	23.62 m	866 m
001	0.48 rad/s	13.09 s	10.91 m	274 m

Table 3.3: Wave characteristics of simulations of loading on a spar platform

The simulations of wave tests 101 and 102 have been performed on a grid with about 60 cells per wavelength, 15 cells in the transverse direction and 52 cells along the total height of the domain. The grid is stretched in the z-direction towards the calm water level. The wave in these tests is the same, except that the wave height in test 201 is about two times as large. The results of test 101 are shown in Figure 3.27 to 3.29.

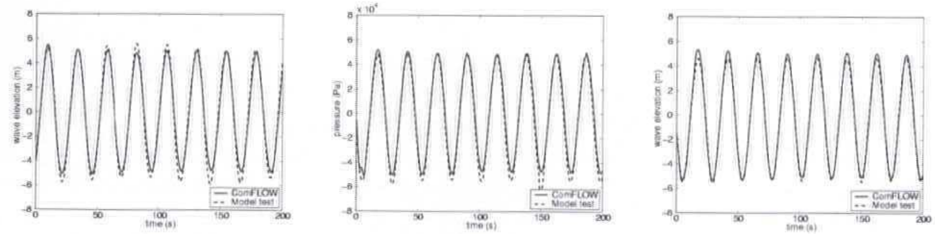


Figure 3.27: Test 101: Wave height 300 meter in front of the spar (left) and pressure (middle) and wave height (right) at the left of the spar

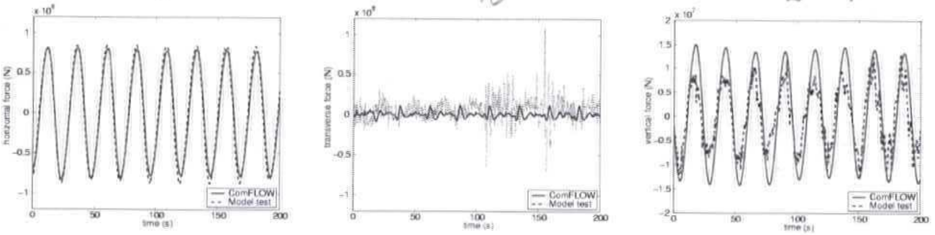


Figure 3.28: Test 101: Total force on the spar in the three coordinate directions: horizontal (left), transverse (middle) and vertical (right)

The global impression of the results is that there is a good agreement between the simulation and the model test. In the left of Figure 3.27 the wave height at about 300 meter in front of the spar has been shown. The comparison between the experiment and the simulation is satisfying. No phase difference is present and the amplitude of the wave is very much the same except for the trough of the wave, which is a bit flatter in the simulation. In the other two pictures of Figure 3.27 the wave height and pressure at the side of the spar is plotted. The results give the same impression as in the previous figure. The wave is not disturbed by the spar, which is a logical result when comparing the very large wavelength to the diameter of the spar.

$\lambda/d \approx$

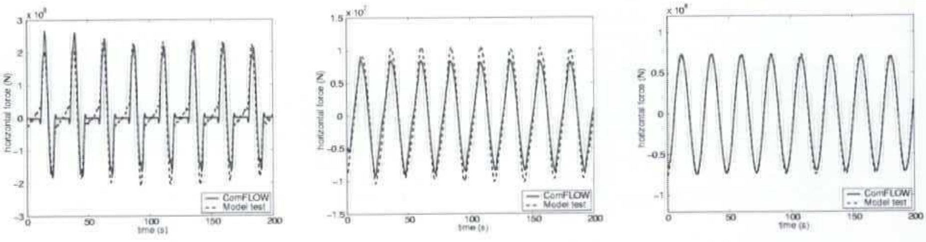


Figure 3.29: Test 101: Horizontal force on the three different segments of the spar: upper segment (left), mid segment (middle) and lower segment (right)

In Figure 3.28 the measured forces on the center of gravity of the spar platform have been compared to the calculated forces. The horizontal force, which is the force in the direction of the wave, has the largest amplitude and shows a very good agreement. The force in the transverse direction is and should be very small because of symmetry characteristics. In the vertical force a clear difference is observed between the measured and calculated amplitude, which cannot be explained at the moment.

The calculated forces on the different segments, which are shown in Figure 3.29, are in good agreement with the measurements. The force on the lower segment is almost equal to the total force on the spar. The upper segment lies above the calm water level, so this segment comes completely out of the water every period, where the force equals zero. The non sinusoidal form of the force on the mid segment is also due to the fact that the wetness of the segment is not constant throughout a period.

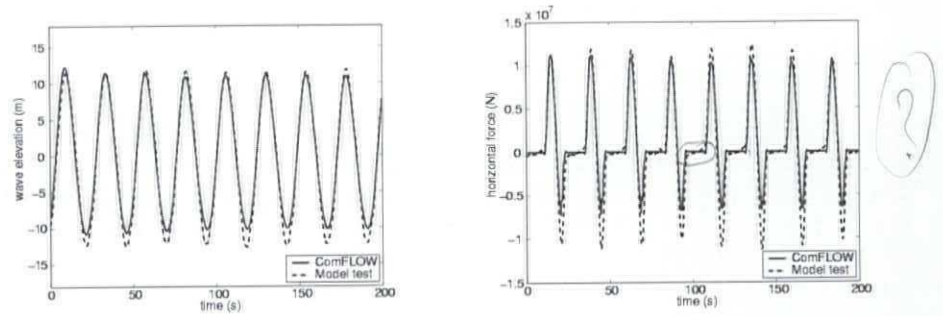


Figure 3.30: Test 201: Wave height 300 meter in front of the spar (left) and horizontal force on the upper segment (right)

The results of test 201 are much the same as the results presented for test 101. To get a good idea of the agreement between this model test and the simulation, the wave height 300 meter in front of the spar and the forces on the different segments are shown in Figures 3.30 and 3.31. The wave is really flatter in the troughs, which is also reflected in the horizontal force on the upper and mid segments where the simulation results show a smaller amplitude of the force. The horizontal force on the lower segment is in good agreement with the experimental data.

The third wave loading simulation that has been performed, deals with a much shorter

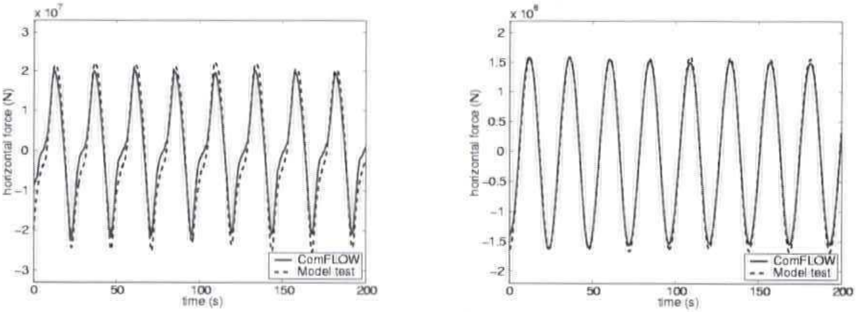


Figure 3.31: Test 201: Horizontal force on the mid segment (left) and on the lower segment (right)

(and more realistic) wave. In this case, the influence of the spar on the wave is not negligible as it was in the previous cases. This is also shown in the results of the simulation of this wave (with the same number of grid cells per wavelength), of which the wave height at the front of the spar and the total horizontal force on the spar are shown in Figure 3.32 (coarse grid). Many more cells are needed around the spar, therefore also stretching in x and y direction is applied in the mid and fine grid of Figure 3.32. The fine grid has about 3.5 times more grid cells in the neighbourhood of the spar than the coarse grid. From Figure 3.32 it can be concluded that the fine grid reproduces the model test very well, whereas the other grids are not accurate enough. In Figure 3.33 time traces of the horizontal forces on the upper and mid segment of the spar have been shown. Both graphs show a rather good agreement with the experiment.

3.6.4 Green water on the deck of a moving FPSO

A very demanding test case for the current simulation method is the calculation of loads due to green water on the deck of a moving FPSO (Floating Production Storage and Offloading vessel). For validation, an experiment performed at MARIN has been used. Measurements were done of the wave in front of the FPSO, relative wave heights in the neighbourhood of the FPSO, water heights and pressures at the deck of the FPSO and the pressure at some places at a deck structure. The FPSO has a total length of 260 meter and is 47 meter wide. The draft is 16.5 meter, the total height of the deck at the fore side of the FPSO is 25.6 meter. There is a bulwark extension of 1.4 meter. At the deck a box-like structure has been placed, at which forces and pressures have been measured. The bow has a fully elliptical shape without flare. The wave has a period of 12.9 seconds and the wave length is 260 meter, equal to the length of the FPSO. The wave amplitude is 6.76 meter. To be sure that the same wave has been used in the experiment and in the simulation, the wave measurement 230 meter in front of the bow of the vessel has been used to initiate the wave at the inflow boundary. The signal from the wave probe has been decomposed in linear components that are prescribed at the inflow boundary. The motion of the ship is prescribed using the measurements of the experiment. The simulation has been performed with only half of the FPSO in a relatively small domain around the bow of the vessel. A grid with 100 × 60 × 80 grid points has been used,

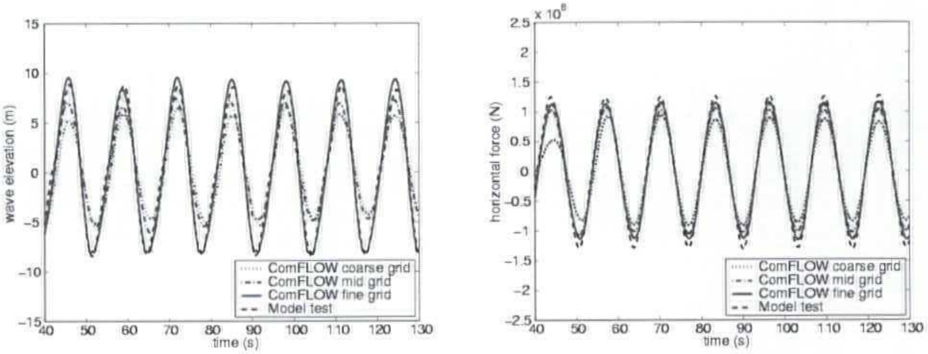


Figure 3.32: Test 001: Comparison of different grids: wave height at the front of the spar (left) and horizontal force on the total spar (right)

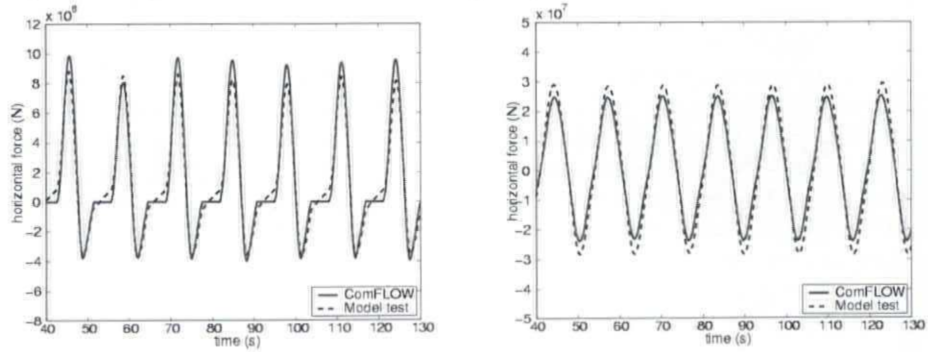


Figure 3.33: Test 001: Horizontal force on the upper segment (left) and on the mid segment (right) using the fine grid

resulting in a calculation time of about 4.5 hours. In Figure 3.34 some snapshots of the simulation are shown during the first period of the simulation. The large wave is building in front of the vessel, after which it starts to flow onto the deck. The water flows off the deck when the ship is straightening.

In Figure 3.35 the relative wave height in front of the vessel is shown. In both pictures there is a good agreement, such that it can be concluded that the motion of the vessel relative to the wave motion does not differ much in simulation and experiment. The water height on the deck of the vessel has been compared in Figure 3.36. When the water has just flowed onto the deck (left figure), the agreement between the experiment and the simulation is reasonable. The moment in time the wave probe gets wet is almost the same. But in the first periods, the water height is somewhat higher in the simulation, whereas the total time the water hits the wave probe is shorter. Closer to the deck structure, in the right of Figure 3.36, the total amount of water passing the wave probe is much smaller in the simulation. This same behaviour can be seen from the pressure on the deck and the deck structure. Whereas the pressure at the deck just behind the fore point of the FPSO agrees reasonably well, the pressure at the deck structure is much

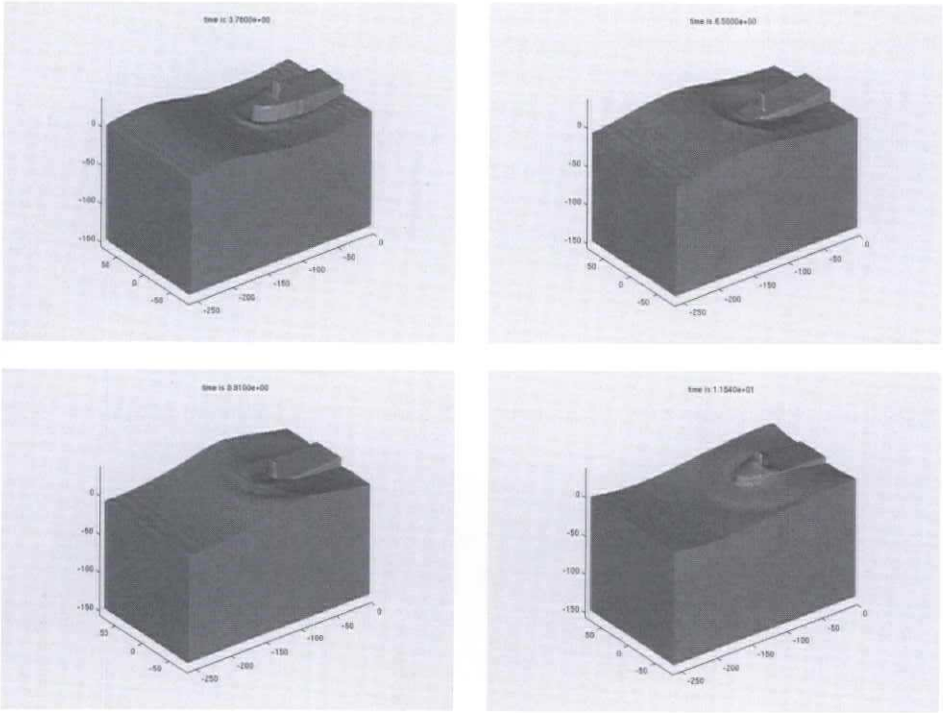


Figure 3.34: Snapshots of half of an FPSO shipping green water

lower, indicating that in the simulation only a small amount of fluid reaches the deck structure. The velocity of the water on the deck is quite well predicted by the simulation. The moment in time the water reaches the deck structure is almost exactly the same in experiment and simulation.

There can be several reasons for the differences between simulation and experiment. Firstly, the grid may not be fine enough to simulate the flow on the deck correctly. The vertical size of a cell at the deck is about 0.5 m, implying that there are at most 10 cells in the water height. Further, there can still be a problem with the phase between the wave and the motion of the vessel that are both prescribed, although this difference is seen when examining Figure 3.35. To check that, a fully interactive simulation would need to be performed, where the motion of the vessel is not prescribed, but calculated during the simulation. At the moment, this is not feasible, but in the future this option will be available. A first attempt in this direction is presented in [51] and [23], where a wedge and a box are falling into calm water. Another way of matching the vessel motion and the incoming wave is to calculate the motion and wave kinematics using an external code and prescribe the calculated wave and vessel motions in the ComFLOW simulation. This method has been used in Chapter 4. A last reason for the differences between the simulation and experiment could be the geometry of the ship, which is an approximation of the ship used in the experiment, based on the given characteristics.

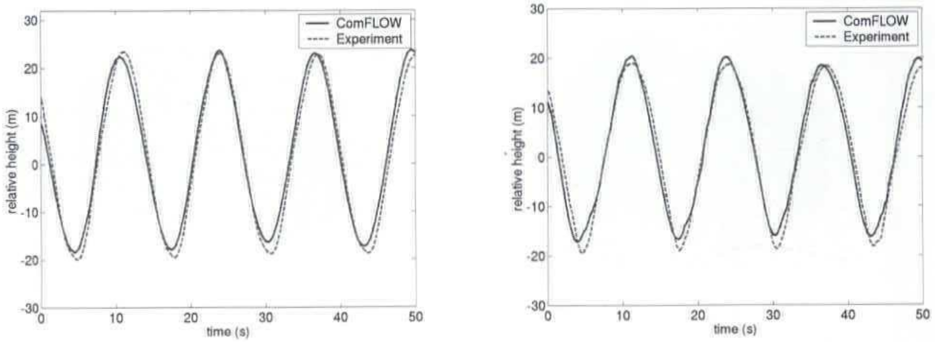


Figure 3.35: Relative wave height 30 meter (left) and 5 meter (right) in front of the FPSO

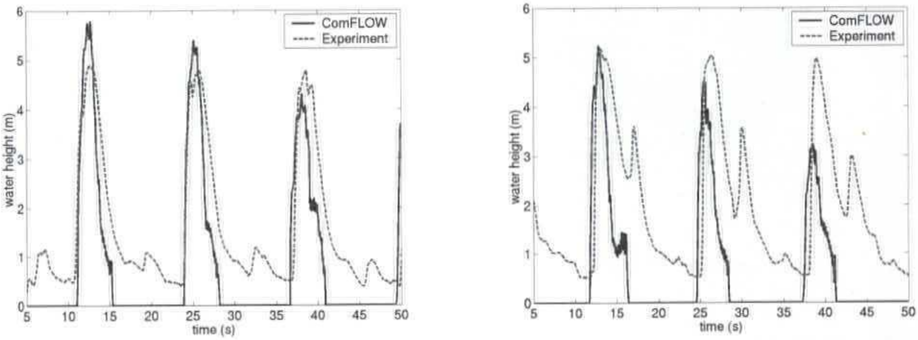


Figure 3.36: Water height on the deck of the FPSO: at the fore side of the bow (left) and near the deck structure (right)

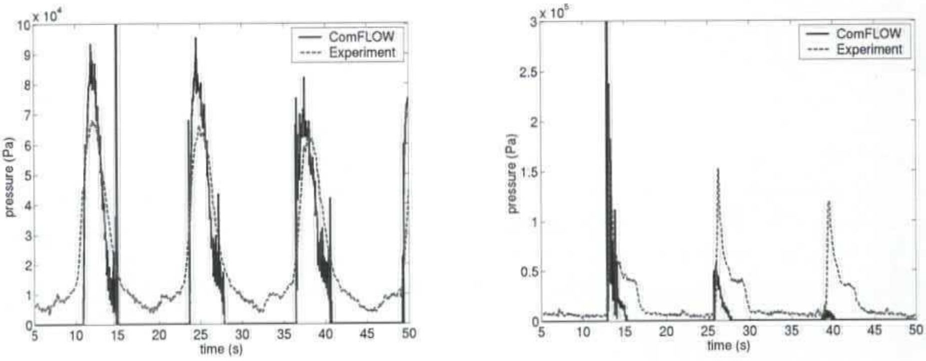


Figure 3.37: Left: pressure on the deck of the FPSO; right: pressure at the deck structure

Chapter 4

Generation of waves using zonal modelling

4.1 Introduction

In order to initialise waves in a numerical wave tank like ComFLOW, several methods can be used. First, the waves can be generated using a wave maker, which is modelled by a moving object. Second, the waves can be generated by prescribing velocities using known wave theories, for example Airy wave theory or 5th order Stokes theory. This method has been used in the previous chapter. The third method for the generation of waves uses zonal modelling, where the domain is split into a domain focused on the object in the flow and a far wave field domain. In the smaller domain ComFLOW is used for the local flow computation and the far wave field kinematics are calculated using an external code (see Figure 4.1 for a schematic picture). The physics covered by the external code can and will be more limited than in the local domain, where the full Navier-Stokes equations are solved. Thus, zonal modelling is used to increase the efficiency of the simulations. The water velocities calculated by the external code are prescribed on the open boundaries of the Navier-Stokes domain. Also the motion of the object is prescribed from the external code. This method has been adopted in the current chapter.

4.2 Description of the zonal modelling method

The external code for the calculation of the outer domain is a linear diffraction code based on potential theory. The diffraction code is able to calculate the wave kinematics and vessel motions, but can not calculate the local flow phenomena close to the vessel like wave impact and overturning waves. The local flow will be calculated by the Navier-Stokes solver, which is able to produce arbitrary interface configurations and can predict local wave impact. However, the use of the Navier-Stokes solver is limited to the close surroundings of the vessel because of the required computational effort. So, as sketched in Figure 4.2, first the diffraction code is used to calculate the far wave field and the vessel motions and then the Navier-Stokes equations are solved in the local domain to calculate the local wave dynamics. This is a one-way coupling, the Response Amplitude Operator (RAO's) calculated by the diffraction code in the frequency domain are used

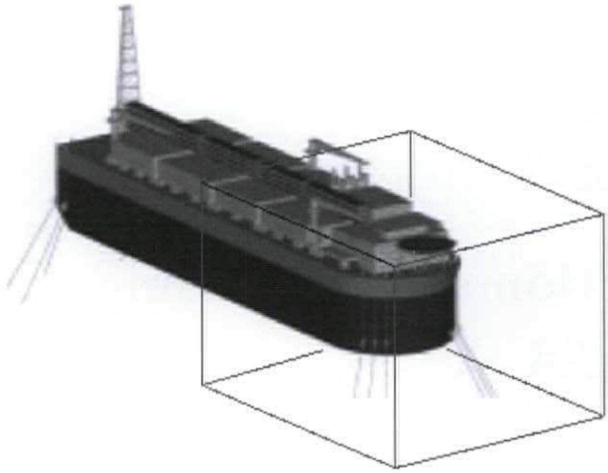


Figure 4.1: Zonal modelling: in small domain around the bow of the vessel local impact calculations will be performed by ComFLOW; the wave kinematics and vessel motion follow from an efficient external method

to prescribe the initial flow in the complete Navier-Stokes domain and to prescribe the velocities and the water height at the boundaries of the Navier-Stokes domain during the time domain simulation. The advantages of using zonal modelling for wave impact simulations are listed below.

- A diffraction code is very efficient and can easily calculate the fluid kinematics and response of the vessel in a three-hour storm. From these results a critical event can be chosen, which will be simulated with ComFLOW.

- The motion of the ship is calculated by the external code, and is used to prescribe the ship motion in the local domain calculation.

- The boundaries of the Navier-Stokes domain can be prescribed using the calculated wave kinematics by the diffraction code. The kinematics calculated by the diffraction code already consist of a part of the incoming wave and a part of the diffracted and radiated waves due to the moving object. This is a large advantage above prescribing a wave on the open boundaries using a wave theory, because in that case only the incoming wave components are prescribed.

To begin with, a linear code has been selected for the far wave field in order to gain experience on how far this method can be stretched as a first step towards developing an engineering tool. In this thesis only preliminary results are shown and problems that were met during the investigation are indicated. The zonal modelling method for wave generation will be investigated further in a future project.

An interface has been created, for coupling of the inner Navier-Stokes domain to the outer diffraction domain. Input for the interface are the wave kinematics calculated by

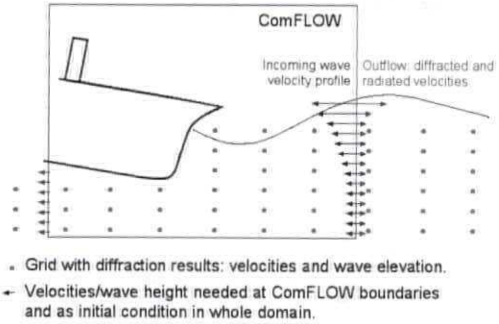

ComFLOW
Incoming wave velocity profile
Outflow: diffracted and radiated velocities
. Grid with diffraction results: velocities and wave elevation.
← Velocities/wave height needed at ComFLOW boundaries and as initial condition in whole domain.

Figure 4.2: A diffraction code calculates the vessel motions and wave kinematics at a grid of points; these are used to give the initial configuration in the smaller Navier-Stokes domain and the velocities at the open domain boundaries

the diffraction code. In the interface, time series are created for specified wave conditions, using the diffraction data. Based on the time series of wave height and vessel motions (which can be for example a 3-hour storm), an event of a few wave periods is selected to be calculated by ComFLOW. Output from the interface are the kinematics of the wave field and the ship motion. The interface consists of the following steps:

1. Read the output of the diffraction analysis program. The diffraction program has been run for a specified geometry at a range of wave frequencies ω_m and wave directions β_n and has produced RAO's (or transfer functions) for 6 degrees-of-freedom vessel motions, and transfer functions for pressure and velocities in a grid surrounding the vessel (the dots in Figure 4.2). The transfer functions are dependent on the wave frequency and wave direction, and are given by $U(\omega_m, \beta_n)$, $V(\omega_m, \beta_n)$, $W(\omega_m, \beta_n)$ and $P(\omega_m, \beta_n)$ for the velocity components and pressure, respectively.

2. Specify wave conditions, for which a simulation will be run. Several conditions can be chosen, for example conditions based on the JONSWAP spectrum (a spectrum based on a wave measurement program in the North Sea [22]), or a regular wave.

3. Generate time series belonging to the specified wave conditions using the transfer functions of the diffraction analysis. For example, the time series for the horizontal velocity in a point P is given by

$$u_P(t) = \sum_m \sum_n \hat{a}_{m,n} U_P(\omega_m, \beta_n) e^{i\omega_m t},$$

with $\hat{a}_{m,n}$ the complex amplitude, which is given by

$$\hat{a}_{m,n} = A(\cos \phi_{m,n} + i \sin \phi_{m,n}).$$

Here, $\phi_{m,n}$ is the phase angle belonging to wave component (m, n).

4. Calculate time series of the vessel motion. Select from the time series of wave elevation and vessel motions a wave event by specifying a start and end time for

the ComFLOW simulation. This event is chosen such that a critical situation occurs with green water risk or bow slamming due to a steep wave. For example, a high wave event can be chosen, where the deck level of the ship is expected to be exceeded by a high wave based on examining the relative wave motion.

5. Interpolate the transfer functions to the grid used in ComFLOW and prepare the data to be read by ComFLOW. The data needed are the initial velocity and pressure field and initial position of the vessel, and during the time domain simulation the boundary conditions at the open domain boundaries and the vessel motion.

4.3 Simulation of an irregular wave

The first test case that has been run with the zonal modelling method is the simulation of a two-dimensional linear irregular wave without an object in the flow. The wave has wave height $H_s = 0.455$ m, period $T_p = 12$ s and the water depth is 400 m. The JONSWAP wave spectrum has been used with 80 frequencies. The ComFLOW domain is 250 m long. The kinematics calculated by the diffraction code have been prescribed at the inflow and outflow boundary. The kinematics above the mean sea level are calculated using Wheeler stretching, because no theoretically correct linear wave kinematics exist there. The result of ComFLOW perfectly agrees with the predicted linear wave (see the left of Figure 4.3, where theory and simulation are on top of each other). This demonstrates correct implementation of the interface, the assumption that these waves behave linearly, and that ComFLOW is run with sufficient resolution and does not introduce unphysical nonlinearities for this wave train.

[illegible]

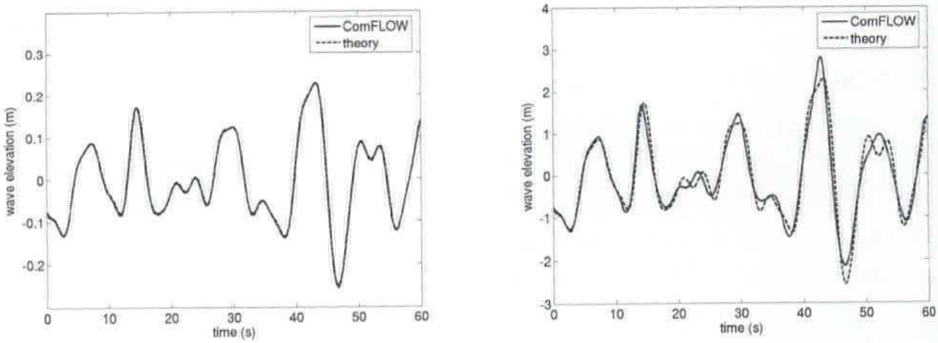


Figure 4.3: Time trace of an irregular linear wave (left) and nonlinear wave (right) computed with prescribed velocities from linear diffraction theory; the time trace is taken halfway the ComFLOW domain; note the difference in the vertical scale

When using the same wave, but ten times higher ($H_s = 4.55$ m), the wave has become nonlinear. There are differences now between ComFLOW and the predicted linear wave (see the right of Figure 4.3). These differences have two origins. Firstly, the wave has become nonlinear, so the predicted wave using linear theory is not correct any more. Secondly, reflections from the outflow boundary occur, because the velocities prescribed at the outflow boundary using the linear diffraction code are not correct any more and do not fit to the interior of ComFLOW. After some time, these reflections will also reach the inflow boundary, where the same problems will occur. To prevent this problem of reflections, another procedure than just prescribing velocities from the diffraction code should be used on the outflow boundary. For example, a damping zone could be used. But also more sophisticated methods can be implemented, because already a reasonable guess for the velocities is present from the diffraction results. For example, a damping zone that only damps the difference between the Navier-Stokes velocities and the diffraction velocities could be used. At the inflow boundary an active wave generating-absorbing boundary condition could be used, as introduced by e.g. [89]. In this model, an extra term is added above the incoming wave term that neutralises the reflected wave from an object for example. Options to minimise the differences between the prescribed boundaries and the interior of the domain will be investigated further in a future project.

4.4 Wave loading on a spar platform

The second simulation that has been performed to test the initialisation using a diffraction code consists of regular wave loading on a fixed spar. This case has already been validated in Section 3.6.3 using a 5th order Stokes wave on the inflow boundary and a damping zone at the outflow boundary. Now, the wave is initialised using results of a diffraction code, where the same simulation has been performed. So both components, inflow and diffraction, are included in the diffraction results. Two different simulations have been performed. First, all the walls of the Navier-Stokes domain have been prescribed by the diffraction results. Second, the inflow boundary has been prescribed by the diffraction results and at the outflow boundary a damping zone has been added. The side walls are closed walls. The nonlinear wave that has been used has period 24.2 seconds and wave height 10.8 meter. The water depth is 290 meter. The spar is modelled by a cylinder with a diameter of 35 m and a draft of 200 m.

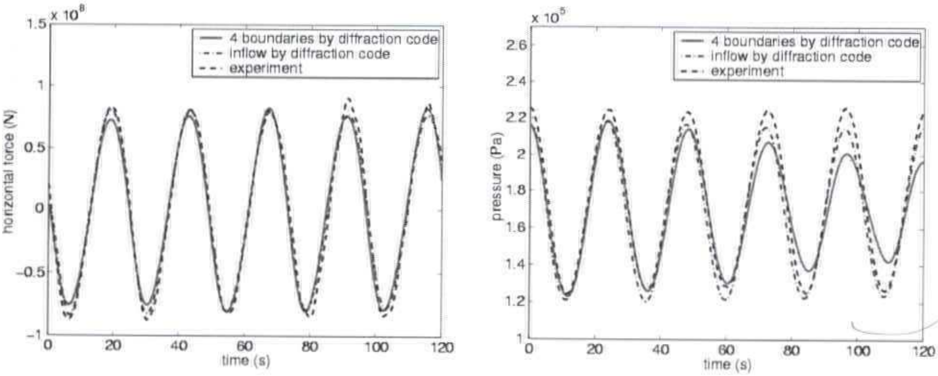


Figure 4.4: Simulation of wave loading on a spar platform; left: total horizontal force on the spar; right: pressure at the spar just below the water line

In Figure 4.4, the total horizontal force at the spar and the pressure at the spar 20 meter below the water line are shown for the two simulations and the experiment. In the total horizontal force there is not a very large difference between both simulations. However, when looking at the pressure just below the water line, a large difference is visible. The pressure in the simulation where four boundaries have been prescribed is damping in time, whereas the pressure in the case that only the inflow boundary is ? prescribed is a regular signal. The damping in the first simulation is caused by reflections from the side walls and outflow wall. The calculated velocity field in ComFLOW does not fit to the boundaries. This can be prevented by using a damping zone at the outflow boundary as can be seen in the second simulation.

linear

4.5 Simulation of green water loading

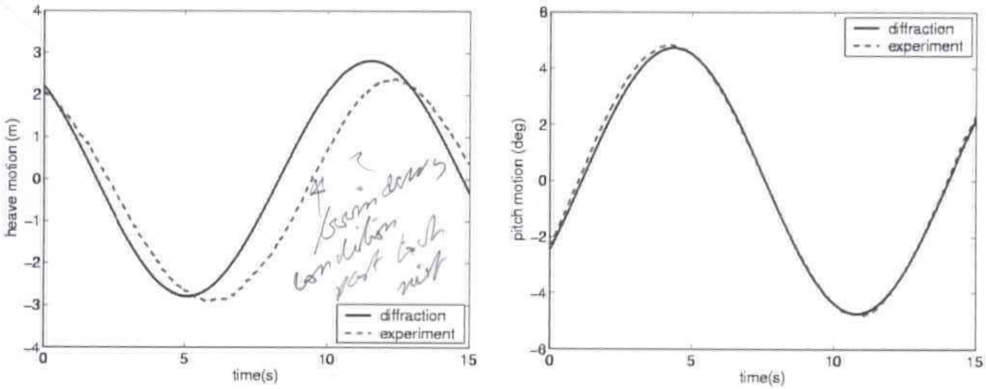
The goal of the coupling of ComFLOW to a diffraction code is to simulate realistic wave events in an efficient way and calculate the accompanying forces on offshore structures. To test the idea, a simulation of green water loading is performed using the same wave field characteristics as in an experiment performed by Buchner [7]. In the experiment a freely moving Floating Production Storage and Offloading vessel (FPSO) has been placed in regular waves with period 12.9 seconds, wave height 13.52 meter and wavelength 260 meter. The total water depth is 150 meter. Measurements were done of the wave in front of the FPSO, relative wave heights in the neighbourhood of the FPSO, water heights and pressures at the deck of the FPSO (see Figure 4.9) and the pressure at some places in a deck structure. The FPSO has a total length of 260 meter and is 47 meter wide. The draft is 16.5 meter, the total height of the deck at the fore side of the FPSO is 25.6 meter. There is a bulwark extension of 1.4 meter. At the deck, a box-like structure has been placed at which forces and pressures have been measured. The bow has a fully elliptical shape without flare.

Two simulations have been run. In the first simulation, the wave and vessel motions are determined by the measurements of the experiment. In Section 3.6.4 of the previous chapter (see also [53]) a study of the results of this simulation is given. The second simulation is initiated using a diffraction calculation for the far field kinematics and the vessel motions. The diffraction code that is used is a linear diffraction code, developed at MARIN [70]. Although the circumstances are very nonlinear, the linear approximation is thought to be a good start to investigate the possibilities of the method. The results of the zonal modelling method are compared with the experiments and also with the first simulation method.

4.5.1 Simulation using zonal modelling

First, the diffraction code has been run, from which the vessel motion RAO's and the kinematics at a grid of points around the vessel are written to a file. In Figure 4.5 the predicted heave and pitch motion of the vessel (heave motion is measured at the center of gravity) is compared with the measured motion. The heave and pitch motion are the most important motions for the prediction of green water on the deck, which occurs when the deck level is exceeded by the water level. As can be seen from the figure, the motions are well predicted by the linear diffraction theory. Only a slight shift in the predicted heave motion is observed and a small reduction of the amplitude. This could be due to the influence of the green water on the deck that is not taken into account in the diffraction calculations.

no infl? on pitch?



4? ? boundary condition part to the rest

Figure 4.5: Ship motion predicted by linear diffraction theory and measured during the experiment: heave (left) and pitch (right).

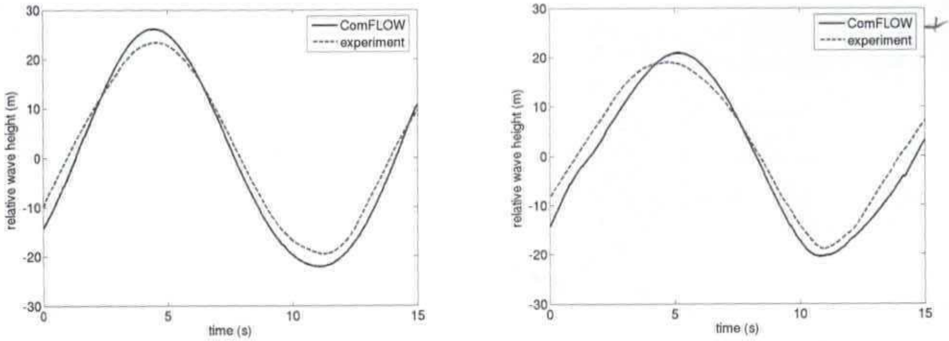
The vessel motions predicted by the linear diffraction code are used to prescribe the motion in the ComFLOW simulation. Furthermore, the velocity field and wave height are prescribed in the entire domain as initial condition and at the open boundaries during the simulation from the calculated kinematics using the diffraction code. The grid on which the kinematics are calculated in the diffraction code consists of 61 × 21 × 13 grid points with a distance between the grid points of 13, 13 and 12.5 m in x, y and z-direction, respectively. The ComFLOW domain is focused on the bow of the ship, and extends about half a wavelength up front and half the ship length aft of the bow. The y-coordinate has values between -100 and 100 m, whereas in z-direction the domain is cut off at -100 m. At the boundaries of the domain the three velocity components and the water height are prescribed using the results of the diffraction code. Selecting a grid for such computations is always a compromise between accuracy, computer memory and computing time. Here, a grid of 150 × 70 × 100 grid points is used in the ComFLOW simulation with stretching towards the bow of the ship. Using this grid, cells in the neighbourhood of the bow have sizes in x, y and z-direction of 1 m, 2 m and 1 m respectively. The simulation has been carried out for 15 seconds. The calculation time for this simulation is 10 hours and 23 minutes on a 2.8 GHz PC with 1Gb memory.

(16K)

1.05 10^6

To investigate the behaviour of the wave close to the bow, relative wave probes have been positioned at 30 and 5 meter in front of the bow. In Figure 4.6 the relative wave height calculated by ComFLOW is compared with the experiment. The agreement between measurement and calculation is good, indicating that the combined vessel motion and wave is well predicted.

Figure 4.7 shows contours describing the propagation of the water front on the deck of the FPSO. Buchner [7] describes the water flow on the deck in the following words: "First, the horizontal velocity of the water front on the deck is almost zero. Then, the water front starts to translate onto the deck with a similar velocity from all sides, perpendicular to



+ diff

Figure 4.6: Relative wave height 30 meter (left) and 5 meter (right) in front of the bow of the FPSO.

the local deck contour. Finally, the water contours from the front and sides meet at the centerline of the ship and result in a high velocity 'jet', which flows with a high velocity aft along the middle of the deck. "These stages can be recognised in the contour plots in Figure 4.7. In the experiment, the time interval between two contours is 0.31 s, and in the simulation the difference is 0.30 s. The agreement between the propagation of the water front in experiment and simulation is rather good. The water jet is formed a bit earlier in the simulation than in the experiment. In Figure 4.8 a snapshot of the simulation at time 7.5 s is shown, where the high velocity jet is very well visible.

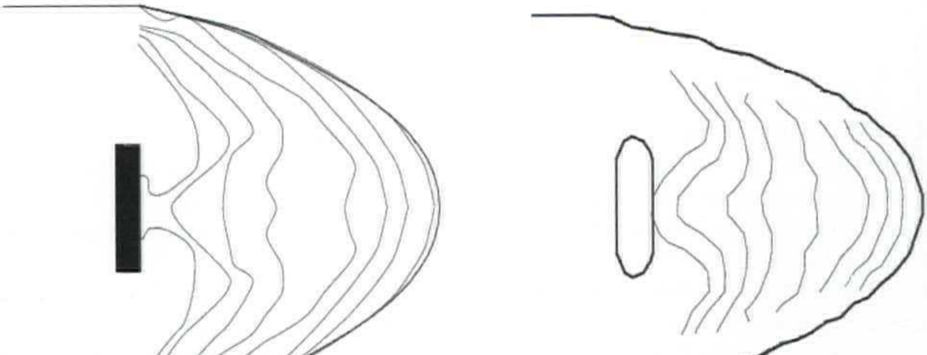


Figure 4.7: Contours of the water front propagating over the deck of the FPSO, model test every 0.31 s (left) and ComFLOW every 0.30 s (right).

To make a further comparison of the behaviour of the water on the deck with the experiment, the measurements of the water probes at the deck and the pressure panels at the deck have been used. Figure 4.9 shows the measurement positions of the water probes and pressure panels at the deck of the FPSO: at four positions the water height is measured, with distance between two of them of 10 m; the distance between the pressure

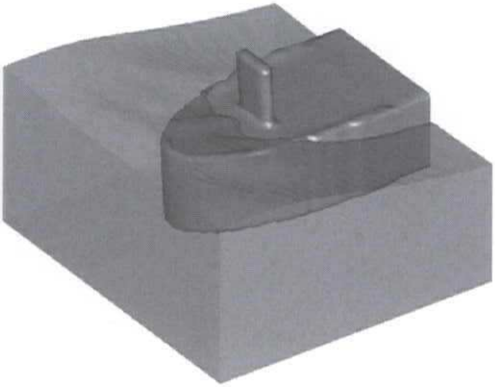


Figure 4.8: Snapshot of a simulation with green water on the bow of the FPSO.

panels that are positioned in between the water height probes is also 10 meter, with P1 positioned 7 m left of H1. Height probe H4 is positioned just in front of the deck structure.

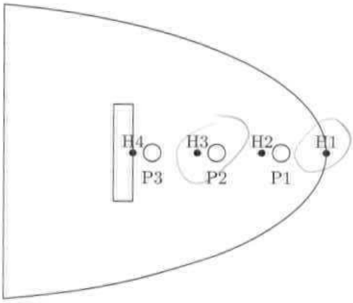


Figure 4.9: Positions of measured pressure and water height at the deck of the FPSO.

In Figure 4.10 the water height at the deck at positions H1 and H3 is shown. The moment the water reaches the water probes is well predicted by ComFLOW. In the first stage of the water flow on the deck, the amount of water is larger than in the experiment. Close to the deck edge (at H1) the water height is 2 meter higher than measured in the experiment; note that this difference is only two grid cells. Further aft on the deck at H3, the agreement of the water on the deck becomes better. The second hump in the right of Figure 4.10 is predicted by ComFLOW at the same moment as in the experiment. This hump is present due to the overturning water, after the run-up against the deck structure. In Figure 4.11 the pressure at the deck at positions P1 and P3 is shown. The same conclusion can be drawn from this plot as from the plotted water heights: the amount of water on the deck is too large, but further aft on the deck the agreement becomes better. In both pictures of Figure 4.11 an oscillating behaviour of the pressure can be observed. Every oscillation represents a switch of the monitor point fixed at the moving structure to another cell. The pressure value changes when such a switch of cell happens, because the pressure is positioned in the cell center (no interpolation is applied).

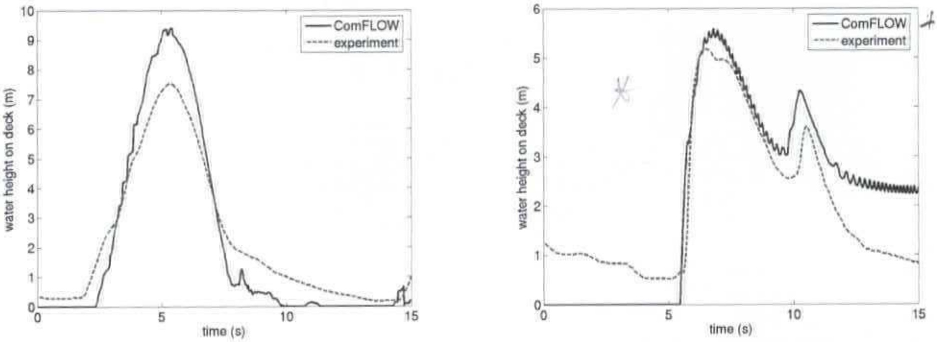


Figure 4.10: Water height at the deck close to the bow H1 (left) and closer to the deck structure H3 (right).

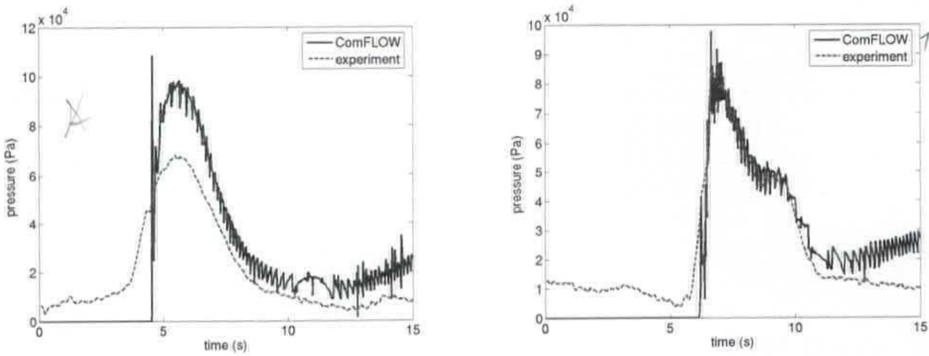


Figure 4.11: Pressure at the deck close to the bow P1 (left) and close to the deck structure P3 (right).

At the deck structure also pressure panels are positioned to measure the wave impact. The lowest panel is positioned 2.4 m above the deck level. The time trace of the load on that panel is shown in Figure 4.12, which shows a good agreement between simulation and experiment. There is a peak in the simulation during the first impact that is too high in the simulation, but it is not clear whether this peak is physical or whether it is originating from the numerical calculation.

4.5.2 Comparison with simulation initialised using experimental data

As described above, two simulation methods for green water loading are used. In the previous section zonal modelling is used, where the domain is split into a small domain

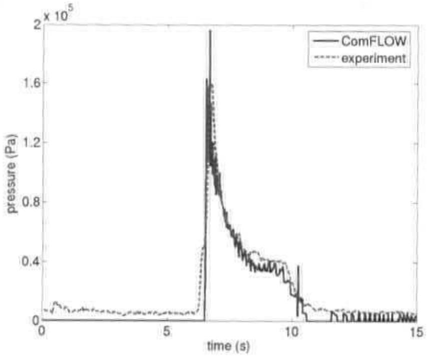


Figure 4.12: Pressure at the center line of the deck structure, 2.4 m above the deck.

around the bow and a far field. In these different domains ComFLOW and a linear diffraction code are used, respectively. In this section the results are compared with the results of ComFLOW initiated using the experiment. In this simulation the motion of the FPSO is prescribed from the measurements of the experiment. The wave is generated at the inflow boundary by a superposition of linear components. The components are deduced from the time signal of a wave probe at 230 m in front of the bow using Fourier analysis. The side walls are solid walls and at the outflow boundary a non-reflecting condition is used. Results of this simulation have already been presented in the previous chapter, Section 3.6.4. The upper part of Figure 4.13 shows the pressure and water height at the deck (positions P1 and H3, see Figure 4.9). The results of the two simulations have been compared with the measurements. At the forward part of the deck both simulation methods predict too much water as can be seen from the pressure at P1. Further aft on the deck the water height is better predicted by the simulation method using zonal modelling. The water height calculated using the measurements to initiate ComFLOW is too low. This results in smaller loads on the deck structure as can be seen from the bottom of Figure 4.13. There the pressure at the deck structure, 2.4 m above the deck level, is plotted. The picture is zoomed in to be able to compare the results of the two methods with the experiment. Concluding, using the zonal modelling method gives superior results above initiating ComFLOW with the experimental data. Two reasons are identified for this behaviour. First, the domain is smaller in the zonal-modelling simulation, resulting in a finer grid. The domain is 280 m long, 200 m wide and the water depth is cut off at 100 m, compared to 400 m length, 300 m width and 150 m depth in the other simulation. Also the number of grid points is higher in the zonal modelling simulation. The second reason can be found in the prescription of the boundary conditions at the open boundaries. Especially, reflections can occur at the side walls and the outflow wall, where closed walls and a Sommerfeld condition, respectively, are used in the simulation initialised using the experiment. This probably influences the results.

??

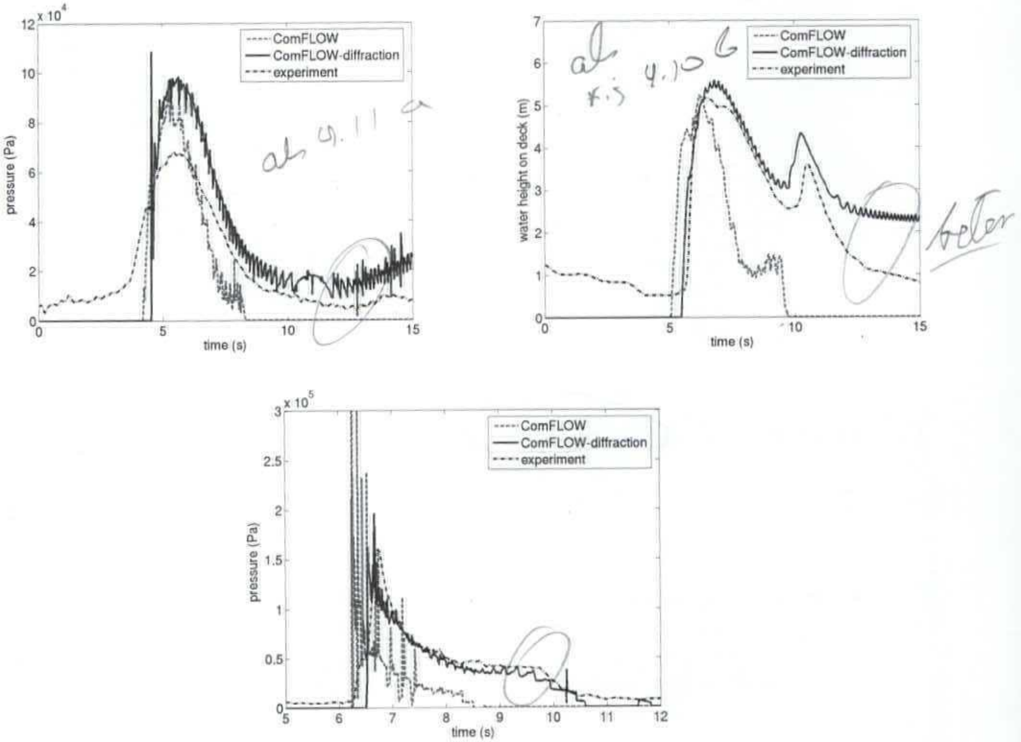


Figure 4.13: Pressure at position P1 (left), water height at position H3 (right) and pressure at deck structure (bottom): ComFLOW driven by experimental results and by the diffraction code compared.

4.5.3 Grid refinement

The simulation of green water on the deck has been run on three different grids to investigate the behaviour under grid refinement. The grid of points on which the diffraction results are calculated is not changed, only the ComFLOW grid is altered. The number of grid points used is 75 × 35 × 50, 113 × 53 × 75, and 150 × 70 × 100. All three grids are focused towards the bow of the FPSO.

Figure 4.14 shows the water height at position H3 at the deck of the ship (see Figure 4.9) and the pressure at the deck structure. The results on the finest grid are clearly better than the coarse grid results. At position H3, where the water height on the deck is measured and calculated, the difference in the first stage between the results on the three grids is not very large. Only the prediction of the second hump, where the water has overturned after run-up against the deck structure, gets clearly better on the finest grid. The impact of the water on the deck structure is not predicted accurately on the coarsest grid, but the two finer grids perform equally well.

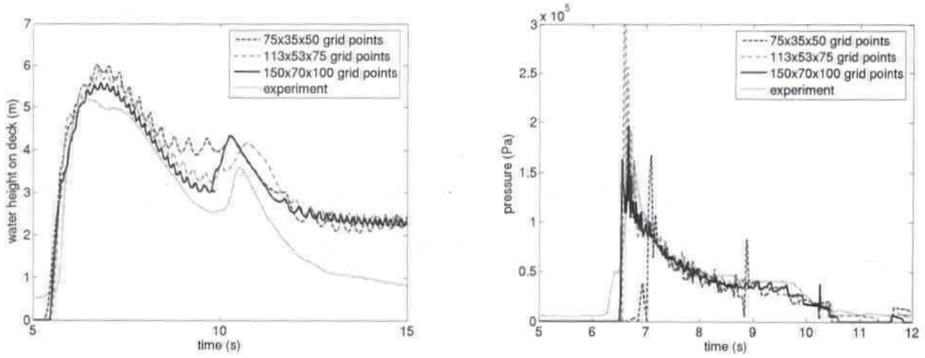


Figure 4.14: Water height H3 at the deck of the FPSO and the pressure at the deck structure calculated on three different grids.

4.6 Discussion of the results

In this chapter the first results of a zonal modelling method for the generation of waves have been presented. In this method waves are generated on the boundaries of the Navier-Stokes domain using calculated far field kinematics by a linear diffraction code. The aim of this investigation is to determine possibilities and limitations of this method, which are described in this section.

First, calculations have been performed of a two-dimensional irregular wave without an object in the flow, resulting in zero contributions from diffracted and radiated waves. At the open boundaries diffraction results are prescribed and inside the small Navier-Stokes domain the wave profile is calculated. In the case of a linear wave, the resulting wave elevation calculated by ComFLOW compares perfectly with the wave elevation calculated by the diffraction code. But when the wave is higher and becomes nonlinear the results do not agree very well. Two reasons have been identified: the linear diffraction code does not give correct answers for a nonlinear wave and the boundaries of the Navier-Stokes domain do not fit any more with the interior of the domain. So, to improve the results a nonlinear external code has to be used. [illegible] problem

Second, wave loading on a fixed spar platform has been computed, where the waves are regular and nonlinear. A few periods have been calculated to investigate the ability of the coupled method in a longer time simulation. First, all open boundaries of the Navier-Stokes domain are prescribed from the diffraction code results. In that case, the wave is damping during the periods because of the reflections from the walls of the domain that do not fit very nicely to the interior anymore. When prescribing only the inflow boundary using linear diffraction results and using a damping zone at the opposite wall the results improve, because the reflections are diminished. So, a proper way of fitting the interior of the Navier-Stokes domain to the boundaries where diffraction results are used is needed. For example, a sophisticated damping procedure could be used, where the wave is not damped completely, but only the components that do not fit to the diffraction results are damped.

Finally, a moving FPSO in high waves is simulated resulting in green water on the bow of the vessel. Only one period is simulated to prevent problems with reflections from the walls. The calculated results agree rather well with measurements that are available. The zonal modelling method is better able to predict the green water loading than a wave prescribed using Fourier analysis of the measured wave elevation. The results of this simulation show the great potential of the zonal method indicating that it is worthwhile to elaborate on this further.

Chapter 5

Conclusions and recommendations

In this thesis a numerical method has been presented for the simulation of free-surface flow. The focus of this work has been on the simulation of wave impact loading on off-shore structures. Especially, the event of high waves resulting in green water on the deck of an FPSO has been an important application. The method has been incorporated in a simulation program called ComFLOW. At the start of this project, the method was able to simulate fluid flow in complex geometries with the presence of a free liquid surface and moving bodies. At the domain boundaries no-slip or free-slip conditions were imposed and simple inflow and outflow conditions were implemented. To be able to perform wave impact simulations as aimed for in this project, wave generation options have been implemented. Also an investigation has been performed of outflow boundary conditions that are needed to prevent waves from reflecting at the domain walls. It has been investigated whether the waves are damped due to numerical choices, like the upwind discretisation of the convective terms in the Navier-Stokes equations. Since the impact is measured by pressure time traces, attention has been paid to the smoothness of the pressure signal. Further, attention has been paid to the robustness of the method, especially in handling the free surface that can be deformed severely in these kind of applications. Thereto, and to get as accurate results as possible, the displacement algorithm and the boundary conditions at the free surface have been reconsidered. The method has been validated extensively; results of the most demanding test case of impact due to green water on a moving vessel have been presented.

Discretisation method

For the simulation of fluid flow, the Navier-Stokes equations have been discretised on a fixed Cartesian grid using staggered variables. The moving object cuts through the fixed grid, resulting in cut cells, of which the open part can be arbitrary small. A finite volume method has been used for the discretisation of the equations with starting point the conservative formulation of the Navier-Stokes equations.

The discretisation has been chosen such that the properties of the underlying analytic operators are preserved in the discrete operators. This leads to a discretisation scheme where the kinetic energy is only dissipated due to the diffusive term, and further energy change is fully controlled by the motion of a moving object. So the discretisation does not lead to uncontrolled energy increase. For the time discretisation the forward Euler method has been used. This method leads for uncut cells to the CFL-criterion $|u|\delta t/h < 1$, with h the size of the uncut cell. For cut cells the criterion is not sharpened, even for the smallest cells. When moving objects are present, this is not true any more. To prevent the cut-cell size occurring in the stability criterion, a formulation based on a weighted average of the fluid velocity and the object velocity has been applied in cut cells. Using this formulation, fulfilling the CFL-criterion is again sufficient for stability.

Using the cut cell method for moving objects results in the occurrence of numerical spikes in the pressure time series. This is largely due to the sudden change of a label, when a cell changes from boundary cell to fluid cell and vice versa. To prevent these spikes, the pressure has been decoupled in a physical pressure and a potential that contains the spikes. This physical pressure part is calculated by putting back the moving object to its former position, such that no change in labels occurs. When the geometry apertures are calculated exactly, as can be done in two dimensions, the resulting pressure signal is much smoother. However, in three dimensions, where geometry apertures are calculated approximately, this decoupling procedure does not have much effect, since the transition from boundary cell to fluid cell is much smoother due to the approximate geometry apertures. Still many spikes are present in the pressure and force time traces, which cannot be prevented in a straightforward manner. In the post-processing standard filtering techniques can be used to get smooth signals.

Free surface displacement

For the displacement of the free surface a Volume-of-Fluid method has been adopted. In the applications of the current thesis, where the free surface is deformed severely, the VOF method is extremely suitable, since it can handle highly distorted interfaces, such as overturning waves, merging and splitting. In literature VOF methods are often classified by the method for the reconstruction of the interface. In this thesis the method of Hirt and Nichols, with a constant reconstruction of the interface, and Youngs' method, with a piecewise linear interface reconstruction, have been described. The methods have been combined with a local height function to establish perfect mass conservation and to prevent flotsam and jetsam that appear in Hirt-Nichols' original method. From standard kinematic tests, it can be concluded that Youngs' method gives the best results. In a real case simulation of a breaking dam Hirt-Nichols' and Youngs' methods both combined with a local height function give similar satisfactory results, whereas both methods without local height function do not conserve mass. For Youngs' method, this lack of mass conservation is due to the choices of free surface boundary conditions that have nonzero divergence of the velocity in a surface cell. In a simulation of a steep wave event, the conclusion about the performance of the methods is different: Youngs' method without local height function performs best, whereas the combination with local height function gives worst results.

At the moment no clear conclusion can be drawn about which method should be used. The classification of methods in accuracy is not unambiguous for the different kinds of simulations. Therefore, this should be investigated further. Especially, the bad result of Youngs with local height function in the simulation of waves should be understood. This investigation should be performed in combination with an evaluation of the method for boundary conditions at the free surface. When a method for the free surface velocities is used with zero divergence in surface cells (method 1 in Section 2.3.10), Youngs' method can be used without local height function, since mass conservation is then achieved. In the present situation it is recommended to use Hirt-Nichols' method in combination with a local height function. Since Youngs' method sometimes gives more accurate results, simulations can also be performed with Youngs' method, keeping in mind the problems that could occur.

Boundary conditions at the free surface

It has been shown that the choice of the velocity boundary conditions at the free surface has a large effect on the accuracy and robustness of the method. Using mass conservation in surface cells results in a smooth pressure signal, but it is not robust and not accurate in wave simulations. Using extrapolation of velocities gives accurate wave simulation results, especially when using linear extrapolation, but results in numerical spikes in the pressure. A combination of methods is used: when the velocity field is smooth, the velocities are determined using linear extrapolation, whereas constant extrapolation is used when the velocity field is not smooth. To prevent the occurrence of numerical spikes, velocities at the free surface are determined using mass conservation when a cell is changing from surface or empty cell to fluid cell. This robust and accurate method is based on engineering decisions. A mathematically more elegant method is to perform calculations in the whole domain, such that two-phase flow is simulated. Then the velocity field is smooth in the whole domain, mass conservation automatically holds in every cell (also the surface cells, so no problems with pressure spikes), and no velocity boundary conditions at the free surface are needed. Currently, a two-phase method is implemented in the simulation program [94].

To show the performance of the simulation method without inflow and outflow boundaries, results of a breaking dam and water entry tests have been presented. The dambreak simulation has been performed with a box in the flow, on which pressures and forces have been calculated. The comparison of the present method with experimental results is very good. The method is well able to predict the impact loads resulting from a highly complex flow. To validate the impact of a moving object, water entry tests have been performed for wedges with different dead-rise angles, a cone and a circular cylinder. The visual agreement between snapshots of the simulations and photographs of experiments by Greenhow and Lin [37] is very good. On a fine grid the thin jets at the sides of the entering object are resolved fairly well. For the cone and the circular cylinder slamming coefficients have been calculated and when compared with available experiments and theoretical estimates, a satisfactory agreement was obtained.

Wave simulations: open boundaries

For the simulation of wave impact on offshore platforms, the simulation program has been extended with wave generation options. Two methods are available: first, the waves are generated at the inflow boundary using a wave description; second, an external program is used for the calculation of the wave kinematics that are prescribed at the open boundaries. In the first option, the waves can be initiated using linear wave theory or 5th order Stokes theory. Velocities and wave height are calculated using the wave description and are prescribed at the inflow boundary. When using linear theory, the kinematics above the calm water level has been calculated using the Wheeler stretching method. An irregular wave can be created by prescribing a superposition of linear components.

At the boundary opposite of the inflow boundary, conditions should be imposed that prevent the waves from reflecting into the domain. In case of regular waves that are not much deformed, a Sommerfeld boundary condition has been used. When the waves are not regular or are distorted, a dissipation zone has been added at the end of the domain. To damp the waves in the dissipation zone, a pressure term has been added at the free surface, that counteracts the wave motion.

When damping the wave completely, the water level rises during the periods. This is caused by the Stokes drift of the wave: the net mass flux at the inflow boundary is positive. To overcome this problem, the solid wall in front of which the wave is damped has been changed to an outflow boundary, where a hydrostatic pressure is prescribed assuming the water height to be the initial calm water level. This way, the water is forced to flow out when the water level rises.

Validation of wave propagation

Wave propagation through the calculational domain has been validated by two-dimensional simulations of regular waves and irregular wave events without object in the flow. The results of the regular wave simulations have been compared with wave theories. The number of time steps in a period and the number of grid cells per wave length and in the wave height have been varied. Both have shown to be of large influence. When using enough grid cells and small enough time steps, the error in the simulations is less than 5% after a simulation time of four wave periods.

Steep and high wave events have been simulated using measurements from the basin. Linear components are prescribed at the inflow boundary calculated using Fourier analysis of a measured time trace of the wave elevation. Several meters further in the flow, the simulation has been compared to the experimental results. It shows a rather good agreement. The high peaks are not always reproduced exactly. Using Youngs' method for the displacement of the free surface improves the results.

Attention has been paid to the dissipation of energy during the wave simulation. Due to the upwind discretisation of the convective terms, artificial viscosity has been added, which is much larger than the physical viscosity of water. The influence of the artificial viscosity has been measured by performing wave simulations with an upwind discretisation and a central discretisation (that needs very small grid sizes and time step). The influence is clearly visible after a simulation of a regular wave during many periods in a domain containing a large number of wave lengths. The central discretisation has less damping than the upwind discretisation, but also the central discretisation simulation suffers from energy loss. This energy dissipation is probably due to the boundary conditions at the free surface and the displacement of the free surface. In a small domain with only a few periods of simulation the dissipation is so small, that the influence will not be severe in the simulations of local wave impact on offshore platforms. To overcome the problem of damping the waves due to the artificial viscosity, the amount of artificial viscosity could be lessened by using a higher order upwind method. This can be investigated further in the future.

To validate the method for wave loading, a fixed spar in a regular wave field is simulated and the calculated forces on the spar have been compared to experimental results. It has been concluded that the method is able to reproduce results from those model tests with a satisfying accuracy. When a long, low wave is hitting the spar, the wave is almost not influenced by the spar. In that case a rather coarse grid can be used, to reproduce the wave correctly. But when a shorter wave is considered, a much finer grid is necessary to reproduce the details of the flow around the spar correctly. The grid choice is found to be essential in these kind of simulations.

The concept is also validated for the impact of green water on a moving FPSO, as the ultimate goal of this project. The regular waves are high, such that the water is flowing over the deck. The motion of the FPSO and the wave at the inflow boundary are prescribed from model tests. The calculated height of the water on the deck and pressures at the deck and deck structure have been compared with measurements. A reasonable agreement has been found at the start of the simulation. But when the water is flowing onto the deck, the agreement becomes less. An explanation for the differences could be the grid size, which probably should be smaller. Further improvement of the results could be established by doing a fully coupled simulation of ship motion and fluid flow. A first start of this coupling has already been shown in [51]. This can be elaborated further in a future project.

Zonal modelling for wave generation ?

Another way of generating waves is to use zonal modelling. In a relatively small domain the Navier-Stokes equations are solved, and thus the local impact on a structure is calculated. At the open boundaries of the Navier-Stokes domain kinematics is prescribed using an external program, which is a linear diffraction code. Also the motion of the structure is prescribed by the diffraction results.

First, calculations have been performed of a two-dimensional irregular wave without an object in the flow. In the case of a linear wave, the resulting wave elevation in the Navier-Stokes domain perfectly agrees with the wave elevation calculated by the diffraction results. But for a nonlinear wave the results do not agree very well. Two reasons are, that the linear diffraction code does not give correct answers for a nonlinear wave, and the boundaries of the Navier-Stokes domain do not fit any more with the interior of the domain. So, a nonlinear external code should be used to improve the results.

Further, regular wave loading on a fixed spar platform has been simulated using this method. The results of the simulation with all open boundaries prescribed from diffraction results show problems with reflections on the boundaries. When only the inflow boundary is prescribed from diffraction results and a damping zone is used for the opposite boundary, the calculations agree much better with the experiment. In future research attention has to be paid to the reflections at the boundaries, since the interior results do not agree with the boundaries where diffraction results are prescribed.

Finally, a moving FPSO in high waves is simulated resulting in green water on the bow of the vessel. Only one period is simulated to prevent problems with reflections from the walls. The calculated results agree rather well with measurements that are available. The zonal modelling method is better able to predict the green water loading than the

?

method where the wave is prescribed from wave theory. The results of this simulation show the great potential of this method indicating that it is worthwhile to elaborate on this further.

Epilogue

The method described in this thesis has shown to be able to simulate complex flows as wave loading and green water impact to a rather accurate degree. Therewith, the objective as formulated in the introduction has been achieved. The method has been shown very robust and able to predict local wave impact up to a rather satisfactory level. Still much effort must be put in the improvement and further development of the method. In a follow-up of the SafeFLOW project (the ComFLOW-2 Joint Industry Project) the simulation method will be extended with new developments. First, the possibility of calculating two-phase flow, where the air is compressible, a prerequisite in case of modelling air-entrapment, will be established. Second, the zonal modelling method for the generation of waves will be extended and improved. Besides these large-scale functional developments, also continuous attention will be paid to the improvement of the numerical aspects of the existing method.

Bibliography

[1] E. Aulisa, S. Manservisi, R. Scardovelli, and S. Zaleski, A geometrical area-preserving volume-of-fluid advection method, *J. Comput. Phys.*, **192**, pp. 355–364, 2003.

[2] M.R. Barkhudarov, Lagrangian VOF advection method for FLOW3D, Tech. Rep. FSI-03-TN63-R, Flow Science, Inc., July 2004.

[3] D. Battistin and A. Iafrati, Hydrodynamic loads during water entry of two-dimensional and axisymmetric bodies, *J. Fluids Struc.*, **17**, pp. 643–664, 2003.

[4] B. Biausser, S. Guignard, R. Marcer, and P. Fraunié, 3D two phase flows numerical simulations by SL-VOF method, *Int. J. Numer. Methods Fluids*, **45**, pp. 581–604, 2004.

[5] E.F.F. Botta and M.H.M. Ellenbroek, A modified SOR method for the Poisson equation in unsteady free-surface flow calculations, *J. Comput. Phys.*, **60**, pp. 119–134, 1985.

[6] E.H. van Brummelen and B. Koren, A pressure-invariant conservative Godunov-type method for barotropic two-fluid flows, *J. Comput. Phys.*, **185**, pp. 289–308, 2003.

[7] B. Buchner, *Green Water on Ship-Type Offshore Structures.* PhD thesis, Delft University of Technology, The Netherlands, 2002.

[8] B. Buchner, G. Fekken, T.H.J. Bunnik, and A.E.P. Veldman, A numerical study on wave run up on an FPSO bow, in *Proc. 20th Int. Conf. on Offshore Mechanics and Arctic Engineering, Rio de Janeiro, Brazil*, 2001.

[9] T. Bunnik and B. Buchner, Numerical prediction of wave loads on subsea structures in the splash zone, in *Proc. of 14th ISOPE Conf., Toulon, France*, 2004.

[10] I.M.C. Campbell and P.A. Weynberg, Measurement of parameters affecting slamming, Tech. Rep. Report No 440, Wolfson Unit of Marine Technology, Southampton, 1980.

[11] C. Cerjan, D. Kosloff, R. Kosloff, and M. Reshef, A nonreflecting boundary condition for discrete acoustic and elastic wave equations, *Geophysics*, **50**, pp. 705–708, 1985.

[12] G. Cerne, S. Petelin, and I. Tiselj, Numerical errors of the volume-of-fluid interface tracking algorithm, *Int. J. Numer. Methods Fluids*, **38**, pp. 329–350, 2002.

[13] R.K.-C. Chan and R.L. Street, A computer study of finite-amplitude water waves, *J. Comput. Phys.*, **6**, pp. 68–94, 1970.

[14] S. Chen, D.B. Johnson, and P.E. Raad, Velocity boundary conditions for the simulation of free surface fluid flow, *J. Comput. Phys.*, **116**, pp. 262–276, 1995.

[15] A. Clément, Coupling of two absorbing boundary conditions for 2d time-domain simulations of free surface gravity waves, *J. Comput. Phys.*, **126**, pp. 139–151, 1996.

[16] R. Courant, K.O. Friedrichs, and H. Lewy, Uber die partiellen differenzengleichungen der mathematisches physik, *Math. Ann.*, **100**, pp. 32–74, 1928.

[17] E.F.G. van Daalen, J. Gerrits, G.E. Loots, and A.E.P. Veldman, Free surface anti-roll tank simulations with a volume of fluid based Navier-Stokes solver, in *Proc. 15th Intern. Workshop on Water Waves and Floating Bodies, Ceasarea, Israel*, 2000.

[18] E.F.G. van Daalen, K.M.T. Kleefsman, J. Gerrits, H.R. Luth, and A.E.P. Veldman, Anti-roll tank simulation with a volume of fluid (vof) based Navier-Stokes solver, in *Proc. 23rd Symp. on Naval Hydrodynamics, Val de Rueil, France*, 2001. pp. 457–473.

[19] M.T. Dröge, *Cartesian-Grid Methods for Turbulent Flow Simulation in Complex Geometries.* PhD thesis, University of Groningen, The Netherlands, 2005. To be published.

[20] M. Dröge and R. Verstappen, A new symmetry-preserving Cartesian-grid method for computing flow past arbitrarily shaped objects, *Int. J. Numer. Meth. Fluids*, **47**, pp. 979–985, 2005.

[21] G. Ersdal and A. Kvitrud, Green water on Norwegian production ships, in *Proc. of 10th ISOPE Conf., Seattle*, 2000.

[22] O.M. Faltinsen, *Sea Loads on Ships and Offshore Structures*, Cambridge University Press, Cambridge, 1990.

[23] G. Fekken, *Numerical Simulation of Free-Surface Flow With Moving Rigid Bodies.* PhD thesis, University of Groningen, The Netherlands, 2004. URL: http://www.ub.rug.nl/eldoc/dis/science/g.fekken.

[24] G. Fekken, A.E.P. Veldman, and B. Buchner, Simulation of green-water loading using the Navier-Stokes equations, in *Proc. 7th Int. Conf. on Numerical Ship Hydrodynamics, Nantes, France*, 1999. pp. 6.3-1-6.3-12.

[25] J.D. Fenton, A fifth-order Stokes theory for steady waves, *J. Waterway, Port, Coastal and Ocean Engineering*, **111**, pp. 216–234, 1985.

[26] J. Gerrits, *Dynamics of Liquid-Filled Spacecraft.* PhD thesis, University of Groningen, The Netherlands, 2001. URL: http://www.ub.rug.nl/eldoc/dis/science/j.gerrits.

[27] J. Gerrits and A.E.P. Veldman, Numerical simulation of coupled liquid-solid dynamics, in *Proc. ECCOMAS 2000, Barcelona, Spain*, 2000.

[28] J. Gerrits and A.E.P. Veldman, Transient dynamics of containers partially filled with liquid, *Moving Boundaries VI, B.Sarler, C.A. Brebbia (eds.)*, pp. 63–72, 2001.

[29] J. Gerrits and A.E.P. Veldman, Dynamics of liquid-filled spacecraft, *J. Engrg. Math.*, **45**, pp. 21–38, 2003.

[30] D. Givoli, Non-reflecting boundary conditions, *J. Comput. Phys.*, **94**, pp. 1–29, 1990.

[31] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, and J. Périaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, *J. Comput. Phys.*, **169**, pp. 363–426, 2001.

[32] R. Glowinsky, T.W. Pan, and J. Périaux, A Lagrange multiplier / fictitious domain method for the numerical simulation of incompressible viscous flow around moving rigid bodies: (i) case where the rigid body motions are known a priori, *C.R. Academic Science Paris t. 324, Série 1, Mathematical Problems in Mechanics*, 1997.

[33] M. Gòmez-Gesteira, D. Cerqueiro, C. Crespo, and R.A. Dalrymple, Green water overtopping analyzed with a SPH model, *Ocean Engineering*, **32**, pp. 223–238, 2005.

[34] P. Gorf, N. Barltrop, B. Okan, T. Hodson, and R. Rainey, FPSO bow damage in steep waves, in *Proc. Rogue Waves, Brest, France*, 2000.

[35] M. Greco, O.M. Faltinsen, and M. Landrini, Shipping of water on a two-dimensional structure, *J. Fluid Mech.*, **525**, pp. 309–332, 2005.

[36] M. Greenhow, Wedge entry into initially calm water, *Appl. Ocean Research*, **9**, pp. 214–223, 1987.

[37] M. Greenhow and W.-M. Lin, Non linear free surface effects: experiments and theory, Tech. Rep. 83-19, Department of Ocean Engineering, MIT, Cambridge, Mass., 1983.

[38] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski, Volume-of-Fluid interface tracking with smoothed surface stress methods for three-dimensional flows, *J. Comput. Phys.*, **152**, pp. 423–456, 1999.

[39] F.H. Harlow and J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids*, **8**, pp. 2182–2189, 1965.

[40] D.J.E. Harvie and D.F. Fletcher, A new volume of fluid advection algorithm: the stream scheme, *J. Comput. Phys.*, **162**, pp. 1–32, 2000.

[41] D.J.E. Harvie and D.F. Fletcher, A new volume of fluid advection algorithm: the defined donating region scheme, *Int. J. Numer. Meth. Fluids*, **35**, pp. 151–172, 2001.

[42] B. Higdon, Absorbing boundary conditions for difference approximations to the multi-dimensional wave equation, *Math. Comp.*, **47**, pp. 437–459, 1986.

[43] C. Hirsch, *Numerical Computation of Internal and External Flow*, John Wiley & Sons, Volume 2, 1990.

[44] C.R. Hirt and B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.*, **39**, pp. 201–225, 1981.

[45] A. Iafrati, A. Di Mascio, and E.F. Campana, A level set technique applied to unsteady free surface flows, *Int. J. Numer. Meth. Fluids*, **35**, pp. 281–297, 2001.

[46] M. Israeli and S.A. Orszag, Approximation of radiation boundary conditions, *J. Comput. Phys.*, **41**, pp. 115–135, 1981.

[47] B. Iwanowski, H. Grigorian, and I. Scherf, Subsidence of the Ekofisk platforms: wave in deck impact study. Various wave models and computational methods, in *Proc. OMAE 2002, Oslo*, ASME Conferences, 2002. Paper Number OMAE2002-28063.

[48] M.S. Kim and W.I. Lee, A new VOF-based numerical scheme for the simulation of fluid flow with free surface. Part I: new free surface-tracking algorithm and its verification, *Int. J. Numer. Methods Fluids*, **42**, pp. 765–790, 2003.

[49] M.S. Kim, J.S. Park, and W.I. Lee, A new VOF-based numerical scheme for the simulation of fluid flow with free surface. Part II: application to the cavity filling and sloshing problems, *Int. J. Numer. Methods Fluids*, **42**, pp. 791–812, 2003.

[50] S.-O. Kim, Y. Sim, and E.-K. Kim, A first-order volume of fluid convection model in three-dimensional space, *Int. J. Numer. Methods Fluids*, **36**, pp. 185–204, 2001.

[51] K.M.T. Kleefsman, G. Fekken, A.E.P Veldman, B. Iwanowski, and B. Buchner, A Volume-of-Fluid based simulation method for wave impact problems, *J. Comput. Phys.*, **206**, pp. 363–393, 2005.

[52] K.M.T. Kleefsman, G.E. Loots, A.E.P. Veldman, B. Buchner, T. Bunnik, and E. Falkenberg, The numerical simulation of green water loading including vessel motions and the incoming wave field, in *Proc. 24th Int. Conf. on Offshore Mechanics and Arctic Engineering, Halkidiki, Greece*, 2005.

[53] K.M.T. Kleefsman and A.E.P. Veldman, An improved volume-of-fluid method for wave impact, in *Proc. ECCOMAS, Jyväskylä*, 2004.

[54] B. Koren, Lewis M.R., E.H. van Brummelen, and B. van Leer, Riemann-problem and level-set approaches for homentropic two-fuid flow computations, *J. Comput. Phys.*, **181**, pp. 654–674, 2002.

[55] D.B. Kothe and R.C. Mjolsness, RIPPLE: a new model for incompressible flows with free surfaces, *AIAA J.*, **30**, pp. 2692–2700, 1992.

[56] D.B. Kothe, W.J. Rider, S.J. Mosso, and J.S. Brock, Volume tracking of interfaces having surface tension in two and three dimensions, in *AIAA Paper 96-0859, 34th Aerospace Sciences Meeting and Exhibit, Reno*, 1996.

[57] G.E. Loots, *Fluid-Structure Interaction in Hemodynamics.* PhD thesis, University of Groningen, The Netherlands, 2003. URL: http://www.ub.rug.nl/eldoc/dis/science/g.e.loots.

[58] G.E. Loots, B. Hillen, H.W. Hoogstraten, and A.E.P. Veldman, Fluid-structure interaction in the basilar artery, *Finite Volumes for Complex Applications III, R. Herbin, D. Kroener (eds.)*, pp. 607–614, 2002.

[59] G.E. Loots, B. Hillen, and A.E.P. Veldman, The role of hemodynamics in the development of the outflow tract of the heart, *J. Engrg. Math.*, **45**, pp. 91–104, 2003.

[60] J. López, J. Hernández, P. Gómez, and F. Faura, A volume of fluid method based on multidimensional advection and spline interface reconstruction, *J. Comput. Phys.*, **195**, pp. 718–742, 2004.

[61] B. Le Méhauté, *An Introduction to Hydrodynamics and Water Waves*, Springer-Verlag, New York, 1976.

[62] G. Meskers, Realistic inflow conditions for numerical simulation of green water loading, Master's thesis, Delft University of Technology, The Netherlands, 2002.

[63] S. Mizoguchi, Analysis of shipping water with the experiments and the numerical calculations, *JSNA*, **163**, 1988.

[64] S. Mizoguchi, Design of freeboard height with the numerical simulation on the shipping water, in *PRADS'89*, 1989.

[65] J.J. Monaghan, Simulating free surface flows with SPH, *J. Comput. Phys.*, **110**, pp. 399–406, 1994.

[66] W.D.M. Morris, J. Millar, and B. Buchner, Green water susceptibility of North Sea FPSO/FSUs, in *Proc. 15th Conf. on Floating Production Systems (FPS), London*, 2000.

[67] S. Muzaferija, M. Peric, P. Sames, and T. Schellin, A two-fluid Navier-Stokes solver to simulate water entry, in *22nd Symp. on Naval Hydrodynamics*, 2000.

[68] K.B. Nielsen and S. Mayer, Numerical prediction of green water incidents, *Ocean Engineering*, **31**, pp. 363–399, 2004.

[69] W.F. Noh and P. Woodward, SLIC (simple line interface calculation), in *Lecture Notes in Physics*, A.I. van de Vooren and P.J. Zandbergen, eds., vol. 59, pp. 330–340, Springer, New York, 1976.

[70] G. van Oortmerssen, *The Motions of a Moored Ship in waves.* PhD thesis, Delft University of Technology, The Netherlands, MARIN publication No. 510, 1973.

[71] I. Orlanski, A simple boundary condition for unbounded hyperbolic flows, *J. Comput. Phys.*, **21**, pp. 251–269, 1976.

[72] S. Osher and J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.*, **79**, pp. 12–49, 1988.

[73] C.S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.*, **25**, pp. 220–252, 1977.

[74] X.P. Pham and K.S. Varyani, Evaluation of green water loads on high-speed containership using CFD, *Ocean Engineering*, **32**, pp. 571–585, 2005.

[75] J.E. Pilliod and E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, *J. Comput. Phys.*, **199**, pp. 465–502, 2004.

[76] E.G. Puckett, A volume of fluid interface tracking algorithm with applications to computing shock wave rarefraction, in *Proceedings of the 4th International Symposium on Computational Fluid Dynamics*, pp. 933–938, 1991.

[77] E.G. Puckett and J.S. Saltzman, A 3d adaptive mesh refinement algorithm for multimaterial gas dynamics, *Physica D*, **60**, pp. 84–93, 1992.

[78] W.J. Rider and D.B. Kothe, Reconstructing volume tracking, *J. Comput. Phys.*, **141**, pp. 112–152, 1998.

[79] M. Rudman, Volume-tracking methods for interfacial flow calculations, *Int. J. Numer. Methods Fluids*, **24**, pp. 671–691, 1997.

[80] R. Scardovelli and S Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Ann. Rev. Fluid Mech.*, **31**, pp. 567–603, 1999.

[81] R. Scardovelli and S. Zaleski, Interface reconstruction with least-square fit and split Eulerian-Lagrangian advection, *Int. J. Numer. Meth. Fluids*, **41**, pp. 251–274, 2003.

[82] M. Schiffman and D.C. Spencer, The force of impact on a cone striking a water surface (vertical entry), *Comm. Pure Appl. Math.*, **4**, pp. 379–417, 1951.

[83] L. Skjelbreia and J.A. Hendrickson, Fifth order gravity wave theory, in *Proc. 7th Coastal Engineering Conf., The Hague*, 1960.

[84] A. Sommerfeld, *Partial Differential Equations in Physics*, Academic Press, New York, USA, 1949.

[85] C.T. Stansberg, Ø. Hellan, J.R. Hoff, and V. Moe, Green sea and water impact: numerical predictions validated against model tests, in *Proceedings of OMAE 2002, Oslo*, ASME Conferences, 2002. Paper Number OMAE2002-28562.

[86] M. Sussman and E. Fatemi, An efficient, interface preserving level set re-distancing algorithm and its application to interfacial incompressible fluid flow, *SIAM, J. Sci. Comput.*, **20**, pp. 1165–1191, 1999.

[87] M. Sussman and E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, *J. Comput. Phys.*, **162**, pp. 301–337, 2000.

[88] H. Tank, L.C. Wrobel, and Z. Fan, Tracking of immiscible interfaces in multiple-material mixing processes, *Comp. Materials Science*, **29**, pp. 103–118, 2004.

[89] P. Troch and J. De Rouck, An active wave generating-absorbing boundary condition for VOF type numerical model, *Coastal Engineering*, **38**, pp. 223–247, 1999.

[90] P.S. Tromans, A.R. Anaturk, and P. Hagemeijer, A new model for the kinematics of large ocean waves, in *Proc. 1st Offshore and Polar Engineering Conf., Edinburgh*, 1991.

[91] H.S. Udaykumar, R. Mittal, P. Rampunggoon, and A. Khanna, A sharp interface Cartesian grid method for simulation flows with complex moving boundaries, *J. Comput. Phys.*, **174**, pp. 345–380, 2001.

[92] R.W.C.P. Verstappen and A.E.P. Veldman, Symmetry-preserving discretization of turbulent flow, *J. Comput. Phys.*, **187**, pp. 343–368, 2003.

[93] J. Wackers and B. Koren, A surface capturing method for the efficient computation of steady waves, in *Proc. 3rd Intern. Conf. on Advanced Computational Methods in Engineering, to be published*, 2005.

[94] R. Wemmenhove, E. Loots, R. Luppes, and A.E.P. Veldman, Modeling two-phase flow with offshore applications, in *Proc. 24th Int. Conf. on Offshore Mechanics and Arctic Engineering, Halkidiki, Greece*, 2005.

[95] J. Westhuis, *The Numerical Simulation of Nonlinear Waves in a Hydrodynamic Model Test Basin.* PhD thesis, University of Twente, The Netherlands, 2001.

[96] F. Xiao and A. Ikebata, An efficient method for capturing free boundaries in multi-fluid simulations, *Int. J. Numer. Methods Fluids*, **42**, pp. 187–210, 2003.

[97] O. Yilmaz, A. Incecik, and J.C. Han, Simulation of green water flow on deck using non-linear dam breaking theory, *Ocean Engineering*, **30**, pp. 601–610, 2003.

[98] D.L. Youngs, An interface tracking method for a 3D Eulerian hydrodynamics code, Tech. Rep. AWRE/44/92/35, Atomic Weapons Research Establishment, 1987.

[99] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys.*, **31**, pp. 335–362, 1979.

[100] S. Zhang, D. Liut, K. Weems, and W.-M. Lin, A 3-D finite volume method for green water calculations, in *Proc. 24th Int. Conf. on Offshore Mechanics and Arctic Engineering, Halkidiki, Greece*, 2005.

[101] R. Zhao, O. Faltinsen, and J. Aarsnes, Water entry of arbitrary two-dimensional sections with and without flow separation, in *21st Symp. on Naval Hydrodynamics*, 1997.

[102] Z.Q. Zhou, J.O. de Kat, and B. Buchner, A non-linear 3-D approach to simulate green water on deck, in *Proceedings 7th Intern. Conf. on Numerical Ship Hydrodynamics, Nantes, France*, 1999.

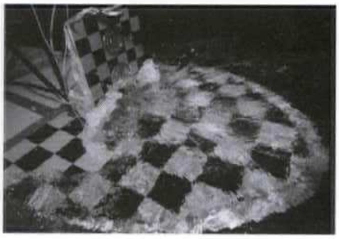
Samenvatting

Watergeweld bij offshore constructies

Een offshore constructie voor de productie of het opslaan van olie of gas ligt vaak op een vaste plaats voor een aantal jaren. Zulke constructies staan vast op de bodem van de zee, of het zijn drijvende constructies die bevestigd zijn aan de zeebodem met lijnen. Omdat deze constructies niet eenvoudig verplaatst kunnen worden bij naderend slecht weer, moeten ze ontworpen worden om alle soorten weertypen te doorstaan. Speciaal bij scheepsvormige constructies, zoals de Floating Production, Storage and Offloading constructies (FPSO's, zie de foto in Figuur 5.1) kan water op het dek grote problemen veroorzaken. Die ontstaan doordat golven zo hoog zijn dat ze over de boeg van het schip heenslaan, zodat een grote massa water over het dek stroomt. Deze massa water wordt groen water genoemd, naar de kleur van het zeewater dat groen is door fytoplankton. De snelstromende watermassa vormt een risico voor gevoelige apparatuur die zich op het dek bevindt en voor de veiligheid van de bemanning.

Figuur 5.1: FPSO in zwaar weer

Om dit risico te minimaliseren is er inzicht nodig in het fenomeen van groen water. Daartoe worden experimenten uitgevoerd waarbij het gedrag van het water wordt bestudeerd en waarbij berekend wordt hoe groot de krachten zijn die het water uitoefent op dekconstructies. In de foto hiernaast is zo'n experiment te zien waarbij het water op het dek tegen een dekhuis botst. De sensoren op het dekhuis meten de druk van het water waar de krachten op het dekhuis vanaf geleid kunnen worden. Experimenten zijn erg kostbaar en daarom is de wens om een deel van het experimentele onderzoek te vervangen door berekeningen. Daarvoor is dan wel een goed simulatieprogramma nodig, dat alle stadia van het groen water probleem kan simuleren. De volgende stadia kunnen worden onderscheiden: (1) De combinatie van een hoge golf en het stampen van het schip leidt tot het opzwellen van water rond de boeg. (2) De muur van water die zo is ontstaan, stroomt het dek op (in een soortgelijke stroming



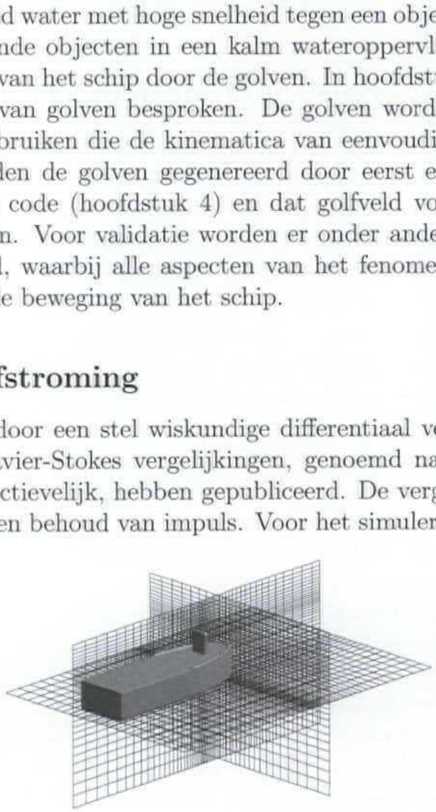
Figuur 5.2: Experiment met groen water op het dek

als bij het breken van een dam). (3) Het water stroomt met hoge snelheid over het dek in de vorm van een tong door de samenkomende stromen van water van de zijkanten. (4) Het water botst tegen een constructie op het dek en slaat uiteindelijk over.

Dit proefschrift beschrijft een methode die deze stadia kan simuleren en de methode wordt gevalideerd door de resultaten te vergelijken met experimenten. In hoofdstuk 2 wordt een basis gelegd voor de complexe groen-water simulaties door waterstroming te simuleren in eenvoudige configuraties. Eerst wordt een brekende dam gesimuleerd, waarbij in een gesloten bak een grote hoeveelheid water met hoge snelheid tegen een object botst. Verder wordt het vallen van verschillende objecten in een kalm wateroppervlak gesimuleerd, als versimpeling van de beweging van het schip door de golven. In hoofdstuk 3 en 4 worden de generatie en voortplanting van golven besproken. De golven worden enerzijds gegenereerd door golftheorieën te gebruiken die de kinematica van eenvoudige golven geven (hoofdstuk 3). Anderzijds worden de golven gegenereerd door eerst een golfveld te laten uitrekenen door een externe code (hoofdstuk 4) en dat golfveld voor te schrijven op de randen van het rekendomein. Voor validatie worden er onder andere resultaten van groen-water simulaties getoond, waarbij alle aspecten van het fenomeen worden meegenomen, zoals ook de golven en de beweging van het schip.

Numerieke simulatie van vloeistofstroming

De beweging van vloeistof wordt beschreven door een stel wiskundige differentiaal vergelijkingen. Deze vergelijkingen heten de Navier-Stokes vergelijkingen, genoemd naar Navier en Stokes die ze in 1823 en 1845, respectievelijk, hebben gepubliceerd. De vergelijkingen zijn gebaseerd op behoud van massa en behoud van impuls. Voor het simuleren van de beweging van een vloeistof, bijvoorbeeld water, dat een onsamendrukbare, viskeuze vloeistof is, moeten de Navier-Stokes vergelijkingen worden opgelost. Alleen in sterk vereenvoudigde vorm kan dat met pen en papier (analytisch) worden gedaan. Daarom worden de vergelijkingen benaderd door ze op een discrete, dat is een stapsgewijze, manier op te lossen. Het vloeistofdomein wordt daarvoor overdekt met een Cartesisch (rechthoekig) rekenrooster, zie Figuur 5.3. In elke cel van het rooster worden de druk en de snelheden van de vloeistof berekend en zo wordt het gedrag van de vloeistof bepaald. Hoe meer cellen er gebruikt worden, hoe beter de simulatie de werkelijkheid benadert. In de methode van dit proefschrift is het rooster vast in de tijd, dus het omstroomde object beweegt door het rooster (dit in tegenstelling tot een 'boundary-fitted' rooster dat aansluit aan de geometrie en met de geometrie meebeweegt). Omdat de geometrie niet rechthoekig is van vorm, snijdt hij door de roostercellen heen waardoor er doorsneden cellen ontstaan.

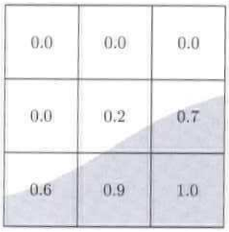


Figuur 5.3: Cartesisch rekenrooster

De vergelijkingen worden ook in de tijd op een discrete manier opgelost, door van tijdstap naar tijdstap te gaan. Hiervoor wordt de expliciete voorwaarts-Euler methode gebruikt. Voor de ruimtelijke discretisatie van de vergelijkingen (d.w.z. voor iedere cel een vergelijking opstellen waarin de variabelen afhangen van de buurcellen) wordt de eindige volume methode gebruikt. De discretisatie wordt zo gedaan dat de onderliggende eigenschappen van de analytische operatoren in de vergelijkingen behouden zijn in de eigenschappen van de discrete operatoren. Dit heeft voordelige gevolgen voor de stabiliteit van de methode: de discrete kinetische energie blijft behouden (afgezien van viskeuze demping) en de methode is stabiel. Alleen in de buurt van de bewegende randen van de geometrie moet een extra stabiliserende term worden toegevoegd.

Verplaatsen van een vrij vloeistof oppervlak

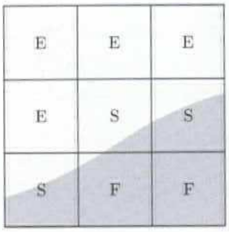
Een belangrijk onderdeel van de soort van simulaties die in dit proefschrift worden gedaan is het bijhouden van de positie van het vrije oppervlak (ofwel het wateroppervlak). Globaal zijn daarvoor twee manieren van aanpak. Bij de eerste aanpak wordt het oppervlak zelf gevolgd door er bijvoorbeeld deeltjes op te leggen en die deeltjes te volgen, of door een hoogtefunctie te gebruiken. Bij de tweede aanpak wordt naar het vloeistofvolume gekeken: voor elke cel van het rooster wordt bijgehouden hoeveel water erin zit en deze waarde wordt elke tijdstap bijgewerkt. Het grote voordeel van de tweede manier is dat wilde vloeistofconfiguraties, zoals overslaande golven, zonder problemen kunnen worden gesimuleerd. In dit proefschrift wordt de VOF-methode (Volume of Fluid) gebruikt die onder de tweede aanpak valt. In deze methode wordt de VOF-functie geïntroduceerd die voor elke cel een waarde heeft tussen 0 en 1. Als de VOF-waarde 0 is, zit er geen vloeistof in de cel; als de VOF-waarde 1 is, is de cel helemaal gevuld met vloeistof; een waarde tussen 0 en 1 geeft een gedeeltelijk gevulde cel aan. Op basis van deze waarden kan het vrije oppervlak worden gereconstrueerd. In de verschillende VOF-methodes die in de literatuur beschreven zijn, zijn daarvoor twee manieren. De eerste is een constante reconstructie, die inhoudt dat het oppervlak evenwijdig loopt met een van de coördinaatrichtingen (dus horizontaal of verticaal in twee dimensies). Deze methode is ontwikkeld door Hirt en Nichols in 1981. De tweede methode, ontwikkeld door Youngs (1987), is de lineaire reconstructie, waarbij het oppervlak ook schuin door een cel heen kan lopen. Deze algemenere reconstructie maakt de methode van Youngs lastiger te implementeren en duurder in rekentijd. Nadat het vrije oppervlak is gereconstrueerd, wordt het verplaatst door nieuwe waarden van de VOF-functie te berekenen op basis van het snelheidsveld. De originele methode van Hirt en Nichols heeft problemen met massabehoud en er ontstaan kleine druppels die los zijn geraakt van het vrije oppervlak (de zogenaamde 'flotsam' en 'jetsam'). Om dat te voorkomen wordt deze methode in dit proefschrift gecombineerd met een lokale hoogtefunctie, wat ervoor zorgt dat de vloeistof bij elkaar blijft en massa behouden is. Ook Youngs' methode kan worden gecombineerd met de lokale hoogtefunctie.



Figuur 5.4: VOF-waarden

In dit proefschrift worden de methodes van Youngs en van Hirt-Nichols met elkaar vergeleken. In testen waar het snelheidsveld wordt voorgeschreven (en niet berekend uit de Navier-Stokes vergelijkingen) levert Youngs duidelijk nauwkeuriger resultaten. Bij 'echte' sommen, waar ook het snelheidsveld berekend wordt, zoals een brekende dam of de simulatie van een golf, is Youngs iets nauwkeuriger, maar het maakt dan niet veel uit. Beide methodes kunnen het best gecombineerd worden met de lokale hoogtefunctie om de hoeveelheid massa te behouden.

Randvoorwaarden op het vloeistofoppervlak
Op het vrije oppervlak moeten randvoorwaarden worden gekozen. In Figuur 5.5 is een stukje van het rekendomein te zien met de labels van de verschillende cellen: S(urface)-cellen, E(mpty)-cellen en F(luid)-cellen. Snelheden zijn gedefinieerd op celwandjes, en er zijn onder andere randvoorwaarden nodig voor de snelheden tussen S en E-cellen. Het is gebleken dat de keuze van deze SE-snelheden veel invloed heeft op verschillende aspecten van de simulatie, namelijk op de robuustheid van de code, op het wel of niet ontstaan van numerieke pieken in het drukveld en op de nauwkeurigheid van de oplossing. In het proefschrift worden twee methodes beschreven voor het definiëren van de SE-snelheden. In de eerste wordt massabehoud geëist in de S-cel. Het voordeel van deze methode is dat er geen drukpieken ontstaan, maar de nadelen zijn dat het onnauwkeurige resultaten geeft wanneer er een golfstroming wordt berekend en de methode is niet robuust doordat er rekening moet worden gehouden met doorsneden cellen. In de tweede methode worden de SE-snelheden geëxtrapoleerd van snelheden in het inwendige van de vloeistof. Deze methode is robuust en nauwkeurig, maar veroorzaakt pieken in de druk wanneer een S-cel verandert in een F-cel. Een combinatie van deze beide methodes is gebruikt om de voordelen van beide methodes mee te nemen. In het proefschrift wordt dit uitgelegd en worden de beweringen gestaafd met simulatieresultaten.

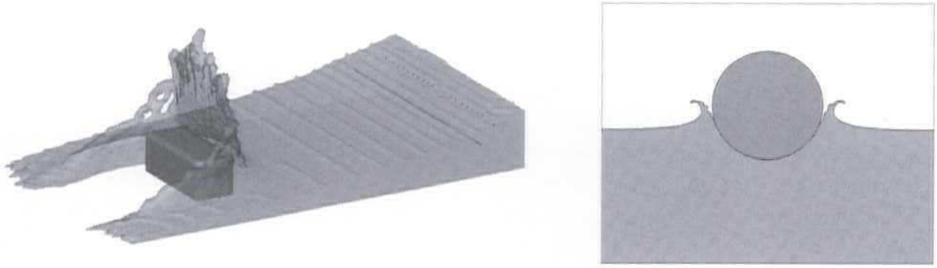


Figuur 5.5: cel labels

Validatie
Het tweede hoofdstuk wordt afgesloten met een aantal validatiesimulaties om aan te tonen dat onze methode geschikt is voor het soort simulaties waarvoor we het ontwikkelen (zie Figuur 5.6). Eerst worden de resultaten gepresenteerd van een brekende dam, waar een blokje in de stroming is geplaatst dat model staat voor een container op het dek van een schip. De resultaten worden vergeleken met experimenten en tonen een goede overeenkomst daarmee. Hierna wordt het gedrag van water gesimuleerd wanneer er een object invalt. In twee dimensies laten we een cilinder en een wig vallen en in drie dimensies een kegel. Het vrije oppervlak en de berekende krachten die op het vallende object worden uitgeoefend, worden vergeleken met experimenten en analytische voorspellingen. De overeenstemming van de resultaten van deze simulaties geven vertrouwen in de methode.

Numerieke simulatie van watergolven

Er zijn verschillende methodes om golven te genereren in een numerieke berekening. In de eerste methode worden golven opgewekt door de kinematica die volgt uit een golftheorie voor te schrijven op de instroomwand van het rekendomein (hoofdstuk 3). In de tweede methode (die in hoofdstuk 4 wordt geïntroduceerd) wordt een snel en efficiënt extern programma gebruikt om het golfveld te berekenen. Dit golfveld wordt dan voorgeschreven op de randen van het rekendomein. Deze tweede methode is een voorbeeld van zonale modellering, omdat het domein als het ware in twee zones wordt verdeeld: de buitenzone



Figuur 5.6: Links: simulatie van een brekende dam met een modelcontainer in de stroming; rechts: vallende cilinder

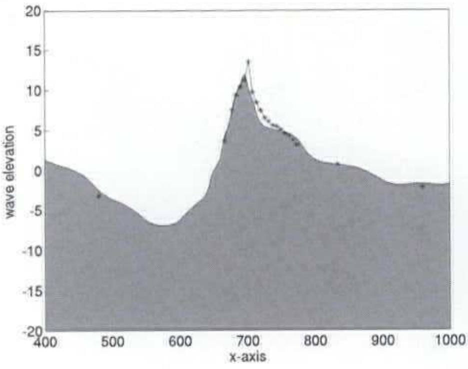
waar het golfveld wordt berekend en een binnenzone waar onze methode wordt toegepast. Een derde methode om golven te genereren is om golfopwekkers te gebruiken zoals ook in experimentele bassins, door de golfflappen als een bewegend object te modelleren.

Voortplanting van golven

In hoofdstuk 3 wordt de voorplanting van golven in het rekenprogramma bestudeerd door regelmatige en steile onregelmatige golven door een twee-dimensionaal rekendomein te laten lopen. De golven worden opgewekt door op de instroomwand van het rekendomein watersnelheden en de waterhoogte voor te schrijven die volgen uit een golftheorie. In dit proefschrift worden lineaire theorie (voor regelmatige en onregelmatige golven) en vijfde orde Stokes theorie gebruikt. Aan de wand tegenover de instroomwand, moet de golf het rekendomein verlaten zonder de golf in het domein te beïnvloeden, dus alsof de golf geen wand ziet. Het is lastig deze randvoorwaarden te bepalen, omdat er niet direct informatie over de golf voorhanden is. Een bekende methode om een golf netjes het rekendomein te laten uitgaan is de Sommerfeld conditie, waar de golfvergelijking wordt gediscretiseerd om snelheden op de uitstroomwand te bepalen. In de Sommerfeld conditie moet de golfsnelheid van te voren worden bepaald, waardoor dus vooraf kennis over de golf nodig is. Deze methode werkt het best bij regelmatige golven die niet te erg zijn verstoord. Een andere methode is het gebruik van een dissipatiezone: een verlenging van het rekendomein waar de golf wordt gedempt. Deze methode kan ook bij onregelmatige golven gebruikt worden. Het nadeel is dat er meer roostercellen nodig zijn door de verlenging van het domein. Beide methodes worden in het proefschrift gebruikt.

In de studie naar de voorplanting van golven wordt ook gekeken naar het effect van verlies van energie door keuzes in het numerieke model. Door het gebruik van de upwind methode voor de convectieve term in de Navier-Stokes vergelijkingen wordt extra viscositeit toegevoegd die energie onttrekt aan de stroming en een dempend effect heeft op de golven. Voor een kortdurende simulatie van een paar periodes in een klein domein (het soort situaties dat is beoogd in dit proefschrift) is het dempend effect van de extra viscositeit echter niet zo sterk.

Voor validatie van de methode worden regelmatige en onregelmatige golven door het domein gestuurd en wordt de berekende golfhoogte vergeleken met een golftheorie of experiment. In Figuur 5.7 is een plaatje te zien van de golfhoogte van een onregelmatige steile golf. De berekende golfhoogte wordt vergeleken met gemeten waarden, aangegeven met asterisken. De overeenkomst is over het algemeen goed, alleen wordt de golfhoogte bij steile golven niet helemaal bereikt. Dit komt door de methode die wordt gebruikt voor de vrije-oppervlak verplaatsing (Youngs' methode geeft nog betere resultaten) en door dissipatie van energie. Verder is de methode gevalideerd door offshore constructies in een golfveld te plaatsen. De golfhoogtes in de buurt van de constructies en de krachten op de constructies zijn vergeleken met metingen. Als constructies zijn een spar platform (een cilindervormig platform) en een FPSO gebruikt, waarbij er groen water optreedt op het dek. De overeenkomst met de metingen is naar tevredenheid.

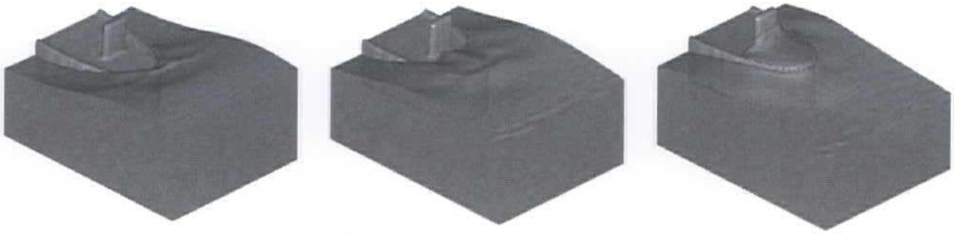


Figuur 5.7: Een momentopname van de berekende vorm van een steile golf; de asterisken geven de metingen weer

Zonale modellering

In hoofdstuk 4 is een begin gemaakt met het ontwikkelen van een methode, die zonale modellering gebruikt voor de berekening van het golfveld. Bestaande methodes, zoals een diffractie methode, zijn erg efficiënt voor de berekening van het golfveld in combinatie met de beweging van het schip, maar de meer extreme stroming van groen water of brekende golven kunnen deze methodes niet berekenen. Daarvoor is de relatief dure methode uit dit proefschrift beter geschikt. Bij zonale modellering gebruiken we de voordelen van beide methodes door alleen in een klein domein rond (een gedeelte van) de constructie de Navier-Stokes vergelijkingen op te lossen. De randen van dit lokale domein en het golfveld aan het begin van de simulatie worden voorgeschreven vanuit een extern programma, bijvoorbeeld een lineaire diffractie code. Een bijkomend voordeel is dat ook de beweging van het schip door het externe programma wordt voorspeld. Deze voorspelling wordt dan gebruikt in de Navier-Stokes berekening.

Deze gekoppelde methode (waarbij een lineaire diffractiecode is gebruikt voor de externe code) wordt toegepast op drie simulaties. Eerst wordt een onregelmatige golf gesimuleerd, zonder object in de stroming. Zolang de golf niet al te hoog is, komen de lineaire theorie en de berekende golfhoogte uit onze methode perfect overeen. Voor hogere golven ontstaan te verwachten verschillen met de lineaire theorie. Doordat de lineaire diffractieresultaten die aan de randen worden voorgeschreven niet meer goed overeenkomen met het berekende golfveld, ontstaan er reflecties van de randen die de interne golf verstoren. De invloed van zulke reflecties is ook zichtbaar in de tweede simulatie, waar krachten op een spar platform worden berekend. In de derde simulatie wordt de gekoppelde methode gebruikt voor een simulatie van groen water over de boeg van een FPSO (zie Figuur 5.8). De berekende waterhoogte op het dek en de drukkrachten op het dek en een dekhuis zijn vergeleken met metingen van experimenten. De overeenkomst tussen metingen en berekeningen is goed, wat vertrouwen geeft in een verdere ontwikkeling van de gekoppelde methode.



Figuur 5.8: Een simulatie met groen water op het dek van een FPSO; de golven en de beweging van de FPSO zijn geïnitieerd door een externe code

Epiloog

In dit proefschrift hebben we laten zien dat de ontwikkelde methode geschikt is voor de simulatie van complexe stromingen met vrije wateroppervlakken, zoals groen water op het dek van een schip. Toch is er nog veel te verbeteren aan de methode om nauwkeuriger resultaten te behalen en de methode nog algemener te kunnen toepassen. Eén richting van verbetering is om niet alleen berekeningen te doen in het water, maar ook de stroming in de lucht te berekenen (een twee-fase methode). Daarmee verdwijnt het probleem van de randvoorwaarden voor de snelheid aan het vrije oppervlak en kan het fenomeen van luchtinsluiting tijdens de impact van een golf worden meegenomen. Verder kan de zonale modellering worden uitgebouwd, door niet-lineaire potentiaalmethodes te gaan gebruiken en slimmere koppelingen te maken over de randen van het lokale (Navier-Stokes) domein. Aan deze aspecten wordt aandacht besteed in een vervolgproject.

Dankwoord

Nu mijn proefschrift bijna bij de drukker ligt, wil ik van de gelegenheid gebruik maken een aantal mensen te bedanken zonder wie dit werk niet tot stand was gekomen.

Allereerst natuurlijk mijn promotor, Arthur Veldman. Jij kwam met het idee bij me om een promotieonderzoek te gaan doen. Zonder de duw in die richting was ik niet als AiO aan het werk gegaan. Ook tijdens de jaren van het onderzoek had je er altijd veel vertrouwen in en kwam je op de goede momenten met nieuwe ideeën en kwam steeds weer met verbeteringen voor de te schrijven stukken. De deur van je kamer staat altijd open en dat wordt door ons AiO's erg gewaardeerd.

Further, I would like to thank the reading committee, prof. Nigel Barltrop, prof. Barry Koren and prof. Jo Pinkster, for carefully reading my thesis and giving suggestions for improvement.

Also the members of the SafeFLOW project contributed to my thesis a lot. The nice meetings with lots of good suggestions have always stimulated me to proceed with the research project. Special thanks to Bas Buchner and Tim Bunnik of MARIN and Bogdan Iwanowski and Erik Falkenberg of FORCE Technology for the good cooperation in the ComFLOW part of the project. Bogdan, thanks for all the helpful tips and the ideas for validation, I really appreciated your enthusiasm. MARIN (Arjan Voogt) is acknowledged for providing validation material by means of measurement data.

Ik wil mijn kamergenoten, Marc en Conny, bedanken voor de gezelligheid in de afgelopen jaren. Of het werk nu wilde vlotten of niet, in onze kamer was het (bijna) altijd goed toeven. Marc, speciaal ook bedankt voor alle inhoudelijke hulp: voor numerieke vragen, latex vragen, matlab vragen en ga zo maar door, hoefde ik nooit ver voor hulp. Ook het systeembeheer wil ik hier bedanken voor het altijd klaarstaan met ondersteuning van het IT-deel. Verder, alle (ex-)TMNW-collega's, Hendrik, Arthur, Eugen, Fred, Roel, Kurt, Ena, Roel, Jeroen, Edith, Erwin, Geert, Marc, Dirk-Jan, Arie, Rik, Gerk en Konstantin: bedankt voor de goede discussies en de goede sfeer. Ook een aantal afstudeerders, in het bijzonder Geert Meskers, Michael Ten Caat en Joost Heemskerk, wil ik bedanken voor hun bijdrage

Voor het stuk afleiding van het werk wil ik de (ex-)AiO's van harte bedanken voor de leuke lunches en theepauzes. In de avonduren en de weekenden stond mijn basketbalteam altijd klaar voor een gezonde portie beweging en gezelligheid. De grootste afleiding in het laatste AiO-jaar is (de komst van) Jelte geweest, waarvoor ik heel dankbaar ben.

Tot slot, dit werk was zeker niet gelukt zonder de voortdurende steun van Kristian, mijn ouders en verdere familie. Kristian, bedankt voor het luisteren naar mijn verhalen en meedenken met mijn onderzoek. Het vertrouwen dat je in mij had is erg belangrijk geweest voor het slagen van dit project.