Dynamics of Liquid-Filled Spacecraft

Numerical Simulation of Coupled Solid-Liquid Dynamics

Jeroen Gerrits

Rijks*universiteit* Groningen

Dynamics of Liquid-Filled Spacecraft

Numerical Simulation of Coupled Solid-Liquid Dynamics

Proefschrift

ter verkrijging van het doctoraat in de
Wiskunde en Natuurwetenschappen
aan de Rijks*universiteit* Groningen
op gezag van de
Rector Magnificus, dr. D.F.J. Bosscher,
in het openbaar te verdedigen op
vrijdag 21 december 2001
om 14.15 uur

door

Jeroen Gerrits

geboren op 9 maart 1972
te Veendam

Promotor:	Prof. dr. A.E.P. Veldman
Referenten:	Dr. ir. R.W.C.P. Verstappen
	Dr. ir. J.P.B. Vreeburg
Beoordelingscommissie:	Prof. dr. ir. H.W. Hoogstraten
	Prof. dr. ir. J.A.M. Kuipers
	Prof. dr. ir. P. Wesseling

ISBN 90-367-1477-x

The research presented in this thesis was funded by the Netherlands Organisation for Scientific Research / Space Research Organisation in the Netherlands NWO/SRON.

The cover shows an artist impression, created by the Graphic Design Group of the Dutch National Aerospace Laboratory NLR, of the experiment satellite SloshSat FLEVO flying in space.

Contents

Chapter 1

Introduction

1.1 Liquid Dynamics in Spacecraft

Since the launch of the first satellite Sputnik on October 4, 1957, a lot of satellites have been launched for scientific or commercial reasons. Nowadays, satellites carry relatively more fuel (necessary for course corrections) than in the beginning of the space era. This has consequences for the controllability of satellites. Indeed, when thrusters are fired for course correction, the onboard fuel starts to accelerate inducing a force and torque on the satellite, whence the satellite starts to accelerate, and so forth.

This interaction between the motion of the satellite and the onboard sloshing liquid can have undesirable consequences as happened quite recently (in 1998) with NASA's Near Earth Asteroid Rendezvous (NEAR) craft, which was on its way to the asteroid 433 Eros [38, 93] (see figure 1.1). A propellant burn that would put the spacecraft on track was aborted after sensors detected accelerations that exceeded limits programmed into its onboard computer. Further investigation revealed that the spacecraft was tumbling, which eventually caused a 13 months delay in the mission.

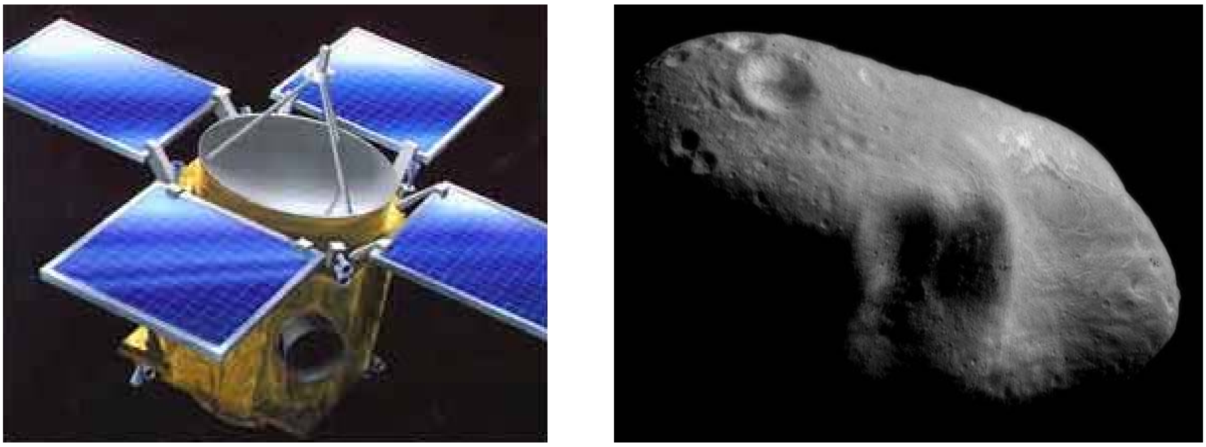


Figure 1.1: *Left: Mock-up of the Near Earth Asteroid Rendezvous (NEAR) spacecraft. Right: Picture of asteroid 433 Eros taken by the NEAR spacecraft. These pictures were taken from http://www.space.com/php/multimedia/imagegallery/archive.php and http://antwrp.gsfc.nasa.gov/apod/ap010211.html respectively.*

Controllability of spacecraft plays also an important role in the operation of the International Space Station (ISS). When the Space Transportation System (STS), better known as the Space Shuttle, docks the ISS, small manoeuvres due to sloshing of fuel or liquid aboard the STS are highly undesirable.

These examples demonstrate the importance of increasing the knowledge in controllability of spacecraft, in particular the interaction between the liquid dynamics and the dynamics of the spacecraft. An important aspect of these fluid-flow problems in space is the absence of gravitational effects. Hence, capillary effects, such as wall adhesion and surface tension, can not be neglected [91].

There are three possible methods to study the fluid-flow problems that have been described above, namely

- theoretical,
- experimental,
- numerical.

The equations that govern the motion of a liquid are known for more than 150 years and named after Navier (1823) and Stokes (1845). However, except for some simplifications, the Navier-Stokes equations can not be solved analytically.

Experimental methods have been, and still are, very popular for studying liquid motion. However, if capillary forces play an important role, experiments are rare since experiments in space are very expensive. Hence, the launch of the experiment satellite SloshSat FLEVO (an acronym for Facility for Liquid Experimentation and Verification in Orbit) in a few years is very welcome (see figure 1.2 and the cover of this thesis). For SloshSat, the Dutch National Aerospace Laboratory NLR is the main contractor. SloshSat is a small, free-flying satellite that is launched from the Space Shuttle. Its experiment tank is a cylindrical container with two hemispherical ends having a volume of nearly 87 litres. The tank is partially filled with approximately 33 litres of water, representing a

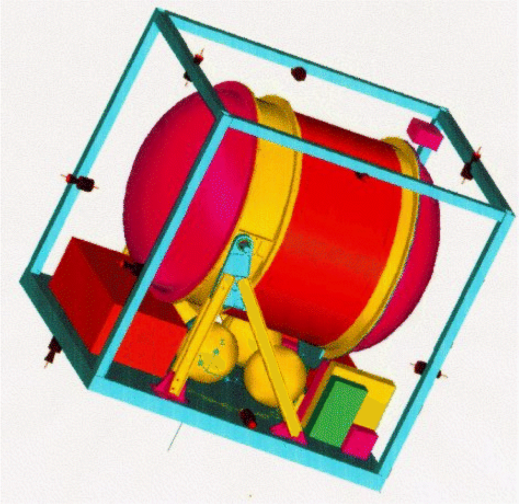


Figure 1.2: *Mock-up of the experiment satellite SloshSat FLEVO.*

liquid mass that will be 26% of the total mass of the spacecraft. The satellite will perform various manoeuvres for studying liquid dynamics in a micro-gravity environment and the interaction between the liquid motion and the motion of the spacecraft [87, 88]. Of course, micro-gravity experiments on earth are possible, but micro-gravity can be obtained for a few seconds only (for example in parabolic flight or in a drop tower). Thus the flight of SloshSat, which will take approximately 48 hours, is potentially very valuable for better understanding of liquid dynamics and coupled solid-liquid dynamics under extra-terrestrial conditions.

The third method, numerical simulation, for studying liquid dynamics is known as Computational Fluid Dynamics (CFD) and becomes increasingly popular. Not only because of the huge growth in computer resources, but also due to the development of numerical algorithms, simulations of complex fluid-flow problems are feasible and reliable nowadays. Especially for fluid flow in a micro-gravity environment, where theoretical and experimental methods are rare, numerical simulations play a crucial role in better understanding of the flow phenomena.

In this thesis a numerical method is developed (which has been implemented in a computer program called ComFlo) for the simulation of fluid flow in a micro-gravity environment. Fluid flow in partially-filled tanks under extra-terrestrial conditions is characterised by low values of the Bond number Bo, which is a dimensionless number indicating the relative importance of gravitational forces compared to capillary forces, and the Weber number We, indicating the relative importance of inertial forces compared to capillary forces. Hence, much attention is given to the modelling of effects of capillarity. The model is developed in three dimensions and complex-shaped flow domains are incorporated; both being crucial ingredients for accurate simulation of extra-terrestrial flows. Also included in the model is the interaction between the liquid dynamics and the solid-body dynamics (see [89], and the many references therein, for more information about coupled solid-liquid dynamics in a micro-gravity environment). As a spin-off, ComFlo has been used for simulating terrestrial fluid-flow problems (*i.e.* for high Bo) [14, 20].

1.2 Computational Grid

One of the first choices that has to be made for the development of a numerical model is the type of computational grid. Basically, a computational grid has two characteristics: the grid is structured or unstructured and it is boundary fitted or non-boundary fitted.

In a structured grid, all the grid cells have the same number of cell faces and the number of cells surrounding each grid point is constant. An orthogonal grid is a structured grid with the extra feature that all the cell faces meet at right angles. Such a grid is usually called a Cartesian grid. If either the number of cell faces per grid cell or the number of grid cells surrounding a grid point is not constant, then the grid is called unstructured. In the left of figure 2.3, in which the circle is assumed to be the flow domain, an example of an unstructured grid is shown. In the middle and in the right of this figure, two examples of structured grids are drawn.

A computational grid is called boundary fitted if the boundary of the flow domain coincides completely with cell faces of the grid [32]. In a non-boundary-fitted grid, some computational cells may be cut by the boundary of the flow domain (such cells are called cut cells) [11, 53, 75, 82]. The two grids in the left and the middle of figure 2.3 are boundary fitted, the Cartesian grid in the right of this figure is non-boundary fitted.

In general, it is extremely difficult to generate structured, boundary-fitted grids for arbitrary complex-shaped flow domains (especially in three dimensions). Even the generation of qualitative good unstructured grids — in a qualitative good triangular grid, for example, the angles between grid lines should not be too small — is not a trivial task and is often more time consuming than the flow simulation itself. This is in huge contrast to the generation of a Cartesian grid, which takes negligible time with respect to the remaining part of the flow simulation.

An advantage in using unstructured grids is the ability to generate grids for very complex-shaped flow domains (also in three dimensions). Further, it is relatively easy (compared to structured grids) to obtain a finer mesh size in regions of interest, *e.g.* in sharp corners of the flow domain or in regions with high gradients in the velocity field. Refinement of a structured (but non-Cartesian) grid to obtain a desired mesh size in a certain region of the flow domain is often very difficult. For Cartesian grids, it is possible to use (global) stretching to obtain a larger mesh resolution in certain areas [53]. However, since the mesh size in one Cartesian direction is a function of this direction only, the mesh size may become small also in regions where such a fine grid is not required. In this case, local refinement of a Cartesian grid is more appropriate and relatively easy to accomplish [97]. However, discretising the governing equations on a locally refined grid, such that momentum and kinetic energy are conserved, is not trivial.

An important advantage of boundary-fitted grids is the easy discretisation of the boundary conditions. Indeed, discrete boundary conditions are applied directly at the boundary of the flow domain, which coincides with cell faces. If grid lines are not aligned with the boundary, then special care has to be taken in discretising the boundary conditions in cut cells. Moreover, cut cells may become arbitrarily small, which can result in severe time-step restrictions. To overcome these restrictions, it is possible to redistribute grid cells; small cells are treated simultaneously with neighbouring uncut cells [1, 95]. Another option is to permit the fluid to flow through more than one computational cell per time step [48]. Time-step restrictions originate because of an explicit discretisation of the convective and diffusive terms in the Navier-Stokes equations. However, it can be shown that the convective terms do not worsen this time-step restriction in the presence of small cut cells. Only the diffusive terms cause problems in this situation [79]. By treating diffusion implicitly, these problems can be overcome [7]. Despite the difficulties encountered in using cut cells, this approach often improves the results compared to a staircase approximation of the flow domain [34, 50].

One aspect that has to be taken into account in choosing the grid topology is the desire to simulate interface flow or free-surface flow. Since many of the numerical models for advecting interfaces are based on geometrical observations, a structured grid is more practical for this type of simulations. In particular much research has been done on the simulation of interface flow on Cartesian grids. Simulation of interface flow on unstructured grids, however, is feasible [49, 74].

An interesting technique to account for complex geometries on Cartesian grids is described in [84], where the solid boundary is treated as a free surface. A boundary condition for the pressure is applied here, such that liquid can not flow in a direction normal to the solid boundary.

1.3 Discretisation Methods

Once a choice for the type of grid has been made, the governing equations can be discretised on this grid. For this, two further choices have to be made. First, the positioning of the variables or control volumes has to be determined. Second, the discretisation method has to be chosen.

For incompressible fluids, only the pressure p and three components u, v, and w of the velocity vector $\boldsymbol{u} = (u, v, w)^T$ are required to describe the flow (in the presence of an interface, one or more variables are needed to describe its location, but this will be explained in section 1.4). Usually, in incompressible computational fluid dynamics, the pressure is positioned at cell centroids, while velocities are positioned at cell-face centroids. This positioning is known as the marker-and-cell (MAC) positioning [29] and prevents a decoupling in the pressure when it is solved from a Poisson equation. The positioning of the velocities at cell faces is called a staggered positioning with respect to the pressure. Another option is to position all variables at cell centroids; a collocated positioning [3, 75]. This approach is popular in combination with three-dimensional, unstructured grids since this leads to a simplification in the implementation. Apart from a collocated and a staggered positioning, all kinds of hybrid methods are imaginable. However, no particular method is optimal in the sense that it performs superior in all possible simulations [58, 68].

Roughly speaking, a discretisation method can be classified as one of the following:

- spectral method,
- finite-element method (FEM),
- finite-difference method (FDM),
- finite-volume method (FVM).

In spectral methods [13], the unknown velocity is written in terms of a number of global basis functions, a Fourier series for example, whence high-order approximations to spatial derivatives are easily feasible. However, these methods are most suitable for flow domains with simple geometries and periodic domains.

Usually, on an unstructured grid, the finite-element method is applied [28, 51, 54, 55]. Grid cells or control volumes are then called elements. On each of these elements a basis function (*e.g.* an interpolation function), say U_i, where $i = 1, \ldots, N$ runs over the elements, is defined. The variable, say u, that has to be solved from the governing equations is written as a linear combination of these basis functions, *i.e.* $u = \sum_{i=1}^{N} c_i U_i$. This finite series is then substituted in an integral or weak form of the governing equations, resulting in a linear system for the coefficients c_i. The finite-element method is in particular suitable for solving parabolic or elliptic partial differential equations. For hyperbolic equations, for example the equation for advecting an interface (see section 1.4), almost no research has been done using the finite-element method.

Contrary to the finite-element method, the finite-difference method starts from the differential or strong form of the governing equations [29, 44, 46, 52]. By using finite-difference approximations of the derivatives and linearisation of nonlinear terms, the partial differential equations are replaced by a system of linear equations. Mostly, a finite-difference method is used on structured, in particular Cartesian, grids. On these grids, derivatives are easily approximated using Taylor-series expansions. A well-known finite-difference method is the MAC method, where the governing equations are discretised on a fixed, Cartesian mesh and markers are used for tracking a free surface [29, 39, 83].

Often, the finite-volume method [41] is thought of being similar to the finite-difference method, which can be explained by the fact that, after using the finite-volume method, the discretised equations can be rewritten in a form that looks similar to the discretised equations that result from the finite-difference method. In fact, the finite-volume method is more related to the finite-element method since it starts from a weak formulation of the governing equations. In the finite-volume method, this formulation is also called conservative, expressing in the momentum equation, for example, that an increase of momentum in a control volume is due to a net inflow through the boundary of that control volume. By choosing control volumes such that every control-volume face belongs to exactly two control volumes, a fully conservative numerical model is easily achieved [7]. Indeed, by computing mass and momentum fluxes at control-volume faces, the amount of mass and momentum that leaves one control volume is gained in an adjacent control volume. Finite-volume methods are attractive since rigorous conservation of mass and momentum is considered more important than the formal accuracy of the discretisation method [47, 76]. Further, it is possible to discretise the Navier-Stokes equations such that symmetry properties of the continuous differential operators in these equations are inherited by the discrete difference operators, which is advantageous for the stability of the numerical method [82].

1.4 Interface Flow

An important feature in many problems in fluid dynamics is the presence of an interface: a separation between two (or more) fluid phases. In general, the location of the interface is not known in advance and is part of the problem that needs to be solved. An important class of interface flows is formed by the free-surface flows, in which the density and molecular viscosity of one phase are much smaller than those of the other phase, *e.g.* air and water. For free-surface flows, it is often possible to simulate only the dynamics of the heavier phase (water) since the motion of the lighter phase (air) has negligible effect on it [6, 35, 47, 52]. However, applications exist in which the dynamics of the air can not be neglected [57, 92]. For simulation of interface flows, two aspects have to be taken into account, namely the advection of the interface and the application of boundary conditions at the interface. Comprehensive reviews of interface advection methods can be found in [43, 45, 63, 66].

1.4.1 Advection Methods

Advection algorithms can be classified as one of the following two types:

- tracking,
- capturing.

In tracking methods, discrete points, say x_i, positioned on the interface are tracked. This is done by integrating the equation

$$\frac{dx_i}{dt} = u_i,$$

where u_i is the velocity at x_i. Thus, tracking methods are Lagrangian methods for interface advection.

The points x_i can coincide with points of the computational grid. In this case the grid deforms in time and the method is called a moving-mesh method [21, 28, 54]. A drawback of a moving-mesh method is the possibly large deformation of grid cells due to considerable shear or vorticity in the velocity field.

Another way of tracking the interface is by moving (massless) particles, distributed along the interface, through a stationary grid [52, 60, 75]. Such a method is usually called a front-tracking method [25]. In this case actually two grids are part of the numerical model: an Eulerian grid on which the equations governing the fluid dynamics are solved and a Lagrangian grid (that is in general one dimension lower than the Eulerian grid) for tracking the interface [55]. In front-tracking methods, the grid points distributed over the interface may get deformed. However, since the Lagrangian grid in these methods is two-dimensional at most, it is often feasible to add, remove, or reconnect grid points if the Lagrangian grid gets too deformed [51]. This method is also exploited in [10], where subgrid information is used for converging fluid fronts.

In capturing methods, the interface is not tracked explicitly, but is captured or reconstructed from Eulerian data. In a two-phase flow, for example, a so-called colour function C is introduced in every computational cell by

$$C = \begin{cases} C_1 & \text{in phase 1,} \\ C_2 & \text{in phase 2,} \\ > C_1 \text{ and } < C_2 & \text{at the interface,} \end{cases}$$

where it is assumed that $C_1 < C_2$. The interface is then defined as the transition region $C_1 < C < C_2$. From this it follows that the interface has finite width. In multi-phase flows a well-known colour function is the density [41]. The evolution of the colour function is given by

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

stating that points in one phase remain there. In this equation $\boldsymbol{u}$ is the liquid velocity. The main problem of capturing methods is that the interface has finite width (in the order of the mesh size).

The most intuitive way of advecting the interface in a capturing method is to discretise equation (1.1) using a finite-difference or finite-volume method. This is called a continuum advection method and assumes that the colour function C is smooth. However, in interface flow and in the limit of zero mesh size, the function C is discontinuous over the interface. Hence, continuum advection schemes have difficulties in advecting interfaces accurately. Typically, these methods diffuse the interface, broadening the width of it to several mesh cells.

Obviously, the main problem encountered in the continuum advection method can be solved by introducing a colour function that varies smoothly throughout the liquid, in particular across the interface. This is the key idea of a level-set method [70, 72]. A function $\varphi(\boldsymbol{x}, t)$ is introduced, such that $\varphi(\boldsymbol{x}, 0)$ denotes the signed distance (positive in fluid 1 and negative in fluid 2) of the point $\boldsymbol{x}$ to the interface at time $t = 0$. The interface is then given as the zero level set of the level-set function φ, which is evolved in time by

$$\frac{\partial \varphi}{\partial t} + u_n |\nabla \varphi| = 0, \tag{1.2}$$

where $u_n = \boldsymbol{u} \cdot \boldsymbol{n}$ is the normal velocity of the interface. The normal $\boldsymbol{n}$ is computed as $\boldsymbol{n} = \nabla\varphi / |\nabla\varphi|$. Since the level-set function is a smooth function, it can be advected using standard discretisation of equation (1.2). In general, $\varphi(\boldsymbol{x}, t)$ is not a distance function for all time t (*e.g.* if the velocity field contains shear). The standard level-set method has serious problems with respect to mass conservation. In [71] the level-set method is combined with a VOF method (see hereafter) in order to overcome problems with mass conservation. By using a re-distancing algorithm [69], the standard level-set method can be improved as well.

The most popular capturing method for advecting interfaces is the volume-of-fluid (VOF) method introduced by Hirt and Nichols [35]. In this method, a VOF function F (with values between zero and one) is introduced, indicating the fractional volume of a computational cell that is filled with a certain phase (in a two-phase fluid problem $1 - F$ then denotes the fractional volume of the cell filled with the other phase). The evolution of the VOF function is given by $DF/Dt = 0$. The main difference between the VOF method and other capturing methods is the treatment of the nonlinear advection term in this equation. While continuum advection methods and level-set methods use a discretisation of this term, VOF methods treat it geometrically. In every computational cell the interface is reconstructed. Hereto, several methods can be applied, *e.g.* a piecewise-constant reconstruction, where the interface is aligned with one of the Cartesian directions [44], or a piecewise-linear reconstruction, where the interface is allowed to vary linearly in a cell (see [61] for a comprehensive review of reconstruction methods). Based on the reconstructed interface and the velocity field that has been computed from the governing equations, fluxes are computed at cell faces and fluid is transported from one cell (donor cell) to an adjacent cell (acceptor cell). The method for advecting a linear reconstruction of the interface was introduced by Youngs [96]. Often, the fluxing of fluid from donor cells to acceptor cells is split for the Cartesian directions [62], whence great care has to be taken to keep the values of the VOF function at the end of a time cycle between zero and one in order to prevent gain or loss of mass (usually VOF values below zero and above one are set to zero and one respectively at the end of a time cycle, herewith destroying mass conservation). Unsplit or multi-dimensional methods have less problems with undershoots or overshoots in the VOF function, but are far more complex [74]. Many variations of Youngs' method have been studied, all using a linear reconstruction of the interface. Although these methods are generally more complicated than Youngs' original method, they do not give superior results [2, 27, 30, 31, 42, 63].

For validating advection algorithms, several benchmark problems have been reported by Rider and Kothe [61]. All these problems use a velocity field that is prescribed analytically, rather than being computed from the equations governing the liquid dynamics. Hence, these benchmark problems are not always sufficient for evaluating advection methods.

1.4.2 Boundary Conditions

At the interface a boundary condition is needed for the pressure. Moreover, if only one phase is modelled since the motion of the other phase has negligible influence (*e.g.* in free-surface flows), boundary conditions are needed for the velocity as well [9, 36]. In particular the boundary condition for the pressure is complicated if capillary forces have to be taken into account, like in fluid flow in a micro-gravity environment. In this case the pressure at the interface depends on its curvature (and of course on the surface tension of the fluids). The mean curvature κ of the interface is given by

$$\kappa = \nabla \cdot \boldsymbol{n}, \tag{1.3}$$

where $\boldsymbol{n}$ is the normal of the interface. Several methods for computing the curvature can be applied (see for example [59]). Of course, equation (1.3) can be discretised using finite differences. However, since the normal is only defined at the interface, estimating the normal accurately is a nontrivial task, whence computing the curvature using a discrete version of equation (1.3) may be highly inaccurate. If the interface is single valued, a height function can be defined [8, 52], from which the curvature can be computed more accurately. In fluid flow where the topology of the interface changes dramatically, it is not possible to define a global height function (*i.e.* throughout the entire flow domain). In this case a local height function (*e.g.* in every computational cell) can be defined for estimating the curvature of the interface as is shown in chapter 2 of this thesis.

Another method for incorporating surface tension (instead of applying a boundary condition at the free surface) is to add a surface force to the Navier-Stokes equations. This is a volume integral containing a delta function and the surface-tension force. The delta function is zero everywhere except at the interface. For an accurate discretisation of this integral, the interface is broadened to a width of approximately three computational cells, for which various techniques can be used [46, 90]. This method for incorporating surface tension is called the continuum surface force (CSF) method [5, 44, 63]. Adding a surface force to the momentum equations is an elegant method in the sense that the boundary condition for the pressure is treated simultaneously with the governing equations. However, smearing the interface, necessary for obtaining accurate results, is used because of numerical reasons and does not link up to the physical properties of a discontinuous interface (see also [33, 60]).

For computing the curvature (1.3) of the interface a boundary condition is needed at the intersection of the solid boundary and the interface. This intersection is called the contact line and is, in three dimensions, a one-dimensional subset of the flow domain. For the boundary condition at the contact line basically two methods are available. The first method assumes a static contact line. In this method the position of the contact line is fixed. Hence, the angle between the solid boundary and the interface may change in time [6, 17, 18]. The second method prescribes a static contact angle in which the interface intersects the solid boundary at a fixed angle. As a consequence, in this method, the position of the contact line varies in time [5, 16, 40, 55]. For certain combinations of material properties (of fluid, air, and solid boundary), assuming a static contact angle seems to be correct. For other combinations a dynamic contact angle, where the angle depends on the velocity (magnitude and sign) of the contact line, seems to be more appropriate. Apart from these two methods, hybrid methods are possible (see for example [91]). Presently, the physics of the contact line is not well understood yet and a lot of research is being done on how to model contact-line dynamics correctly. In this thesis contact-line dynamics is modelled by a static contact angle. Although this may lead to a transient behaviour of the liquid that is different from a model that assumes a dynamic contact angle, steady-state solutions are predicted correctly. Moreover, for low Weber numbers, the transient dynamics is expected to show close correspondence between the two contact-line models.

1.5 Dynamical Interaction

As was mentioned in section 1.1, coupling between the solid-body dynamics and the liquid dynamics is an important aspect of fluid-flow problems in space. The lack of references on numerical simulations of coupled solid-liquid dynamics in a micro-gravity environment demonstrates the complexity of these problems. However, some terrestrial applications of coupled solid-liquid dynamics have been simulated. Mostly, the falling of rigid bodies in air (or any other viscous fluid) is studied, *e.g.* the oscillatory behaviour of a falling piece of paper [37]. Other examples are described in [15], where the flow of liquid in anti-roll tanks aboard ships is simulated, and in [64], where the braking characteristics of vehicles, partially filled with liquid, is studied.

For simulation of coupled solid-liquid dynamics two models (one for the liquid dynamics and another for the solid-body dynamics) are discretised (see also [23, 24, 64]). Since both models have different characteristics, it is quite difficult to combine them and solve the discretised systems together. However, solving the systems sequentially is from a numerical point of view not a good idea. Indeed, a hierarchical method will not be stable for all ratios between the solid-body mass and the liquid mass. Similar stability problems are also encountered in other partitioned models, *e.g.* the viscous-inviscid interaction in boundary layer flow, where they are solved using a quasi-simultaneous method [12, 77].

1.6 Outline

This thesis contains four chapters, of which this introduction is the first. In chapter 2 the flow of a liquid in stationary containers is modelled. In chapter 3 this model is extended by coupling the liquid dynamics with the dynamics of the container. Both chapters 2 and 3 start with an explanation of the mathematical model, then discuss in detail the numerical model, and end with results. The thesis ends with a summary and conclusions in chapter 4. In figure 1.3 the structure of this thesis is summarised. Below, the outline of chapters 2 and 3 is discussed in more detail.

Chapter 2 starts with the mathematical model for liquid dynamics (section 2.1). This model consists of the Navier-Stokes equations, which govern the flow of a Newtonian, incompressible fluid. The Navier-Stokes equations are written in conservation form, which is the appropriate form for discretising these equations using the finite-volume method. Also, in this section, the boundary conditions are stated making the mathematical model complete. Boundary conditions are needed at the solid boundary and at the free surface. At the former, the no-slip boundary conditions for a viscous fluid are prescribed. At the free surface, boundary conditions for the velocity and the pressure are needed. Moreover,

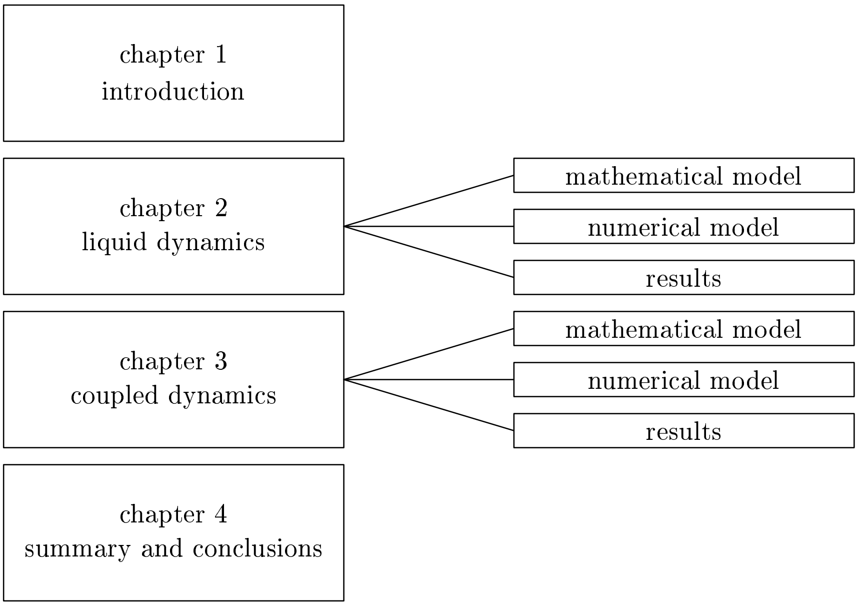


Figure 1.3: *Outline of thesis.*

at the contact line a boundary condition for computing the curvature of the free surface is required.

In this thesis the Navier-Stokes equations are discretised on a Cartesian grid because of the advantages described in section 1.2. To be able to handle complex geometries the cut-cell technique is applied. Cut cells are represented using so-called apertures [1, 7, 56]. Further, for distinguishing the different characteristics of computational cells, a labeling method is used. The use of apertures and labels is explained in section 2.2.

In section 2.3 the governing equations and boundary conditions are discretised in space and time. For the spatial discretisation the finite-volume method is used. This section is split into several subsections discussing subsequently the spatial discretisation of the continuity equation and the momentum equations, conservation of energy, the temporal discretisation, the method for solving the discretised equations, and the discretisation of the free-surface boundary conditions.

The treatment of the free surface is discussed in section 2.4. Two methods for advecting the free surface are presented, namely the method of Hirt and Nichols and the method of Youngs. For Youngs' method a piecewise-linear reconstruction method is used for determining the position of the free surface. The reconstructed free surface is then advected using the computed velocity field. The original method of Hirt and Nichols, described in [35], creates lots of "flotsam" and "jetsam" — small bits of fluid that get (unphysically) separated from the main body of fluid. Hence, a local height function is introduced for transporting fluid near the free surface (see also [65]). This local height function prevents massive creation of flotsam and jetsam. The original methods of Hirt and Nichols and of Youngs lose or gain mass considerably. In combination with a local height function these methods conserve mass rigorously.

Finally, results are presented in section 2.5, covering validation of all aspects of the numerical model. First, some simulations with prescribed velocity fields are performed for testing the advection methods of Hirt and Nichols and of Youngs. Also, the effect of using a local height function is studied. Then, both advection methods are tested in simulations with a velocity field that is computed from the Navier-Stokes equations. These simulations show that the method of Hirt and Nichols combined with a local height function gives the most accurate results, whence this method is exploited throughout the rest of the thesis. The models for surface tension and the contact line are validated in a series of simulations of wall adhesion in square containers. Also, wall adhesion in circular containers is simulated for validating the use of cut cells arising from complex geometries embedded in a Cartesian grid.

In chapter 3 the model of chapter 2 is extended by coupling the liquid dynamics with the dynamics of the container. First, the Navier-Stokes equations are adapted by incorporating a virtual body force, representing the motion of the solid body in which the liquid is contained (section 3.2).

In section 3.3 the mathematical and numerical model for the solid-body dynamics is explained. The motion of the solid body is governed by an equation for linear momentum and an equation for angular momentum. In the right-hand side of these equations, terms representing the force and torque due to the sloshing liquid appear. Discretisation of these equations would result in an unstable method if the liquid mass is too large compared to the mass of the solid body. This instability is exemplified with a simple mass-spring model. To overcome this stability problem the governing equations for the solid-body dynamics are rewritten, such that part of the liquid mass appears on the left-hand side of the equations. This part is treated simultaneously with the motion of the solid body. The remaining part of the liquid mass stays on the right-hand side and represents the sloshing motion with respect to a reference frame that moves with the solid body. For solving the discretised equations a fourth-order Runge-Kutta method is used.

Results of the numerical model for coupled solid-liquid dynamics are presented in section 3.4. First, the stability of the method is validated by simulating the free fall of a liquid-filled container. Then, for validating the solution method, the motion of an empty tank, for which an analytical solution is available, is simulated. For testing the interaction between the liquid dynamics and the solid-body dynamics, the rotational motion of a container, completely filled with liquid, is simulated. For this simulation a grid-refinement study and a time-step analysis is performed. Also, the effect of different liquid/solid mass ratios is studied. Finally, in this section, the flat-spin motion of containers partially filled with liquid is simulated. In a flat spin, initially, the container is rotating around the axis with minimum moment of inertia. From physics it is known that a free-flying body can rotate uniformly around one of the three principal moment-of-inertia axes only; rotation around the axis with intermediate moment of inertia is unstable, while rotation around the other two axes is stable. Thus, the initial condition (rotation around the axis with minimum moment of inertia) can be a steady state. However, if damping in the coupled system occurs (*e.g.* due to the viscous liquid), kinetic energy is lost. Hence, in this case, rotation around the axis with maximum moment of inertia (corresponding to a state of minimum kinetic energy) is the steady state in which the coupled system settles itself. As an example, the flat-spin motion of a small spacecraft (the Ejectable Ballistometer, better known as the Wet Satellite Model or WetSat), which flew in 1992, is simulated. Results from this simulation are compared to actual flight data.

Chapter 2

Liquid Dynamics

In this chapter fluid flow in stationary containers is studied. First, the mathematical model is explained in section 2.1. Then, in section 2.2, the representation of complex geometries on a Cartesian grid is explained. Section 2.3 discusses in detail the discretisation of the governing equations and boundary conditions. Methods for advecting a free surface are studied in section 2.4. Finally, in section 2.5, results are presented. For presentational reasons, the theory in this chapter is developed in two dimensions. In most situations this can be extended to three dimensions straightforwardly. If this is not the case, remarks on the three-dimensional extension are made.

2.1 Mathematical Model

In this section the mathematical model for the liquid dynamics is discussed: first the governing equations and thereafter the boundary conditions are stated.

2.1.1 Governing Equations

Throughout this thesis, the flow of a Newtonian, incompressible fluid with density ρ and molecular viscosity μ is studied. The motion of such a fluid is governed by an equation for conservation of mass

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2.1}$$

and an equation for conservation of momentum

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} = -\frac{1}{\rho} \left(\nabla p - (\nabla \cdot \mu \nabla)\, \boldsymbol{u} \right) + \boldsymbol{F}. \tag{2.2}$$

Equation (2.1) is referred to as the continuity equation, while equation (2.2) is better known as the Navier-Stokes equation (sometimes, the term Navier-Stokes equations refers to the combination of equations (2.1) and (2.2) — in this thesis the context makes clear whether by the term Navier-Stokes equations only the momentum equations or the combination of these with the continuity equation is meant). In these equations $\boldsymbol{u} = (u, v, w)^T$ is the velocity of the liquid and p denotes the liquid pressure. The velocity vector and pressure depend on three spatial variables and on time. The vector $\boldsymbol{F} = (F_x, F_y, F_z)^T$

represents acceleration due to an external force, *e.g.* gravity. Often, the left-hand side of equation (2.2) is abbreviated using the material derivative, *i.e.*

$$\frac{D\boldsymbol{u}}{Dt} = \frac{\partial \boldsymbol{u}}{\partial t} + \left(\boldsymbol{u} \cdot \nabla\right) \boldsymbol{u}.$$

The form in which the Navier-Stokes equations (2.1) and (2.2) are written, *i.e.* in differential form, is in particular suitable for a finite-difference method. In such a method the continuous derivatives are replaced by finite-difference formulas. Since in this thesis the Navier-Stokes equations are discretised using the (more physical) finite-volume method (see section 1.3), the Navier-Stokes equations are rewritten in conservation form. For an arbitrary control volume V (with boundary ∂V) in the liquid, conservation of mass changes to

$$\oint_{\partial V} \boldsymbol{u} \cdot \boldsymbol{n} \, dS = 0, \tag{2.3}$$

while conservation of momentum reads

$$\int_V \frac{\partial \boldsymbol{u}}{\partial t} \, dV + \oint_{\partial V} \boldsymbol{u}\boldsymbol{u}^T \cdot \boldsymbol{n} \, dS = -\frac{1}{\rho} \oint_{\partial V} \left(p\boldsymbol{I}_3 - \mu \nabla \boldsymbol{u}\right) \cdot \boldsymbol{n} \, dS + \int_V \boldsymbol{F} \, dV. \tag{2.4}$$

In these equations $\boldsymbol{n}$ is the outward-pointing normal on the boundary of the control volume V and $\boldsymbol{I}_3$ denotes the 3×3 identity matrix. Equations (2.3) and (2.4) are in the form that will be used for the discretisation; see section 2.3.

In the present study, in most cases, not the entire flow domain is filled with liquid; a free surface separates the liquid from the surrounding air. The location of this free surface is not known in advance and needs to be solved as part of the flow problem. If the free surface is described by an equation $S\left(x, y, z, t\right) = 0$, then the temporal evolution of the free surface satisfies

$$\frac{DS}{Dt} \equiv \frac{\partial S}{\partial t} + \left(\boldsymbol{u} \cdot \nabla\right) S = 0,$$

stating that it propagates with the liquid velocity.

Note that the Navier-Stokes equations are only solved in the liquid; the motion of the ambient air is neglected, which is justified by the large differences in molecular viscosity and density between air and liquid. However, in certain applications that are beyond the scope of this thesis, the dynamics of the surrounding air does play an important role, requiring a more extensive mathematical model than applicable here (see for example [57, 92]).

2.1.2 Boundary Conditions

Boundary conditions are needed at the solid boundary and at the free surface. Further, if surface tension can not be neglected (*e.g.* in a micro-gravity environment), a boundary condition is required at the contact line, *i.e.* the intersection of the free surface with the solid boundary.

Solid Boundary

At the solid boundary, in general, the no-slip boundary condition for a viscous fluid is applied, *i.e.*

$$\boldsymbol{u} = 0$$

is prescribed at the boundary of the flow domain, stating that the solid boundary is impermeable and that the fluid sticks to the wall due to viscous effects. However, in some situations a free-slip boundary condition

$$u_n = 0 \quad \text{and} \quad \frac{\partial u_t}{\partial \boldsymbol{n}} = 0$$

is prescribed, neglecting shear stress at the solid boundary. In the latter equation $u_n = \boldsymbol{u} \cdot \boldsymbol{n}$ and $u_t = \boldsymbol{u} \cdot \boldsymbol{t}$ denote the normal and tangential velocity at the solid boundary respectively. See also figure 2.1.

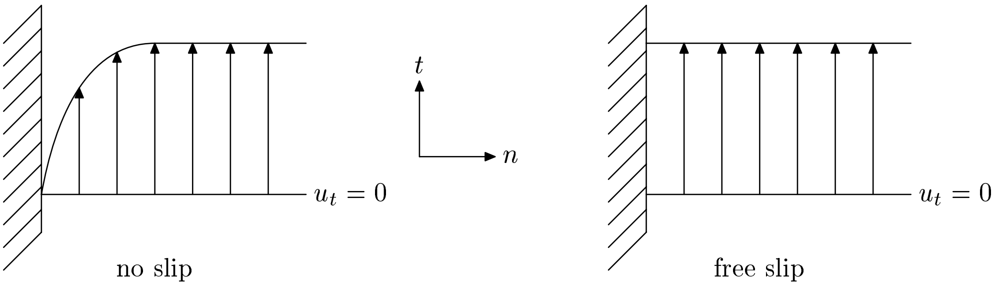


Figure 2.1: *Velocity profiles near the solid boundary for the no-slip and free-slip boundary condition.*

Free Surface

At the free surface, boundary conditions are needed both for the velocity and the pressure. Balancing the forces at the free surface results in a tangential free-surface condition

$$\mu \left(\frac{\partial u_n}{\partial \boldsymbol{t}} + \frac{\partial u_t}{\partial \boldsymbol{n}} \right) = 0, \tag{2.5}$$

which states that the air exerts no tangential stress on the fluid, and a normal condition

$$-p + 2\mu \frac{\partial u_n}{\partial \boldsymbol{n}} = -p_0 + \sigma\kappa. \tag{2.6}$$

In these equations u_n is the velocity normal to the free surface and u_t denotes the velocity in tangential direction. Further, p_0 is the ambient pressure of the air, σ is the surface tension of the fluid, and κ denotes the mean curvature of the free surface. If the free surface is described by $S(x, y, z, t) = 0$, then the mean curvature κ is given by

$$\kappa = \nabla \cdot \boldsymbol{n} = \nabla \cdot \left(\frac{\nabla S}{|\nabla S|} \right), \tag{2.7}$$

where $\boldsymbol{n}$ is the normal at the free surface.

For high values of the Reynolds number

$$Re = \frac{\rho U L}{\mu},$$

where U is a characteristic value of the velocity and L is a characteristic length scale, the second term on the left-hand side of equation (2.6) can be ignored since it is small compared to the other terms [36]. This simplification is made in this thesis also because of the small value of the viscosity that is used throughout this thesis. For high values of either the Bond number Bo or the Weber number We, which are defined by

$$Bo = \frac{\rho g L^2}{\sigma} \quad \text{and} \quad We = \frac{\rho L U^2}{\sigma},$$

respectively, also the second term on the right-hand side of (2.6) can be ignored. Indeed, for high Bo or We numbers, surface tension can be neglected since gravitational or inertial forces dominate. In this thesis, however, the emphasis will be on low values of the Bo and We number, whence the right-hand side of (2.6) can not be simplified.

Contact Angle

If surface tension can not be neglected (low values of the Bond number Bo and the Weber number We), *e.g.* for fluid flow in a micro-gravity environment, then the mean curvature of the free surface needs to be computed in order to apply boundary condition (2.6) at the free surface. For determining the mean curvature, spatial derivatives have to be computed (see equation (2.7)), whence boundary conditions are needed at the contact line where the free surface intersects the solid boundary. In this thesis these boundary conditions are supplied by a static contact angle θ, which is the angle between the normal at the free surface and the normal at the solid boundary. The contact angle is a quantity that depends on the material properties of liquid, air, and solid boundary.

For a schematic overview of the mathematical model see figure 2.2.

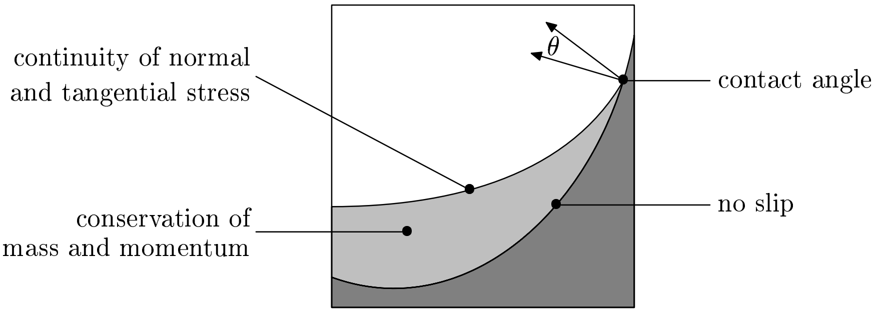


Figure 2.2: *Schematic overview of the mathematical model. Dark shading represents solid body and light shading denotes liquid.*

2.2 Geometry Discretisation

For a numerical method in computational fluid dynamics the choice of the grid is one of the first steps that has to be performed. Basically, a computational grid can be either structured or unstructured and is either boundary fitted or non-boundary fitted (see figure 2.3 and section 1.2). For complex geometries (in the figure the flow domain is a circle) the generation of an unstructured grid is relatively easy. However, the word *unstructured* already reveals the disadvantage of such a grid: the computational cells are not ordered and have different orientation and size. This makes bookkeeping a non-trivial problem, especially when a free surface needs to be tracked through the grid. In a structured grid the cells are ordered, but the generation of a structured, boundary-fitted grid is quite difficult and may be more time consuming than the flow simulation itself. A Cartesian grid, as shown in the right of figure 2.3, has two important advantages: the computing time for the generation of a Cartesian grid is negligible and already a lot of research has been done on free-surface tracking on orthogonal grids (see the references in chapter 1). However, if the flow domain is not staircase shaped, then its boundary is not aligned with grid lines; the boundary of the flow domain cuts the grid cells in various manners as can be seen in the right of figure 2.3. Hence, discretisation of the governing equations and the boundary conditions needs special care on a Cartesian grid. This is the approach that is chosen in this thesis.

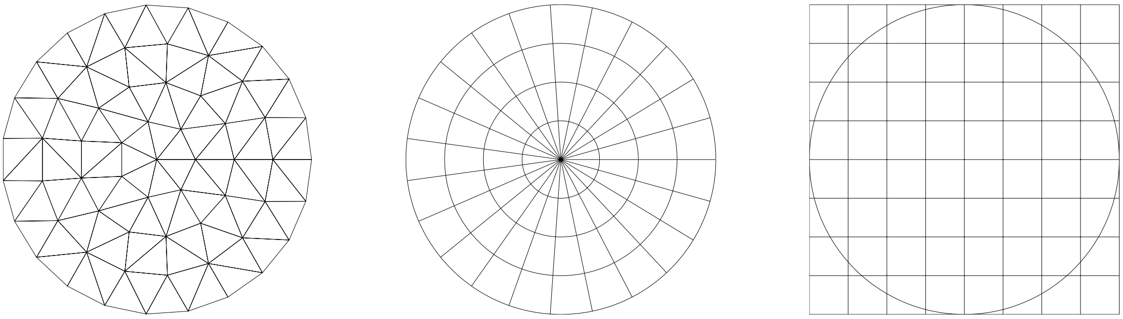


Figure 2.3: *Illustration of an unstructured, boundary-fitted grid (left), structured, boundary-fitted grid (middle), and structured, non-boundary-fitted grid (right) in the case that the flow domain is a circle. The latter is also called a Cartesian grid.*

2.2.1 Apertures

In order to recognise the flow domain on a Cartesian grid, a volume aperture F^b and edge apertures A^x, A^y, and A^z are introduced, indicating the fraction of a cell and cell face that is open for flow. See figure 2.4, where δx and δy denote the mesh size in x- and y-direction, for a graphical illustration of apertures. Note that in three dimensions the edge apertures contain information about the area of a cell face that is open for flow. Volume and edge apertures are used in the discretisation of the Navier-Stokes equations.

Apart from apertures for the geometry, one more volume aperture F^s is introduced for tracking the free surface. This aperture denotes the fractional volume of a cell that actually is occupied with fluid and is also known as the volume-of-fluid (VOF) function

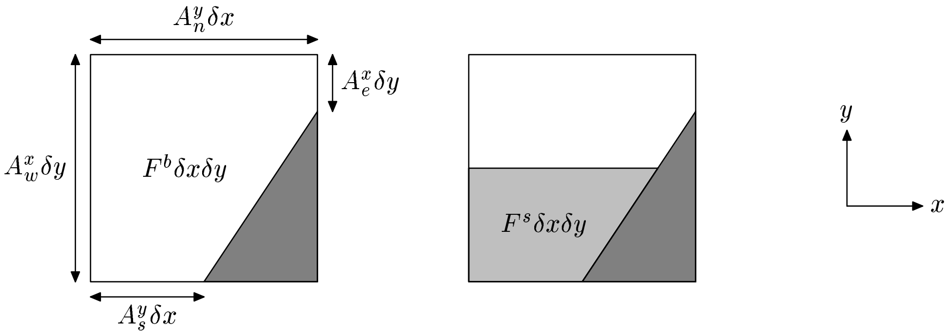
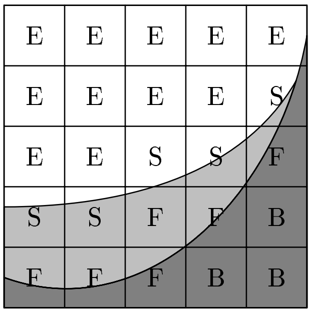


Figure 2.4: *Illustration of volume and edge apertures for the geometry (left) and volume aperture (or VOF function) for the free surface (right). The dark shading represents solid body, the lighter shading represents fluid.*

or colour function (see section 1.4). The volume apertures for the geometry and the free surface are related by $0 \le F^s \le F^b \le 1$.

2.2.2 Labeling

Based on the apertures, every computational cell is given a label. First, all the interior cells containing no fluid, *i.e.* cells with $F^b > 0$ and $F^s = 0$, are labeled as empty cells (abbreviated with E). Non-empty interior cells adjacent to empty cells are labeled as surface cells (S). These cells contain part of the free surface and satisfy $F^b > 0$ and $0 < F^s \le F^b$. All the remaining non-empty interior cells are called full cells (F). Note that these cells need not be full in the regular meaning of the word. The cells with $F^b = 0$ are flagged as boundary cells (B). In figure 2.5 an example of a label configuration is shown. A cell face is labeled based on the two computational cells it belongs to. Thus, for example, a cell face between an F cell and a B cell is called an FB cell face.



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

Figure 2.5: *Cell labels.*

2.3 Discretisation of Governing Equations

For the discretisation of the continuity equation and the Navier-Stokes equations, the finite-volume method is applied in such a way that the discrete difference operators for convection and diffusion preserve the symmetry properties of the corresponding continuous differential operators [81]. Thus, convection is discretised with a skew-symmetric difference operator and diffusion is approximated by a symmetric, negative-definite difference operator. Further, the coefficient matrix of the pressure gradient is equal to minus the transpose of the discrete divergence operator. On the one hand this fulfils the desire to let the discrete operators inherit spectral properties of the continuous operators, on the other hand this approach assures a stable integration of the Navier-Stokes equations on any grid and guarantees conservation of mass, momentum, and, in the absence of viscosity and external forces, kinetic energy as is shown later in this section.

2.3.1 Continuity Equation

Discretisation of the continuity equation (2.3) is rather straightforward. Consider a computational cell, as shown in figure 2.6, of which some part is occupied by the solid body (the shaded area). The velocities are staggered in this cell, meaning that a horizontal velocity (u) is placed at a vertical cell face and a vertical velocity (v) is placed at a horizontal cell face [29]. Note that such a velocity need not be positioned in the centre of that cell face; along the entire cell face (at least that part which is open for flow) this velocity holds. As a control volume V for discretisation of the continuity equation the part of the cell that is open for flow is chosen. Discrete conservation of mass implies that the sum of all the mass fluxes (velocity multiplied by area) through the boundary ∂V of V should vanish. For the cell in figure 2.6 this results in

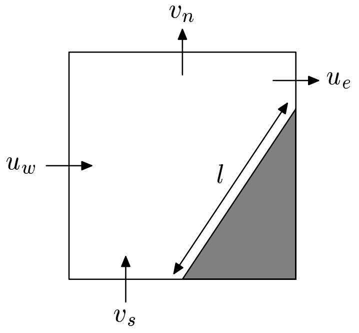


Figure 2.6: *Conservation cell for discretisation of the continuity equation.*

$$u_e A_e^x \delta y + v_n A_n^y \delta x - u_w A_w^x \delta y - v_s A_s^y \delta x + 0 \cdot l = 0, \tag{2.8}$$

where some notation of figure 2.4 has been used (see also [1]). The last term on the left-hand side of this equation represents the mass flux through the solid boundary, which, of course, equals zero. Since the mass flux through a solid boundary always vanishes in the continuity equation, equation (2.8) can be applied in every cell independent of the configuration of the solid boundary. If a cell and its faces do not contain part of the solid body, then all the edge apertures are equal to one, whence equation (2.8) is identical to a second-order central discretisation of equation (2.1).

2.3.2 Navier-Stokes Equations

In this section the spatial discretisation of the momentum equations (2.4) is discussed. Only the momentum equation in x-direction is considered; the other directions follow a similar approach. First the control volumes are defined, thereafter the spatial discretisation of the time derivative, the convective and diffusive terms, the pressure term, and acceleration due to external forces is discussed. Finally, an overview of the spatial discretisation is given and the relation between the discrete divergence and gradient operators is discussed. The temporal discretisation of the momentum equations is explained in section 2.3.4.

Control Volumes

For uncut cells, the control volume of a horizontal velocity $u_{i,j}$, located on a vertical cell face $x = x_i$ and $y_{j-1} \leq y \leq y_j$, is usually taken as illustrated in figure 2.7. The control volume, indicated with horizontal lines, consists of half of the left-hand cell $[x_{i-1}, x_i] \times [y_{j-1}, y_j]$ and half of the right-hand cell $[x_i, x_{i+1}] \times [y_{j-1}, y_j]$ (in later references these cells are denoted with indices (i, j) and $(i + 1, j)$ respectively). For defining control volumes in the case of cut cells, a closer look at figure 2.7 is required. The part of the control volume that is located in cell $(i + 1, j)$, for example, may be regarded as the union of an infinite number of horizontal line segments, running from $x = x_i$ to $x = x_{i+1}$, which are bisected; the half nearest to grid line $x = x_i$ is taken as part of the control volume.

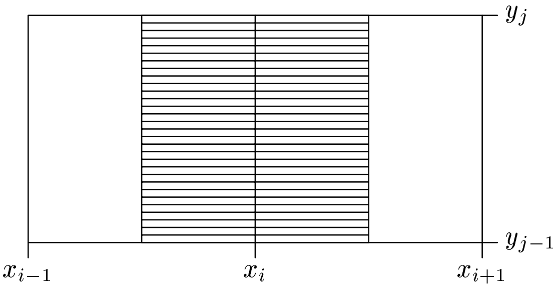


Figure 2.7: *Conservation cell (indicated with horizontal lines) for the momentum equation in the case of uncut cells.*

If the solid boundary intersects cell (i, j) or $(i + 1, j)$ (or both), then the control volume for the velocity $u_{i,j}$ is adapted, but basically the same procedure is used. Consider, again, cell $(i + 1, j)$. Horizontal line segments are drawn either from $x = x_i$ to $x = x_{i+1}$, from $x = x_i$ to the solid boundary, or from the solid boundary to $x = x_{i+1}$ (depending on the location of the boundary). These line segments are bisected and the half nearest to grid line $x = x_i$ is chosen as part of the control volume [82]. This procedure is illustrated by means of two examples in figure 2.8 (in this figure the dark shading denotes the part of the computational cells that is not open for flow). Note that also for uncut cells, because of construction, this procedure leads to a control volume for $u_{i,j}$ that bisects the "open" part of the cells (i, j) and $(i + 1, j)$. In three dimensions exactly the same procedure is used for constructing control volumes.

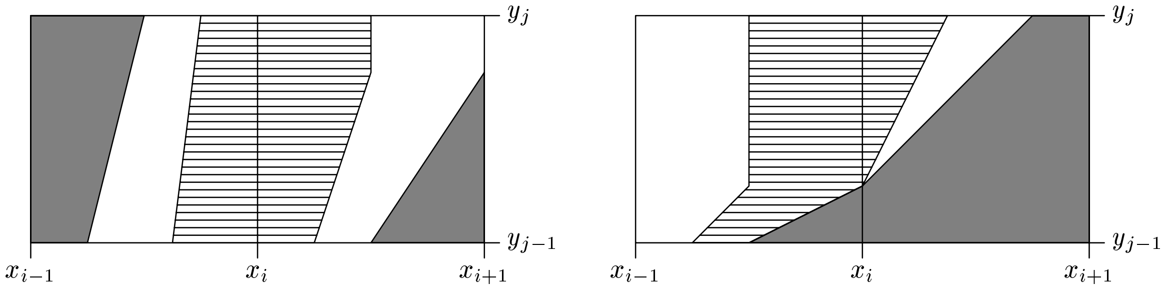


Figure 2.8: *Conservation cell (horizontal lines) for the momentum equation for two different configurations of the solid boundary.*

Time Derivative

The time derivative in the Navier-Stokes equations is discretised in space (the temporal discretisation is discussed in section 2.3.4) using the midpoint rule, *i.e.*

$$\int_V \frac{\partial u}{\partial t}\, dV \doteq \frac{\partial u_c}{\partial t} F_c^b \delta x_c \delta y, \tag{2.9}$$

where V is the control volume corresponding to the horizontal velocity u_c. This control volume has a volume of $F_c^b \delta x_c \delta y$, where $F_c^b = \left(F_w^b \delta x_w \delta y + F_e^b \delta x_e \delta y\right) / \left(\delta x_w \delta y + \delta x_e \delta y\right)$ and $\delta x_c = \frac{1}{2}\left(\delta x_w + \delta x_e\right)$ (F_w^b and F_e^b denote the volume apertures of the western and eastern computational cell contributing to the control volume of u_c; see also figure 2.9 for explanation of the notation).

Convective Terms

In the momentum equation in x-direction the convective terms (in conservative form) read

$$\oint_{\partial V} u\boldsymbol{u} \cdot \boldsymbol{n}\, dS. \tag{2.10}$$

Note the difference between the scalar u and the vector $\boldsymbol{u}$ in this expression. The scalar velocity is the horizontal momentum that is advected with a velocity equal to the vector velocity. The discretisation of (2.10) is best illustrated with an example. Consider hereto the situation in figure 2.9. The boundary ∂V of the control volume is indicated with a dashed line and consists of seven line segments, numbered from 1 to 7. On each of these segments the integrand in expression (2.10) is discretised as a product of the horizontal velocity (u) and the mass flux through this segment ($\boldsymbol{u} \cdot \boldsymbol{n}\, dS$), *i.e.*

$$\oint_{\partial V} u\boldsymbol{u} \cdot \boldsymbol{n}\, dS \doteq \sum_{k=1}^{7} u_k f_k,$$

where u_k and f_k denote the horizontal velocity at segment k and the mass flux through segment k respectively (the mass flux is positive if liquid leaves the control volume). The velocity u_1 at segment 1 is defined as the average of a zero velocity on the right and a

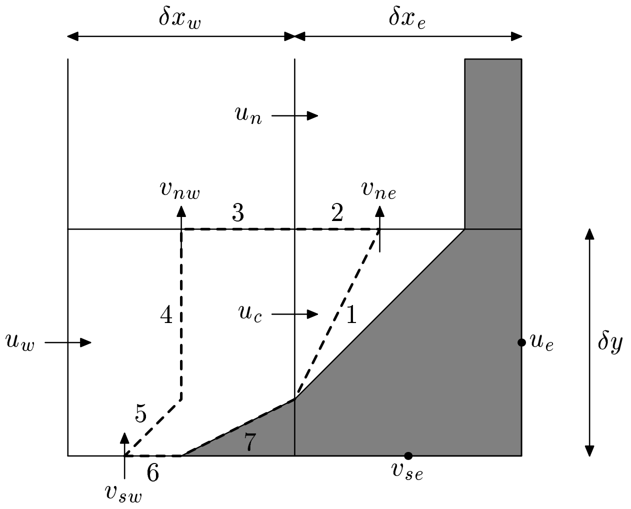


Figure 2.9: *Notation used in the discretisation of the convective and diffusive terms.*

velocity u_c on the left of segment 1. Note that, although the velocity u_c is drawn on the central cell face in figure 2.9, this velocity is characteristic throughout the entire control volume. Similarly, the zero velocity on the right of segment 1 holds for the total volume on the right of this segment and not only at the solid boundary. The mass flux through segment 1 is defined as the average of a zero mass flux through the solid boundary and the mass flux through the central cell face, *i.e.* $u_c A_c^x \delta y$ (see figure 2.4 for notation of the apertures), where A_c^x is the edge aperture of the cell face corresponding to u_c. This seems a complicated definition since it depends on the location of the solid boundary. However, the formula

$$u_1 f_1 = \frac{1}{2}(u_e + u_c)\,\frac{1}{2}\,(u_e A_e^x \delta y + u_c A_c^x \delta y)\,,$$

where u_e is a velocity defined at the eastern cell face (in this case, since the edge aperture A_e^x is zero, $u_e = 0$), can be applied at every eastern segment of a control volume; the edge apertures contain sufficient information to account for the presence of a solid body. The velocities u_2 and u_3 at segments 2 and 3 are defined as the average of the velocities u_n and u_c. Here the average is unweighted, even if the grid is stretched in y-direction, such that the resulting coefficient matrix for convection is skew symmetric. Indeed, if a weighting factor α would be used in the averaging of u_n and u_c, then the northern coefficient for u_c is equal to the sum of mass fluxes through segments 2 and 3 multiplied by α, while the southern coefficient for u_n is the same sum of mass fluxes with opposite sign multiplied by $1 - \alpha$, whence the matrix for convection is skew symmetric if and only if $\alpha = \frac{1}{2}$. The mass fluxes through segments 2 and 3 are computed as $v_{ne}\frac{1}{2}A_{ne}^y \delta x_e$ and $v_{nw}\frac{1}{2}A_{nw}^y \delta x_w$ respectively. Thus, summarised for segments 2 and 3

$$u_2 f_2 + u_3 f_3 = \frac{1}{2}(u_n + u_c)\left(v_{ne}\frac{1}{2}A_{ne}^y \delta x_e + v_{nw}\frac{1}{2}A_{nw}^y \delta x_w\right).$$

For segments 4 and 5 the same approach is used as for segment 1, which results in

$$u_4 f_4 + u_5 f_5 = -\frac{1}{2}(u_w + u_c)\,\frac{1}{2}\,(u_w A_w^x \delta y + u_c A_c^x \delta y)\,.$$

For both these segments the horizontal velocity is defined as the average of u_w on the left and u_c on the right. From this equation the separate mass fluxes through segments 4 and 5 are not clearly visible. However, since $A^x_w = 1$ in the example under consideration, these mass fluxes can be defined as $f_4 = -\frac{1}{2}\left(u_w A^x_c \delta y + u_c A^x_c \delta y\right)$ and $f_5 = -\frac{1}{2}\left(u_w \left(1 - A^x_c\right)\delta y + 0\right)$ respectively. Note that this formulation for segments 4 and 5 is similar to the one for segment 1: the eastern velocity and edge aperture have been replaced by their western equivalents and a minus sign has been added. Nevertheless, the situation with respect to the solid boundary is completely different in the eastern and western cell. For segment 6 a similar equation as for segments 2 and 3 is used. For segment 7, clearly, the mass flux is equal to zero. This makes the value for u_7 irrelevant. The convective contribution of segments 6 and 7 is written in a single formula for the southern part of the control volume, namely

$$u_6 f_6 + u_7 f_7 = -\frac{1}{2}\left(u_s + u_c\right)\left(v_{sw}\frac{1}{2}A^y_{sw}\delta x_w + v_{se}\frac{1}{2}A^y_{se}\delta x_e\right).$$

The terms of the discretisation along the seven segments are rearranged, such that the distinction between the unknown horizontal velocities and the coefficient matrix for convection (the mass fluxes) becomes more visible. This results in the following discrete version of expression (2.10)

$$\begin{aligned}\frac{1}{4}\left(u_e A^x_e \delta y + v_{ne} A^y_{ne}\delta x_e + v_{nw} A^y_{nw}\delta x_w - u_w A^x_w \delta y - v_{sw} A^y_{sw}\delta x_w - v_{se} A^y_{se}\delta x_e\right) u_c&\\ +\frac{1}{4}\left(u_e A^x_e \delta y + u_c A^x_c \delta y\right) u_e + \frac{1}{4}\left(v_{ne} A^y_{ne}\delta x_e + v_{nw} A^y_{nw}\delta x_w\right) u_n&\\ -\frac{1}{4}\left(u_w A^x_w \delta y + u_c A^x_c \delta y\right) u_w - \frac{1}{4}\left(v_{sw} A^y_{sw}\delta x_w + v_{se} A^y_{se}\delta x_e\right) u_s.&\end{aligned} \tag{2.11}$$

The coefficient of u_c, the diagonal entry of the coefficient matrix, vanishes since it represents the net mass flow through the boundaries of the eastern and western cell (this is equal to zero because of conservation of mass in F and S cells). Further, it is clear that the off-diagonal entries are skew symmetric. Thus, the discrete coefficient matrix for convection inherits this symmetry property of the continuous convective operator.

Diffusive Terms

Discretisation of the diffusive terms

$$\oint_{\partial V} \nabla u \cdot \boldsymbol{n}\, dS, \tag{2.12}$$

where the constant μ/ρ has been omitted, is more difficult than the convective terms since derivatives of the horizontal velocity are needed at the boundary of the control volume. Moreover, since the diffusive flux through the solid boundary need not be zero, a discretisation independent of the exact location of the solid boundary is not possible anymore. In figure 2.9, for example, the convective flux through segments 6 and 7 could be discretised together since the mass flux vanishes through a solid boundary. For the diffusive flux segments 6 and 7 have to be treated separately. In fact, segments of the boundary of the control volume which coincide with the solid boundary require a different approach than segments located in the interior of the flow domain as is demonstrated in this section.

First, the general idea of the discretisation of expression (2.12) is explained. Hereto, note that the integrand in this expression can be written as $\partial u/\partial \boldsymbol{n}$. This formulation motivates the following discretisation of (2.12) for the control volume in figure 2.9

$$\oint_{\partial V} \nabla u \cdot \boldsymbol{n}\, dS \doteq \sum_{k=1}^{7} \frac{u_k - u_c}{|\boldsymbol{n}_k|} \oint_k dS,$$

where u_k is a velocity that is characteristic on the side of segment k in the direction of the outward-pointing normal. Further, $|\boldsymbol{n}_k|$ is the distance between u_k and u_c. However, in the finite-volume method, a velocity does not have a fixed position but instead has a constant value throughout its control volume. Hence, the distance between two velocities has to be approximated. Since $|\boldsymbol{n}_k|$ is a geometric quantity having the dimension of length, it is approximated by

$$|\boldsymbol{n}_k| = \frac{V_k}{A_k} \quad \text{with} \quad A_k = \oint_k dS, \tag{2.13}$$

where V_k is a volume corresponding to segment k of the control volume and A_k is the area of segment k. The precise choice of V_k varies per segment and is explained below.

Now, for the seven segments in figure 2.9 the discretisation of the diffusive terms is discussed in more detail. Hereto, the volume of the control volume is decomposed in two volumes, namely V_e and V_w, representing the parts of the control volume located in the eastern and western cell respectively. Using volume apertures it follows that $V_e = \frac{1}{2}F_e^b \delta x_e \delta y$ and $V_w = \frac{1}{2}F_w^b \delta x_w \delta y$. For segment 1 the volume V_1 is set to the volume of the eastern cell, *i.e.* $V_1 = 2V_e$, and the velocity u_1 is equal to u_e (which is zero in this particular example since $A_e^x = 0$). Thus, for the eastern part of the control volume the contribution to the diffusive flux equals

$$\frac{u_1 - u_c}{|\boldsymbol{n}_1|} A_1 = \frac{u_e - u_c}{2V_e} A_e^2,$$

where $A_e = A_1$ is the area of the eastern part of the control volume. The velocities u_2 and u_3 are equal to u_n. The volume V_2 is computed as $\frac{1}{2}V_{ne} + \frac{1}{2}V_e$, where V_{ne} is the volume of the part of the control volume for u_n that is located in the north-eastern cell, *i.e.* $V_{ne} = \frac{1}{2}F_{ne}^b \delta x_e \delta y_n$. Similarly, if $V_{nw} = \frac{1}{2}F_{nw}^b \delta x_w \delta y_n$, then $V_3 = \frac{1}{2}V_{nw} + \frac{1}{2}V_w$. Hence, for the northern part

$$\frac{u_2 - u_c}{|\boldsymbol{n}_2|} A_2 + \frac{u_3 - u_c}{|\boldsymbol{n}_3|} A_3 = \frac{u_n - u_c}{\frac{1}{2}V_{ne} + \frac{1}{2}V_e} \left(\frac{1}{2} A_{ne}^y \delta x_e\right)^2 + \frac{u_n - u_c}{\frac{1}{2}V_{nw} + \frac{1}{2}V_w} \left(\frac{1}{2} A_{nw}^y \delta x_w\right)^2.$$

The terms between brackets represent the area of segments 2 and 3. Segments 4 and 5 can be treated simultaneously since these are located in the same computational cell, which is expressed by the following definition

$$\frac{u_4 - u_c}{|\boldsymbol{n}_4|} A_4 + \frac{u_5 - u_c}{|\boldsymbol{n}_5|} A_5 \equiv \frac{u_{4,5} - u_c}{|\boldsymbol{n}_{4,5}|} A_{4,5}.$$

The velocity $u_{4,5}$ is set to u_w and the volume $V_{4,5}$ is set to $2V_w$, whence the western contribution becomes

$$\frac{u_{4,5}-u_c}{|\boldsymbol{n}_{4,5}|}A_{4,5} = \frac{u_w-u_c}{2V_w}A_w^2,$$

where A_w is the area of the western part of the control volume. Note the similarity between the eastern and western expressions. For the southern part of the control volume the same approach is used as for the northern part, *i.e.*

$$\frac{u_6-u_c}{|\boldsymbol{n}_6|}A_6 = \frac{u_s-u_c}{\frac{1}{2}V_{sw}+\frac{1}{2}V_w}\left(\frac{1}{2}A_{sw}^y\delta x_w\right)^2 + \frac{u_s-u_c}{\frac{1}{2}V_{se}+\frac{1}{2}V_e}\left(\frac{1}{2}A_{se}^y\delta x_e\right)^2.$$

The last term in this equation is somewhat unexpected since the southern part consists of segment 6 only (segment 7 is treated separately as shown below). However, this term is added to acquire a formula that can be applied in every cell. In fact, in the example in figure 2.9, this term does not contribute to the diffusive terms since $A_{se}^y = 0$. The last segment ($k = 7$) is different from the previous six segments because it coincides with the solid boundary. Since a velocity is needed on both sides of the segment, the part of the control volume that is located in the western cell (with volume V_w) is reflected about this segment. In this mirrored volume a velocity $-u_c$ (because of the no-slip boundary condition) is assumed. See figure 2.10 for a graphical illustration. The volume V_7 is taken equal to V_w or, since the volume of the western control volume is equal to the mirrored volume,

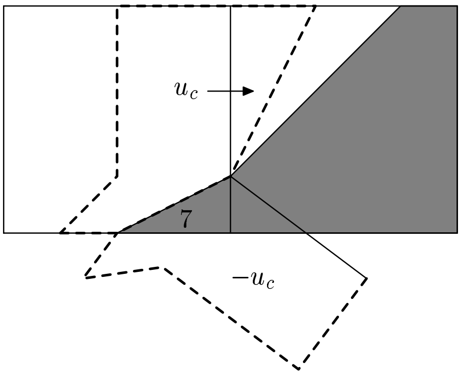


Figure 2.10: *Mirrored control volume for computing the diffusive flux through the solid boundary (segment 7).*

$$V_7 = \frac{1}{2}V_w + \frac{1}{2}V_w,$$

i.e. the sum of half the control volumes on both sides of the segment. This trivial decomposition is made to show consistency with the treatment of segments 2, 3, and 6. So, for the last segment an extra contribution to the diffusive terms arises, namely

$$\frac{u_7-u_c}{|\boldsymbol{n}_7|}A_7 = \frac{-u_c-u_c}{\frac{1}{2}V_w+\frac{1}{2}V_w}A_{\mathrm{bnd}}^2 = \frac{-2u_c}{V_w}A_{\mathrm{bnd}}^2,$$

where A_{bnd} is the area of the solid boundary that is located in the western cell.

After rearranging terms the discretised diffusive terms for the control volume in figure 2.9 read

$$\begin{aligned}-\frac{1}{2}\left(\frac{A_e^2}{V_e}+\frac{(A^y_{ne}\delta x_e)^2}{V_{ne}+V_e}+\frac{(A^y_{nw}\delta x_w)^2}{V_{nw}+V_w}+\frac{A_w^2}{V_w}+\frac{(A^y_{sw}\delta x_w)^2}{V_{sw}+V_w}+\frac{(A^y_{se}\delta x_e)^2}{V_{se}+V_e}\right)u_c-\frac{2A^2_{\text{bnd}}}{V_w}u_c\\ +\frac{1}{2}\frac{A_e^2}{V_e}u_e+\frac{1}{2}\left(\frac{(A^y_{ne}\delta x_e)^2}{V_{ne}+V_e}+\frac{(A^y_{nw}\delta x_w)^2}{V_{nw}+V_w}\right)u_n\\ +\frac{1}{2}\frac{A_w^2}{V_w}u_w+\frac{1}{2}\left(\frac{(A^y_{sw}\delta x_w)^2}{V_{sw}+V_w}+\frac{(A^y_{se}\delta x_e)^2}{V_{se}+V_e}\right)u_s.\end{aligned} \tag{2.14}$$

From this formulation it is clear that the coefficient matrix for diffusion is symmetric, corresponding to the symmetry of the continuous diffusive operator. Further, the diagonal entry is, apart from a minus sign, the sum of the off-diagonal entries added with possible terms due to the solid boundary. Moreover, the diagonal entries are negative and the off-diagonal entries are positive. Thus, the discrete diffusive operator is a negative-definite matrix; all its eigenvalues are real and less than zero.

Pressure Term

Basically, the discretisation of the pressure term, which (without the factor $-1/\rho$) in the momentum equation in x-direction reads

$$\oint_{\partial V} p n_x \, dS, \tag{2.15}$$

where n_x is the first component of the outward-pointing normal vector $\boldsymbol{n}$ on the boundary ∂V of the control volume, follows the same approach as has been used for the convective and diffusive terms; the integrand is evaluated at the various segments of ∂V. However, since the pressure is an unstaggered quantity and only the first component of the normal vector appears in the integrand, the exact procedure is slightly different. Using figure 2.11, the discretisation of the pressure term is written as

$$\oint_{\partial V} p n_x \, dS \doteq \sum_{k=1}^{7} p_k \oint_k n_x \, dS,$$

where, again, the summation is over the seven segments of the control volume. The pressure along segment k is denoted by p_k; thus $p_1 = p_e$ and $p_4 = p_5 = p_7 = p_w$. Since the first component n_x of the normal vector is zero along segments 2, 3, and 6, these segments do not contribute to the sum. Regarding the other segments, denote the angle between segment k and the vertical by α_k (with $0 \leq \alpha_k < \frac{\pi}{2}$). Then

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

where $s_k = +1$ for $k = 1, 7$ and $s_k = -1$ for $k = 4, 5$. The integral in the right-hand side of this equation is equal to the vertical length of segment k. Summarised, the pressure

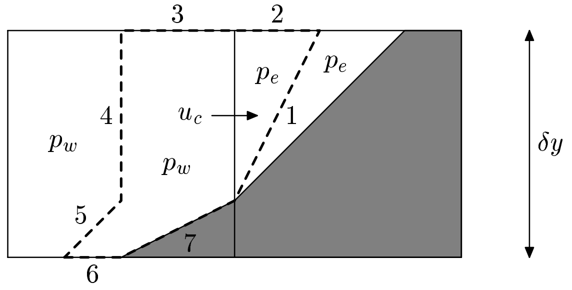


Figure 2.11: *Notation used in the discretisation of the pressure term.*

term is discretised as

$$\oint_{\partial V} p n_x \, dS \doteq p_e A_c^x \delta y - p_w A_c^x \delta y - p_w \left(1 - A_c^x\right) \delta y + p_w \left(1 - A_c^x\right) \delta y.$$

The contributions of segments 5 and 7 cancel each other since these segments lie in the same computational cell. This results in a formulation that, again, is independent of the exact location of the solid boundary, namely

$$\oint_{\partial V} p n_x \, dS \doteq \left(p_e - p_w\right) A_c^x \delta y. \tag{2.16}$$

External Force

An acceleration $\boldsymbol{F} = \left(F_x, F_y, F_z\right)^T$ due to external forces contributes to the momentum equation in x-direction as

$$\int_V F_x \, dV.$$

This term is discretised similar to the discretisation of the time derivative, *i.e.*

$$\int_V F_x \, dV \doteq F_{x_c} F_c^b \delta x_c \delta y_c, \tag{2.17}$$

where $F_c^b \delta x_c \delta y_c$ is the volume of the control volume V of the momentum equation under consideration and F_{x_c} is the value of F_x in this control volume (the acceleration may vary in space). If gravity is the only external force present, *i.e.* $\boldsymbol{F} = \boldsymbol{g} = \left(g_x, g_y, g_z\right)^T$ (constant in space), then this contribution to the momentum equations can be written as a boundary integral

$$\int_V F_x \, dV = \int_V \nabla \cdot \begin{pmatrix} g_x x \\ 0 \\ 0 \end{pmatrix} dV = \oint_{\partial V} \begin{pmatrix} g_x x \\ 0 \\ 0 \end{pmatrix} \cdot \boldsymbol{n} \, dS = \oint_{\partial V} g_x x n_x \, dS.$$

This boundary integral is similar to the pressure term (2.15) except for the factor x in the integrand. For segments 1, 4, 5, and 7 of the control volume in figure 2.11, the value of

x must be defined (the other segments do not contribute to the acceleration since $n_x = 0$ along these segments). In general, this value changes along a segment of the control volume. However, to be consistent with the pressure term, the value of x is kept constant along each segment and along segments sharing the same computational cell. With this in mind, the discretisation of the acceleration due to gravity becomes

$$\oint_{\partial V} g_x x n_x \, dS \doteq g_x x_e A_c^x \delta y - g_x x_w A_c^x \delta y = g_x \left(x_e - x_w\right) A_c^x \delta y, \tag{2.18}$$

where x_e and x_w are the (constant) x-coordinates along segment 1 and segments 4 and 5 respectively. In the case of uncut cells, the distance between x_e and x_w is equal to $\delta x_c = \frac{1}{2}\left(\delta x_e + \delta x_w\right)$. To be consistent with this situation, also for cut cells this choice is made.

Overview of the Spatial Discretisation

In this section the spatial discretisation of the continuity equation and the Navier-Stokes equation has been discussed. Before these equations are discretised in time in section 2.3.4, some notation is introduced and the evolution of kinetic energy is discussed.

Denote the vector of all (discrete) velocities that are solved from the momentum equations by $\boldsymbol{u}_h$ (thus, in two dimensions, this vector consists of velocity components u and v). The way in which the velocities are ordered is not important. Then, the (spatially) discretised continuity equations (in every cell, where the pressure has to be solved, one continuity equation is discretised) are written as

$$\mathcal{M}\boldsymbol{u}_h = 0. \tag{2.19}$$

Every row of the coefficient matrix $\mathcal{M}$ corresponds to the discrete continuity equation in a certain computational cell (the ordering of the computational cells is irrelevant). For example, the number of nonzero elements in the row corresponding to the eastern and western cell in figure 2.11 is equal to 2 and 4 respectively. Note that every entry in the matrix $\mathcal{M}$ contains geometrical information only, namely the area of a cell face that is open for flow (see also equation (2.8)). Similarly, the (spatially) discretised momentum equations are written as

$$\mathcal{V}\frac{d\boldsymbol{u}_h}{dt} + \mathcal{C}\left(\boldsymbol{u}_h\right)\boldsymbol{u}_h = -\frac{1}{\rho}\left(\mathcal{P}\boldsymbol{p}_h - \mu\mathcal{D}\boldsymbol{u}_h\right) + \boldsymbol{F}_h,$$

where $\boldsymbol{p}_h$ is a vector containing all the unknown pressures. The ordering of this vector must correspond to the ordering of the continuity equations, *i.e.* the j-th component of $\boldsymbol{p}_h$ contains the pressure in the cell of which the discrete continuity equation is found in the j-th row of matrix $\mathcal{M}$. The coefficient matrix $\mathcal{V}$ in front of the time derivative is a diagonal matrix. The entries in this matrix represent the volume of the control volumes (see equation (2.9)). Further, $\mathcal{C}$ and $\mathcal{D}$ are the coefficient matrices of the discretised convective and diffusive operators. The entries of matrix $\mathcal{C}$ are mass fluxes through cell faces (see expression (2.11)). Thus, apart from geometrical information, this matrix contains also velocities. The matrix $\mathcal{D}$ contains only geometrical information (see expression (2.14)). The vector $\boldsymbol{F}_h$ contains the discretisation of the acceleration due to external forces (see equation (2.17) or (2.18)). The coefficient matrix $\mathcal{P}$ in front of the pressure contains the same geometrical information as $\mathcal{M}$ (see equation (2.16)).

Analytically, the divergence operator $(\nabla\cdot)$ and gradient operator (∇) are related by $\nabla = -\left(\nabla\cdot\right)^T$. For the discretisation method used in this thesis, this property is preserved, *i.e.* $\mathcal{P} = -\mathcal{M}^T$. This is explained using figure 2.12. Assume that the vector $\boldsymbol{p}_h$ is ordered such that the pressures p_m and p_n appear as the m-th and n-th entry respectively. In this case, the discrete continuity equations in cells m and n have entries in the matrix $\mathcal{M}$ that operate on the velocity u_c. If only these entries (located in rows m and n) in the matrix $\mathcal{M}$ are shown, then the discrete continuity equation (see also equation (2.8)) has the following form

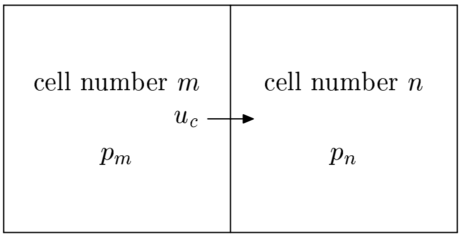


Figure 2.12: *Notation used in demonstrating the relation between the discrete divergence and gradient operator.*

$$\mathcal{M}\boldsymbol{u}_h = \begin{pmatrix} & \vdots & \\ \cdots & A^x_c\delta y & \cdots \\ & \vdots & \\ \cdots & -A^x_c\delta y & \cdots \\ & \vdots & \end{pmatrix} \begin{pmatrix} \vdots \\ u_c \\ \vdots \end{pmatrix} = 0.$$

In the discrete momentum equation for u_c the pressures in cell m and n appear. This means that in the row corresponding to the momentum equation for u_c, columns m and n of the matrix $\mathcal{P}$ contain nonzero elements (see also equation (2.16)), *i.e.*

$$\mathcal{P}\boldsymbol{p}_h = \begin{pmatrix} & \vdots & & \vdots & \\ \cdots & -A^x_c\delta y & \cdots & A^x_c\delta y & \cdots \\ & \vdots & & \vdots & \end{pmatrix} \begin{pmatrix} \vdots \\ p_m \\ \vdots \\ p_n \\ \vdots \end{pmatrix}.$$

From this it follows that indeed $\mathcal{P} = -\mathcal{M}^T$, whence the discretised Navier-Stokes equations can be written as

$$\mathcal{V}\frac{d\boldsymbol{u}_h}{dt} + \mathcal{C}\left(\boldsymbol{u}_h\right)\boldsymbol{u}_h = -\frac{1}{\rho}\left(-\mathcal{M}^T\boldsymbol{p}_h - \mu\mathcal{D}\boldsymbol{u}_h\right) + \boldsymbol{F}_h. \qquad (2.20)$$

An attractive feature of the spatial discretisation that has been presented in this section is that it reduces to a standard second-order finite-difference discretisation when it is applied to a staircase-shaped geometry on a uniform grid.

2.3.3 Evolution of Kinetic Energy

With respect to the stability of the spatial discretisation it is interesting to see how discrete energy evolves in time. Hereto, first the kinetic energy is discretised in space as

$$\frac{1}{2}\int \rho|\boldsymbol{u}|^2\,dV \doteq \frac{1}{2}\left\langle \boldsymbol{u}_h, \rho\mathcal{V}\boldsymbol{u}_h\right\rangle \equiv E_h,$$

where the integral is over the entire flow domain and $\langle\cdot,\cdot\rangle$ denotes the standard inner product for a finite-dimensional vector space. Hence, the evolution of discrete energy is given by

$$\frac{dE_h}{dt} = \frac{1}{2}\left\langle \frac{d\boldsymbol{u}_h}{dt}, \rho\mathcal{V}\boldsymbol{u}_h\right\rangle + \frac{1}{2}\left\langle \boldsymbol{u}_h, \rho\mathcal{V}\frac{d\boldsymbol{u}_h}{dt}\right\rangle = \left\langle \boldsymbol{u}_h, \rho\mathcal{V}\frac{d\boldsymbol{u}_h}{dt}\right\rangle,$$

where the last equality is justified since $\boldsymbol{u}_h$ is a real vector and $\mathcal{V}$ is a real, diagonal matrix. Using the discretised momentum equations (2.20), this is equal to

$$\begin{aligned}\frac{dE_h}{dt} &= \left\langle \boldsymbol{u}_h, -\rho\mathcal{C}\boldsymbol{u}_h + \mathcal{M}^T\boldsymbol{p}_h + \mu\mathcal{D}\boldsymbol{u}_h + \rho\boldsymbol{F}_h\right\rangle \\ &= -\rho\left\langle \boldsymbol{u}_h, \mathcal{C}\boldsymbol{u}_h\right\rangle + \left\langle \boldsymbol{u}_h, \mathcal{M}^T\boldsymbol{p}_h\right\rangle + \mu\left\langle \boldsymbol{u}_h, \mathcal{D}\boldsymbol{u}_h\right\rangle + \rho\left\langle \boldsymbol{u}_h, \boldsymbol{F}_h\right\rangle \\ &= -\rho\left\langle \boldsymbol{u}_h, \mathcal{C}\boldsymbol{u}_h\right\rangle + \left\langle \mathcal{M}\boldsymbol{u}_h, \boldsymbol{p}_h\right\rangle + \mu\left\langle \boldsymbol{u}_h, \mathcal{D}\boldsymbol{u}_h\right\rangle + \rho\left\langle \boldsymbol{u}_h, \boldsymbol{F}_h\right\rangle.\end{aligned}$$

Since $\boldsymbol{u}_h$ is real and the convective terms have been discretised such that the discrete convective operator $\mathcal{C}$ is a skew-symmetric matrix, the first term on the right-hand side of this equation vanishes. Also, the second term is equal to zero because discrete mass is conserved. Hence, in the absence of an external force, the evolution of discrete kinetic energy is given by

$$\frac{dE_h}{dt} = \mu\left\langle \boldsymbol{u}_h, \mathcal{D}\boldsymbol{u}_h\right\rangle.$$

Since the discrete diffusive operator $\mathcal{D}$ is a negative-definite matrix, it follows that $dE_h/dt \leq 0$; energy is dissipated because of viscosity. Thus, in the absence of an external force, the spatial discretisation is guaranteed stable.

Note that stability could be proven since the symmetry properties of the discrete difference operators are inherited from the continuous differential operators. If, for example, only the discrete divergence operator contains apertures and the gradient operator is discretised with standard finite differences [1, 73], then the equality $\mathcal{P} = -\mathcal{M}^T$ is not longer valid and precautions might be necessary for a stable spatial discretisation.

2.3.4 Temporal Discretisation

In this section the continuity equation (2.19) and the momentum equations (2.20) are discretised in time. For the time discretisation the explicit forward Euler method is exploited. Using a superscript indicating the time level, this results in

$$\mathcal{M}\boldsymbol{u}_h^{n+1} = 0, \tag{2.21}$$

$$\mathcal{V}\frac{\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n}{\delta t} + \mathcal{C}\left(\boldsymbol{u}_h^n\right)\boldsymbol{u}_h^n = -\frac{1}{\rho}\left(-\mathcal{M}^T\boldsymbol{p}_h^{n+1} - \mu\mathcal{D}\boldsymbol{u}_h^n\right) + \boldsymbol{F}_h^n.$$

Rearranging terms in the momentum equations gives

$$\boldsymbol{u}_h^{n+1} = \tilde{\boldsymbol{u}}_h^n + \delta t \mathcal{V}^{-1} \frac{1}{\rho} \mathcal{M}^T \boldsymbol{p}_h^{n+1}, \tag{2.22a}$$

where

$$\tilde{\boldsymbol{u}}_h^n = \boldsymbol{u}_h^n - \delta t \mathcal{V}^{-1} \left(\mathcal{C}\left(\boldsymbol{u}_h^n\right) \boldsymbol{u}_h^n - \frac{\mu}{\rho} \mathcal{D} \boldsymbol{u}_h^n - \boldsymbol{F}_h^n \right). \tag{2.22b}$$

The continuity equation is discretised at the new time level $n+1$ to ensure a divergence-free velocity field at the latter time level. Except for the pressure, all the terms in the momentum equations are discretised at the old time level n. Note that if the pressure gradient is discretised at the old time level as well, then both the continuity equation and the momentum equations predict a velocity field at the new time level which need not match. By discretising the pressure at the new time level, a correction to the intermediate vector field $\tilde{\boldsymbol{u}}_h^n$ is applied, such that the new velocity field is divergence free. The method to solve equations (2.21) and (2.22) is discussed in the next section.

2.3.5 Solution Method

First, the solution method for the discretised continuity equation and momentum equations is explained in the case that the entire flow domain is occupied by fluid. Then, some modifications are presented to account for the presence of a free surface. Hereto, some notation is introduced for sets of cells and cell faces with the same labels (for an explanation of the labeling see section 2.2.2). The set of all F cells is denoted by Ω_F, the set of all B cells by Ω_B, and so forth. Similarly, the set of all FF cell faces is denoted by Ω_{FF}, the set of all FB cell faces by Ω_{FB}, and so forth.

Without Free Surface

Assume that the flow domain is completely filled with liquid. In this case, every computational cell belongs to $\Omega_F \cup \Omega_B$ and every cell face to $\Omega_{FF} \cup \Omega_{FB} \cup \Omega_{BB}$. Further, assume that an initial velocity field $\boldsymbol{u}_h^n$ is given on $\Omega_{FF} \cup \Omega_{FB}$ (velocities at BB cell faces are not needed in the model — although, in the finite-volume method, a velocity does not have a fixed position at a cell face, but rather has a constant value throughout the control volume corresponding to that cell face, it is more convenient to speak of velocities located at cell faces). Since every F cell contains fluid and fluid is contained in F cells only, the discrete continuity equation (2.21) is solved on Ω_F. Every cell face having an edge aperture not equal to zero is an FF cell face. Hence, the discrete momentum equations (2.22) are solved on Ω_{FF}. This is done by first constructing a temporary vector field $\tilde{\boldsymbol{u}}_h^n$ on Ω_{FF} using equation (2.22b). Next, equation (2.22a) is substituted in (2.21), which gives a Poisson equation for the pressure on Ω_F

$$\mathcal{M}\mathcal{M}^T \boldsymbol{p}_h^{n+1} = -\frac{1}{\delta t} \rho \mathcal{V} \mathcal{M} \tilde{\boldsymbol{u}}_h^n. \tag{2.23}$$

This equation is solved iteratively with an SOR method (Gauss-Seidel with overrelaxation), where the relaxation parameter is automatically adjusted for optimal convergence behaviour [4]. A checkerboard ordering of the pressure is used to facilitate parallelisation on a parallel computer platform. For the problems that are considered in this thesis, the SOR method performs adequately; a more modern solver does not increase the performance significantly [80]. Once the pressure at the new time level is known, the vector field $\tilde{\boldsymbol{u}}_h^n$ is corrected with the pressure gradient (equation (2.22a)) to obtain the velocity field on Ω_{FF} at the new time level. Note that velocities on Ω_{FB} do appear in the discretised continuity equation and in the discretised convective and diffusive terms (thus are needed in the construction of $\tilde{\boldsymbol{u}}_h^n$). In the continuity equation and convective terms, these velocities are always multiplied with a zero edge aperture, whence the actual value is not relevant. However, for the diffusive terms, velocities on Ω_{FB} have to be zero. Thus, these velocities are initialised to zero once and keep this value throughout the simulation.

With Free Surface

If only part of the flow domain is filled with liquid, then a computational cell and cell face belong to $\Omega_F \cup \Omega_B \cup \Omega_S \cup \Omega_E$ and $\Omega_{FF} \cup \Omega_{FB} \cup \Omega_{BB} \cup \Omega_{FS} \cup \Omega_{SB} \cup \Omega_{SS} \cup \Omega_{EB} \cup \Omega_{SE} \cup \Omega_{EE}$ respectively. In this case the presence of fluid is not restricted to F cells; also S cells contain fluid. On Ω_F, like above, the continuity equation is discretised and a Poisson equation for the pressure is formed. On Ω_S, boundary condition (2.6) for the pressure at the free surface is discretised (see section 2.3.6). These discretised equations (one in every S cell) are added to the system (2.23), giving an equation for the pressure on $\Omega_F \cup \Omega_S$. Because the pressure is solved in F and S cells, the momentum equations can be discretised on $\Omega_{FF} \cup \Omega_{FS} \cup \Omega_{SS}$. Velocities on $\Omega_{FB} \cup \Omega_{SB} \cup \Omega_{EB}$ are initialised to zero once since these are needed (or will be needed in a later time step) in the computation of the diffusive terms. Velocities on Ω_{BB} are not needed in the model. Near the free surface, velocities on $\Omega_{SE} \cup \Omega_{EE}$ may be needed in the discretised continuity equation and momentum equations. The computation of these velocities is discussed in section 2.3.6.

2.3.6 Free-Surface Boundary Conditions

In this section the discretisation of the boundary conditions for computing velocities on $\Omega_{SE} \cup \Omega_{EE}$ and the pressure in S cells is discussed.

Velocities on Ω_{SE}

When velocities at cell faces between an S and an E cell (SE velocities for short) appear in the discretised momentum equations, these velocities need a value. Also, some velocities on Ω_{SE} are required in the computation of velocities on Ω_{EE} (EE velocities for short) as is demonstrated below. By continuing the velocity field to S cells, all the velocities on Ω_{SE} are given a value. In other words, these velocities are computed by demanding conservation of mass (equation (2.8)) in S cells. However, contrary to F cells, the pressure Poisson equation is not formed in S cells. Note that if mass is not conserved in S cells, then diagonal entries of the coefficient matrix for convection are not equal to zero in rows corresponding to FS velocities, whence conservation of kinetic energy can not be guaranteed (see section 2.3.3).

If only one of the cell faces of an S cell belongs to Ω_{SE}, as is shown in the left of figure 2.13, then the other cell faces belong to $\Omega_{FS} \cup \Omega_{SS} \cup \Omega_{SB}$. All the velocities corresponding to these cell faces are either solved from the momentum equations or the

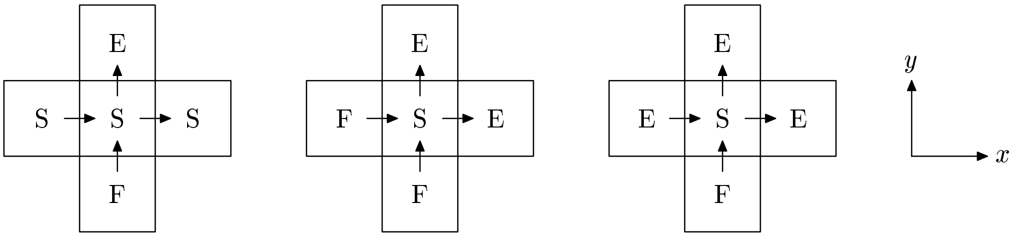


Figure 2.13: *Three possible label configurations around an S cell.*

no-slip boundary condition. Hence, the SE velocity can be solved from equation (2.8). If two or more cell faces of an S cell belong to Ω_{SE}, then conservation of mass is demanded in each Cartesian direction. Thus, for example, in the middle configuration in figure 2.13, the mass flux through each SE cell face is set equal to the mass flux through the opposite FS cell face. This is not possible if two SE cell faces face each other, such as in the right of figure 2.13. In this case, the two SE velocities are set to zero.

In three dimensions an extra complication arises. Consider therefore the middle and right-hand configurations in figure 2.13 with F cells in front of and behind (thus in z-direction) the central S cell. Now, it is not correct to prescribe conservation of mass in the x- and y-direction to solve the SE velocities. Indeed, the two FS velocities in z-direction, which are solved from the momentum equations, need not satisfy $\partial w/\partial z = 0$. Therefore, in these cases an amount $\partial w/\partial z$ is redistributed over the SE velocities such that mass is conserved in all S cells.

Velocities on Ω_{EE}

Velocities on Ω_{EE} are needed in the discrete momentum equations for neighbouring SS velocities (see figure 2.14). Since EE velocities are approximately tangential to the free surface, the tangential boundary condition (2.5) for the velocity is discretised for the computation of these velocities. In this boundary condition derivatives of the velocity normal and tangential to the free surface in the direction tangential and normal to the free surface appear. For an arbitrary orientation of the free surface these derivatives are difficult to compute on a Cartesian grid. Hence, equation (2.5) is split in Cartesian directions. Thus, in two dimensions, every EE velocity that is needed in the model is

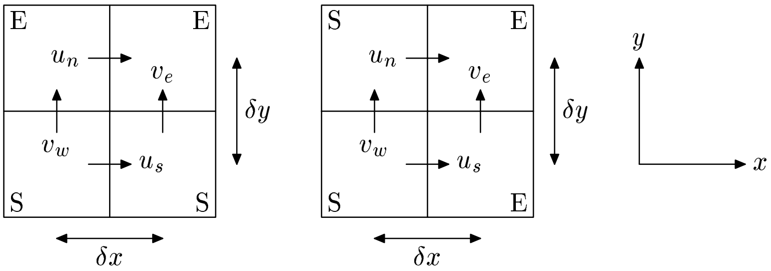


Figure 2.14: *Two possible label configurations in two dimensions near an EE velocity.*

computed by discretising

$$\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)=0. \tag{2.24}$$

Equation (2.24) is discretised using standard finite differences. For example, for the left-hand configuration in figure 2.14, the EE velocity u_n is solved from

$$\mu\left(\frac{u_n-u_s}{\delta y}+\frac{v_e-v_w}{\delta x}\right)=0.$$

In this equation SE velocities v_e and v_w are needed. Hence, it is important that SE velocities are computed before EE velocities. The SS velocity u_s is solved from the discrete momentum equations. In extraordinary occasions it is possible that, in two dimensions, an EE velocity is surrounded by two SS velocities. In this case, equation (2.24) is discretised twice and the resulting values are averaged (a weighted average is taken on a stretched grid) to produce a final value for the EE velocity.

In three dimensions the computation of EE velocities is more complicated since the number of SS velocities around an EE velocity may vary from zero up to four. Further, the tangential boundary condition (2.5) separates in three (Cartesian) equations, namely

$$\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)=0,\quad \mu\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)=0,\quad \text{and}\quad \mu\left(\frac{\partial w}{\partial x}+\frac{\partial u}{\partial z}\right)=0. \tag{2.25}$$

Of course, if an EE velocity has no neighbouring SS velocities, then this EE velocity is not needed in the model. If it has one neighbouring SS velocity, then one of the equations in (2.25), dependent on the direction of the EE velocity and the SE velocities, is discretised. For example, for configurations (a) and (b) in figure 2.15, the third respectively first equation in (2.25) is discretised for computing the EE velocity. If an EE velocity has two neighbouring SS velocities, then either the two SS velocities are located in different Cartesian directions (configuration (c) in figure 2.15) or the two SS velocities face each other (configuration (d)). In both cases, for each of the SS velocities a value for the EE

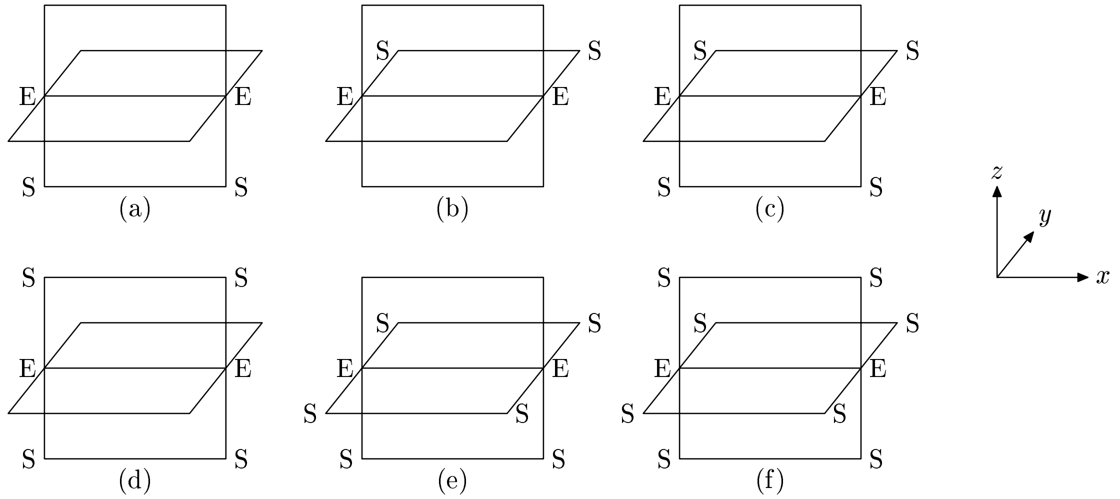


Figure 2.15: *Six possible label configurations near an EE velocity in three dimensions.*

velocity is computed by discretising one of the equations in (2.25). Then, based on these two values, a final value for the EE velocity is computed using weighted averaging. In the case of three neighbouring SS velocities the same procedure as with one SS velocity is applied; the two SS velocities facing each other are, because of symmetry reasons, not used in the computation of the EE velocity. Indeed, in configuration (e) in figure 2.15, the free surface is approximately parallel to the xy-plane, whence only the third equation in (2.25) is discretised. Finally, if an EE velocity is surrounded by four SS velocities (configuration (f)), then for each SS velocity an equation in (2.25) is discretised. The four values for the EE velocity are averaged to produce a final value for the EE velocity.

Pressure in S Cells

In S cells a value for the pressure is needed, which is provided by boundary condition (2.6) for the pressure at the free surface. Since the second term ($2\ \mu \partial u_n / \partial \boldsymbol{n}$) on the left-hand side in this equation is usually small compared to the other terms (especially in a micro-gravity environment, where capillary forces dominate, but also because of the small value of the viscosity that is used in this thesis [36]), this term is neglected in the numerical model, which leaves

$$p = p_0 - \sigma\kappa. \tag{2.26}$$

Here, p is the pressure at the free surface, p_0 is the pressure in the ambient air, σ is the surface tension of the fluid, and κ denotes the mean curvature of the free surface. Since this equation is discretised in S cells, the mean curvature of the free surface needs to be computed in every S cell (this procedure is explained below). Once the curvature is known, the pressure at the free surface is easily found using equation (2.26). In the finite-volume method it is assumed that the pressure has a constant value throughout each computational cell. Hence, it is possible to define the pressure in an S cell to be the pressure at the free surface in this cell. Thus, equation (2.26) is added unchanged to the pressure Poisson equation. Another option is to define the pressure in an S cell based on interpolation of the pressure at the free surface and the pressure in a neighbouring F cell [35, 49]. This is especially beneficiary for simulation of liquid sloshing in a micro-gravity environment, where the shape of the free surface drives the flow. This method is illustrated in figure 2.16. Based on the orientation of the free surface (in two dimensions horizontal or vertical; see the next paragraph for more details), an F cell is determined, which is used for computing the pressure in the centre of the S cell. If the distance between

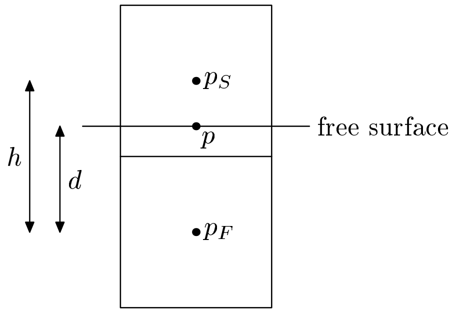


Figure 2.16: *Interpolation of the free-surface pressure to the centre of an S cell.*

the centre of the F cell and the free surface is equal to d and the distance between the centres of the F and S cell is h, then the pressure p_S in the centre of the S cell is given by

$$p_S = \eta p + (1 - \eta)\, p_F = \eta\,(p_0 - \sigma\kappa) + (1 - \eta)\, p_F,$$

where p_F is the pressure in the (centre of the) F cell and $\eta = h/d$.

Curvature of the Free Surface

For applying the boundary condition for the pressure at the free surface, in every S cell the mean curvature κ of the free surface has to be computed. If the free surface is described by a level-set function $S\,(x, y, t) = 0$, then the mean curvature of the free surface is given by $\kappa = \nabla \cdot \boldsymbol{n}$, where $\boldsymbol{n} = \nabla S / |\nabla S|$ is the normal at the free surface. In this thesis the free surface is described locally in every S cell using a height function. In two dimensions (the situation in three dimensions is explained below), based on the orientation of the free surface in the S cell, either a horizontal or vertical height function is defined using the VOF fractions in a 3 × 3 block of cells around the S cell. This is illustrated in figure 2.17. In this figure the free surface is approximately vertical, which is concluded by inspecting differences of VOF fractions (in absolute value). In this case the difference of the eastern and western VOF fraction equals $|0.9 - 0.0| = 0.9$, which is larger than $|0.0 - 0.7| = 0.7$, the difference of the northern and southern VOF fraction. Another method for determining the approximate orientation of the free surface is based on the reconstructed normal $\boldsymbol{n} = (n_x, n_y)^T$ of the free surface (see also section 2.4.1). If $|n_x \delta x| > |n_y \delta y|$ (where δx and δy denote the mesh size of the S cell in x- and y-direction respectively), a vertical height function is defined. Thus, for the centre S cell in figure 2.17, a vertical local height function $x = h\,(y, t)$ is defined. In terms of the level-set function this corresponds to $S\,(x, y, t) \equiv h\,(y, t) - x = 0$. In this case the mean curvature can be written as

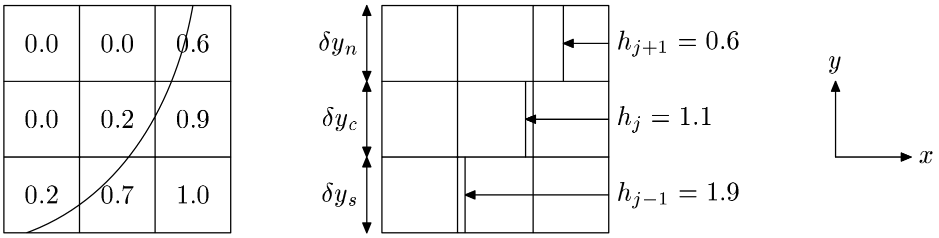


Figure 2.17: *Free surface and VOF fractions (left) and discrete, vertical local height function corresponding to the centre S cell (right).*

$$\kappa = \frac{\partial}{\partial y}\left(\frac{\partial h/\partial y}{\sqrt{1 + (\partial h/\partial y)^2}}\right). \tag{2.27}$$

The local height function h is discretised by adding the VOF fractions in three rows (see figure 2.17). Next, at grid lines $y = j - \frac{1}{2}$ and $y = j + \frac{1}{2}$, the derivative $\partial h/\partial y$ is discretised using standard finite differences, *i.e.*

$$\left.\frac{\partial h}{\partial y}\right|_{j-\frac{1}{2}} \doteq \frac{h_j - h_{j-1}}{\frac{1}{2}\delta y_s + \frac{1}{2}\delta y_c} \equiv h'_s \quad \text{and} \quad \left.\frac{\partial h}{\partial y}\right|_{j+\frac{1}{2}} \doteq \frac{h_{j+1} - h_j}{\frac{1}{2}\delta y_c + \frac{1}{2}\delta y_n} \equiv h'_n.$$

With these discretisations the term between brackets in equation (2.27) can be computed at the same positions $y = j \pm \frac{1}{2}$, whence the discrete mean curvature in the centre S cell is given by

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

Although computing the curvature is slightly more complicated in three dimensions, the procedure is the same as in two dimensions. The normal at the free surface in an S cell is classified as directed primarily in either the x-, y-, or z-direction. Based on this orientation, a local height function is formed, *e.g.* $S(x, y, z, t) \equiv h(x, y, t) - z = 0$. In this case, the mean curvature equals

$$\kappa = \frac{\partial}{\partial x} \left(\frac{\partial h / \partial x}{\sqrt{1 + (\partial h/\partial x)^2 + (\partial h/\partial y)^2}} \right) + \frac{\partial}{\partial y} \left(\frac{\partial h / \partial y}{\sqrt{1 + (\partial h/\partial x)^2 + (\partial h/\partial y)^2}} \right). \quad (2.28)$$

Since this expression contains second-order mixed derivatives, a local height is needed in 9 columns of cells around the centre S cell (see figure 2.18). For computing the discrete height function in these 9 columns, VOF fractions are added as in two dimensions (hereto the 26 cells surrounding the S cell and the S cell itself are used). The curvature κ in (2.28) is then discretised as

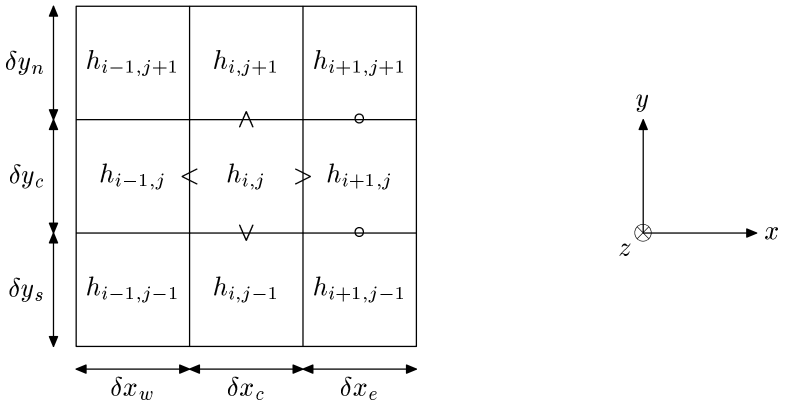


Figure 2.18: *Notation used in the formation of a three-dimensional local height function for the centre S cell.*

$$\kappa \doteq \frac{n_e - n_w}{\delta x_c} + \frac{n_n - n_s}{\delta y_c},$$

where n_e and n_w are discretisations of the first term between brackets in (2.28) at the positions indicated by $>$ and $<$ in figure 2.18 respectively and n_n and n_s are discretisations of the second term between brackets at the positions indicated by $\wedge$ and $\vee$ respectively. For computing n_e, for example, both $\partial h/\partial x$ and $\partial h/\partial y$ are needed at $>$. The former is discretised as

$$\left.\frac{\partial h}{\partial x}\right|_{>} \doteq \frac{h_{i+1,j} - h_{i,j}}{\frac{1}{2}\delta x_c + \frac{1}{2}\delta x_e}.$$

For the latter, $\partial h/\partial y$ is discretised at four horizontal cell faces (indicated by $\wedge$, $\vee$, and $\circ$) using expressions similar to the one for $\partial h/\partial x$. These four values are then averaged to obtain a value for $\partial h/\partial y$ at $>$.

Contact Angle

At the intersection of the free surface and the solid body, which is called the contact line, a boundary condition is needed for computing the mean curvature at the free surface. This is given by a static contact angle θ, which is the angle between the normal of the free surface and the normal of the solid body at the contact line. The discretisation of the contact angle is first explained in two dimensions, thereafter in three dimensions.

In two dimensions the contact line is a set of points. A typical configuration is sketched in the left of figure 2.19, where the contact line is indicated by • and the normal of the solid body (pointing into the flow domain) is denoted by $\boldsymbol{n}_b$ (see also [44], where only staircase approximations of the solid body are used for prescribing the contact angle). Now, for a given contact angle, two possible directions of the normal $\boldsymbol{n}_s$ of the free surface remain. Based on the location of the liquid a definite choice of the normal $\boldsymbol{n}_s$ is made. For example, in figure 2.19 the liquid is positioned in the lower part, whence the normal of the free surface points upward. With this normal vector, the local height function in S cells near the solid body can be computed as is demonstrated in the right of figure 2.19. In this figure the free surface is approximately horizontal, whence a horizontal height function is formed for the centre S cell. Since the cell on the left of this cell is a B cell, no liquid height can be computed in the left-hand column of cells. Hence, in this column, the height is set to a value h_w, such that the line through h_w and h_c is perpendicular to the normal vector $\boldsymbol{n}_s$. Note that in this example the boundary of the solid body is vertical and the free surface is approximately horizontal. In such a case, where the normal of the solid boundary and the normal of the free surface are classified as being perpendicular, it

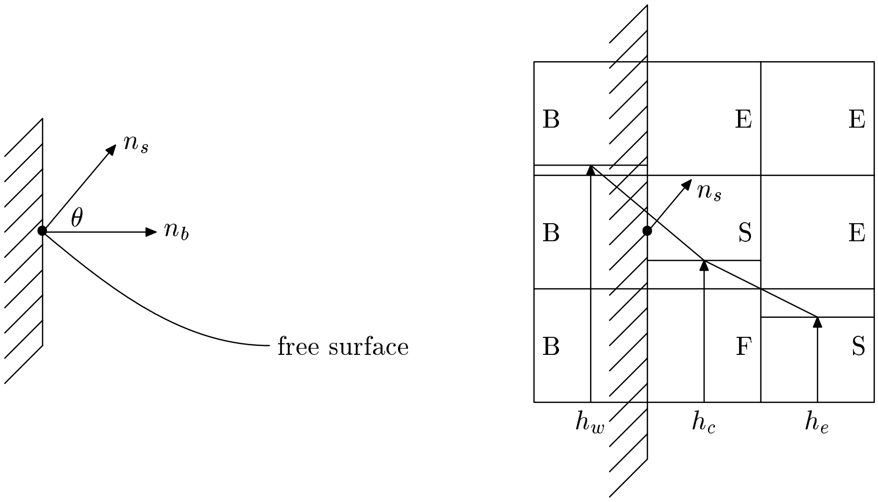


Figure 2.19: *Left: The contact angle θ is the angle between the normal $\boldsymbol{n}_b$ of the solid boundary and the normal $\boldsymbol{n}_s$ of the free surface at the contact line (which is indicated by •). Right: Horizontal local height function near a vertical wall.*

is sufficient to consider SB cell faces for applying the contact angle in computing the local height function. If the normal of the solid boundary and the normal of the free surface at the contact line have approximately the same direction, the occurrence of an SB cell face is not sufficient for applying the contact angle. This is illustrated in figure 2.20, where a vertical height function for the centre S cell is computed near a vertical wall. In this example the presence of an E cell above the S cell is necessary in order to discretise the contact angle. This is done by setting the height in the top row to a value h_n, such that the line connecting h_n and h_s is perpendicular to $\boldsymbol{n}_s$.

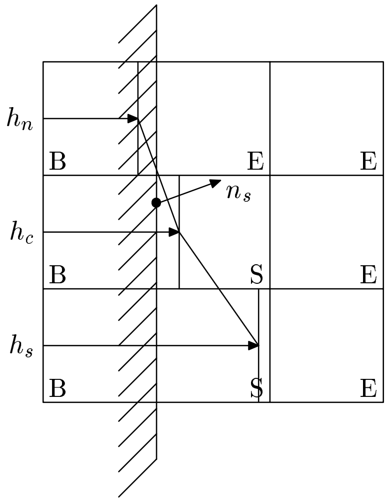


Figure 2.20: *Vertical local height function near a vertical wall.*

Some care has to be taken in computing the height function near the boundary of the solid body. For example, in the bottom row in figure 2.20, it is not sufficient to add the VOF fractions in the three cells since then only the contribution of the S cell is accounted for. Hence, instead of adding the VOF fractions, the quantity $F^s + 1 - F^b$ (which is 1 in B cells that are completely filled with liquid) is added in the three cells for computing the discrete height.

Determining the normal $\boldsymbol{n}_s$ in three dimensions is more complicated than in two dimensions (see figure 2.21). Indeed, if the normal $\boldsymbol{n}_b$ at the solid boundary is given, then an infinite number of vectors make an angle θ with this normal vector. Two of these vectors are perpendicular to the tangent $\boldsymbol{t}_s$ of the contact line and therefore are candidates for $\boldsymbol{n}_s$. Based on the location of the liquid (like in two dimensions), a final decision for the normal of the free surface can be made. Thus, to be able to determine the direction of the normal of the free surface, both the normal of the solid boundary and the tangent to the contact line are needed. For the computation of the former a technique described in section 2.4.1 (for the computation of normals at the free surface) is used. The computation of the tangent $\boldsymbol{t}_s$ is based on the local height function. Hereto, first, an approximation of the direction of the contact line is determined by looking at the approximate Cartesian directions of the solid boundary and the free surface. For example, if the largest component of the normal of the solid boundary is the x-component and if the local height function for the S cell under consideration is formed in z-direction, then a first guess for $\boldsymbol{t}_s$ is given by $\left(0, 1, h'_y\right)$, where h'_y is a finite-difference approximation

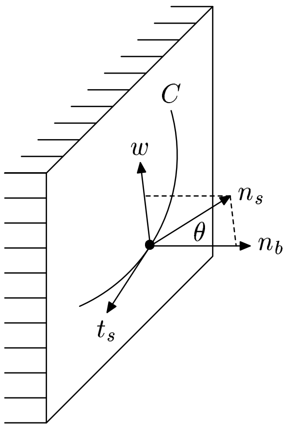
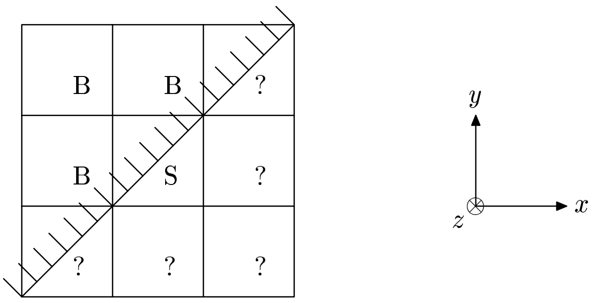


Figure 2.21: *In three dimensions the normal $\boldsymbol{n}_s$ of the free surface makes an angle θ with the normal $\boldsymbol{n}_b$ of the solid boundary and lies in the plane spanned by $\boldsymbol{n}_b$ and $\boldsymbol{w} = \boldsymbol{t}_s \times \boldsymbol{n}_b$, where $\boldsymbol{t}_s$ is the tangent to the contact line C.*

of $\partial h/\partial y$. Next, $\boldsymbol{t}_s$ is found by projecting this initial guess on the plane with normal $\boldsymbol{n}_b$. Finally, the normal $\boldsymbol{n}_s$ of the free surface at the contact line is given by a linear combination of $\boldsymbol{n}_b$ and $\boldsymbol{t}_s \times \boldsymbol{n}_b$, such that $\boldsymbol{n}_s$ makes an angle θ with $\boldsymbol{n}_b$ and points from the liquid into the air.

Similar to the technique described in two dimensions, the normal of the free surface at the contact line is used for constructing fictitious heights in B cells. Hereto, two different cases are considered. In the first case, the approximate direction of the solid boundary (the largest component of $\boldsymbol{n}_b$) is not equal to the approximate direction of the free surface (in two dimensions this situation corresponds to figure 2.19). In this case, if an S cell for which the curvature needs to be computed has a neighbouring B cell, then a vector $\boldsymbol{n}_s$ is computed in the S cell following the procedure described above. This vector is used for computing a local height in the B cell. Note that in three dimensions an S cell may be surrounded by several B cells. In such a situation, a fictitious height is computed in every B cell using the same vector $\boldsymbol{n}_s$. For example, in figure 2.22, if a height function is constructed in z-direction, then in the northern, western, and north-western B cell



B	B	?
B	S	?
?	?	?

Figure 2.22: *Top view of label configuration around an S cell. The local height function in the centre S cell is constructed in z-direction, whence fictitious heights are required in the B cells.*

a fictitious height is computed using the normal of the free surface at the contact line. In the cells indicated with a question mark (either F, S, or E cells) a regular height is computed using the volume-of-fluid function F^s. In the second case, the approximate direction of the solid boundary is equal to the approximate direction of the free surface. In this case the situation is in fact two-dimensional and figure 2.20 is applicable.

2.4 Free-Surface Reconstruction and Advection

At the end of a time cycle, when a velocity field has been computed throughout the computational grid, the free surface needs to be advected. If the location of the free surface at time t is given by

$$S\left(x, y, z, t\right) = 0, \tag{2.29}$$

then its temporal evolution follows from

$$\frac{DS}{Dt} \equiv \frac{\partial S}{\partial t} + \left(\boldsymbol{u} \cdot \nabla\right) S = 0, \tag{2.30}$$

stating that the free surface propagates with the liquid velocity. In this thesis the location of the free surface is not described explicitly by an equation (2.29). Instead, a free-surface aperture or volume-of-fluid (VOF) function F^s is defined, denoting the fractional volume of a computational cell that is occupied by fluid (see also section 2.2.1). Based on the VOF function, the free surface is reconstructed, which is explained in section 2.4.1. Then, the reconstructed free surface is advected in time, giving a VOF function at the new time level (this is explained in section 2.4.2).

2.4.1 Piecewise-Linear Interface Reconstruction

In this section a method for reconstructing the free surface is discussed. In every cell with a free-surface aperture F^s between zero and F^b (such a cell will be referred to as a mixed cell) a linear approximation, say

$$n_x x + n_y y + n_z z = c, \tag{2.31}$$

of the free surface is constructed. For this, first, the normal $\boldsymbol{n} = \left(n_x, n_y, n_z\right)^T$ of the free surface is approximated. Next, the plane constant c is computed such that the fractional volume of the computational cell for which $n_x x + n_y y + n_z z > c$ is equal to F^s. First, the method is explained for cells not close to the solid boundary. Then, some modifications for cells near the solid boundary are presented.

Computation of the Normal

For computing an approximation of the normal of the free surface in a mixed cell, consider the VOF fractions in the mixed cell itself and its 8 nearest neighbours (in three dimensions its 26 nearest neighbours); see figure 2.23 for an illustration. Analytically, the normal of the free surface is given by $\boldsymbol{n} = \nabla F^s / |\nabla F^s|$. An approximation of the interface normal

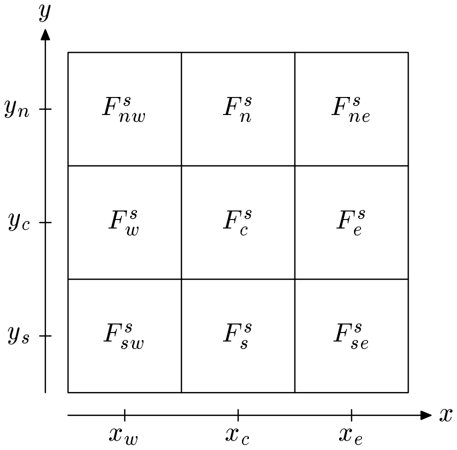


Figure 2.23: *Notation used in computing the normal of the free surface in the centre cell.*

is found by discretising this equation with respect to all the neighbours of the mixed cell, *i.e.*

$$\begin{pmatrix} x_e - x_c & 0 \\ x_e - x_c & y_n - y_c \\ 0 & y_n - y_c \\ x_w - x_c & y_n - y_c \\ x_w - x_c & 0 \\ x_w - x_c & y_s - y_c \\ 0 & y_s - y_c \\ x_e - x_c & y_s - y_c \end{pmatrix} \begin{pmatrix} \tilde{n}_x \\ \tilde{n}_y \end{pmatrix} = \begin{pmatrix} F^s_e - F^s_c \\ F^s_{ne} - F^s_c \\ F^s_n - F^s_c \\ F^s_{nw} - F^s_c \\ F^s_w - F^s_c \\ F^s_{sw} - F^s_c \\ F^s_s - F^s_c \\ F^s_{se} - F^s_c \end{pmatrix}. \tag{2.32}$$

In this equation the normal is not normalised yet (which is indicated by the tilde symbol). This system of linear equations is written in matrix-vector form as $\mathcal{A}\tilde{\boldsymbol{n}} = \boldsymbol{b}$. By multiplying both the left-hand side and the right-hand side with $\mathcal{A}^T$, a 2×2 (3×3 in three dimensions) linear system for $\tilde{\boldsymbol{n}}$ results, which is solved using Gaussian elimination. The normal of the free surface is then computed as $\boldsymbol{n} = \tilde{\boldsymbol{n}}/|\tilde{\boldsymbol{n}}|$.

Computation of the Plane Constant

Given the normal $\boldsymbol{n}$ of the free surface in a mixed cell, the plane constant c in equation (2.31) is computed such that the fractional volume of the mixed cell for which $n_x x + n_y y + n_z z > c$ equals the VOF value F^s in that cell. The correct value of c is found using a bisection method. For computing initial lower and upper bounds in this iterative method, first, in all vertices (four in two dimensions and eight in three dimensions) of the mixed cell the value of c and the corresponding VOF value is determined (the method for computing the VOF value for a given normal and plane constant is explained below). See figure 2.24 for a graphical explanation in two dimensions. For example, the line with normal $(n_x, n_y)^T$ passing through the north-eastern vertex is given by $n_x x + n_y y = c_{ne}$, where $c_{ne} = n_x x_e + n_y y_n$, and the corresponding VOF value F_{ne} is approximately equal to 0.75 in this figure (if the fluid is located below the line). By comparing the VOF values corresponding to the vertices of the mixed cell to the actual VOF value F^s, a lower bound

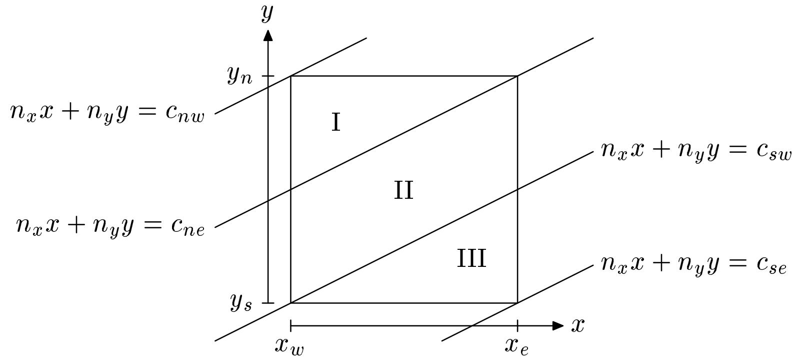


Figure 2.24: *For computation of the plane constant a lower and upper bound is computed by comparing the VOF values corresponding to the lines passing through the vertices of a mixed cell to the actual value F^s.*

c_l and an upper bound c_u for the value of the plane constant c can be found. Then, for the intermediate value $c_m = (c_l + c_u)/2$ the corresponding VOF value is computed and compared to F^s. Based on this comparison, either c_l or c_r is set to c_m. This process is repeated until the difference between the computed VOF value and F^s is smaller than a predefined tolerance.

Of course, an iterative method for finding the plane constant is not necessary in two dimensions. Indeed, the correct value of c can be found by solving a linear equation (in the parallelogram region II in figure 2.24) or a quadratic equation (in the triangular regions I and III). However, in three dimensions, five different regions can be distinguished and in two of these regions a cubic equation has to be solved for computing the VOF value. Since in these two regions an iterative method is needed, this iterative method is applied in all the regions for reasons of consistency. An obvious drawback of this approach is the extra amount of computing time that is required for the iterative method [67]. However, this iteration is performed in mixed cells only, which generally form a two-dimensional subset in three-dimensional space. Hence, the computing time required for this iteration is negligible compared to the iteration method for solving the pressure Poisson equation (see section 2.3.5). The same reason justifies the choice of the bisection method rather than a more efficient iterative method. Moreover, the bisection method is guaranteed to converge.

Computation of the VOF Value for Given Normal and Plane Constant

In the iterative method described above, the fractional volume of a mixed cell that is truncated by an equation (2.31) has to be computed for a given normal $\boldsymbol{n}$ of the free surface and a given plane constant c. This is explained in two dimensions using figure 2.25. The shaded region Ω represents the part of the cell for which $n_x x + n_y y > c$. The volume V of this region is given by (see also [19, 45])

$$V = \int\limits_{\Omega} dV = \frac{1}{2} \int\limits_{\Omega} \nabla \cdot \begin{pmatrix} x \\ y \end{pmatrix} dV = \frac{1}{2} \int\limits_{\Omega} \nabla \cdot \begin{pmatrix} x - cn_x \\ y - cn_y \end{pmatrix} dV.$$

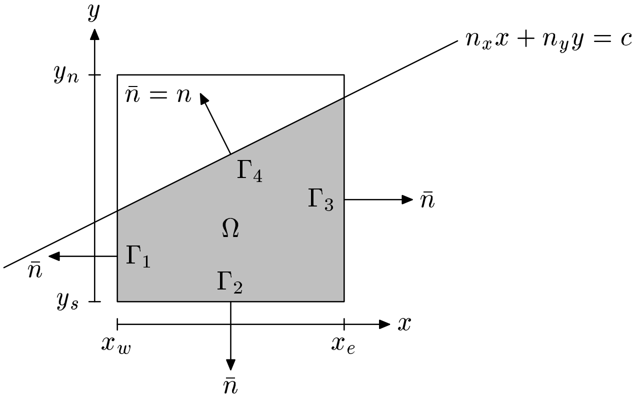


Figure 2.25: *For a given normal* $\boldsymbol{n}$ *and plane constant* c*, the volume of* Ω *(the shaded region) is computed using a boundary integral over* $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$*, which can be rewritten as a boundary integral over the cell faces.*

Using the divergence theorem, this last integral is transformed into a boundary integral over the boundary Γ of Ω, which results in

$$V = \frac{1}{2} \oint\limits_{\Gamma} \begin{pmatrix} x - cn_x \\ y - cn_y \end{pmatrix} \cdot \bar{\boldsymbol{n}}\, dS,$$

where $\bar{\boldsymbol{n}}$ is the outward-pointing normal on Γ. The boundary Γ consists of four parts Γ_1, Γ_2, Γ_3, and Γ_4. On the former three, the normal $\bar{\boldsymbol{n}}$ is aligned with the x- or y-axis. On the latter, $\bar{\boldsymbol{n}}$ is equal to $\boldsymbol{n}$. Hence, the volume V is equal to

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

In this equation the last integral vanishes since $n_x x + n_y y = c$ on Γ_4. The remaining integrals are rewritten as integrals over the cell faces, *i.e.*

$$\begin{aligned} V = &- \frac{x_w - cn_x}{2} \int\limits_{y_s}^{y_n} \mathbf{1}_{\{n_x x_w + n_y y > c\}}\, dy - \frac{y_s - cn_y}{2} \int\limits_{x_w}^{x_e} \mathbf{1}_{\{n_x x + n_y y_s > c\}}\, dx \\ &+ \frac{x_e - cn_x}{2} \int\limits_{y_s}^{y_n} \mathbf{1}_{\{n_x x_e + n_y y > c\}}\, dy + \frac{y_n - cn_y}{2} \int\limits_{x_w}^{x_e} \mathbf{1}_{\{n_x x + n_y y_n > c\}}\, dx, \end{aligned} \tag{2.33}$$

where the last term, an integral over the northern cell face, is added to obtain a formula that can be applied in every mixed cell. Since $(x, y) \mapsto n_x x + n_y y$ is linear in x and y, the integrals that appear in equation (2.33) are easily computed. For example, the integral over the western cell face is computed as

$$\int\limits_{y_s}^{y_n} \mathbf{1}_{\{n_x x_w + n_y y > c\}}\, dy = (y_n - y_s) \frac{\max\{c_{nw}, c_{sw}\}}{\max\{c_{nw}, c_{sw}\} - \min\{c_{nw}, c_{sw}\}},$$

where $c_{nw} = n_x x_w + n_y y_n$ and $c_{sw} = n_x x_w + n_y y_s$. In this equation it is assumed that $c_{nw} c_{sw} < 0$. If this is not the case, then either $c_{nw} > 0$ and $c_{sw} > 0$, whence the integral equals $y_n - y_s$, or $c_{nw} \leq 0$ and $c_{sw} \leq 0$, whence the integral equals zero. The advantage of this method is that the integral over the reconstructed interface is removed; only integrals over the cell faces have to be computed. Further, the method is generic in the sense that no different cases have to be considered; equation (2.33) is applied in every mixed cell.

In three dimensions a similar method is used. The volume V of a cell that is truncated by an equation (2.31) is written as a boundary integral over the six cell faces, *i.e.*

$$\begin{aligned} V = &- \frac{x_l - cn_x}{3} \int\limits_{y_l}^{y_r} \int\limits_{z_l}^{z_r} \mathbf{1}_{\{n_y y + n_z z > c - n_x x_l\}}\, dydz - \frac{y_l - cn_y}{3} \int\limits_{x_l}^{x_r} \int\limits_{z_l}^{z_r} \mathbf{1}_{\{n_x x + n_z z > c - n_y y_l\}}\, dxdz \\ &- \frac{z_l - cn_z}{3} \int\limits_{x_l}^{x_r} \int\limits_{y_l}^{y_r} \mathbf{1}_{\{n_x x + n_y y > c - n_z z_l\}}\, dxdy + \frac{x_r - cn_x}{3} \int\limits_{y_l}^{y_r} \int\limits_{z_l}^{z_r} \mathbf{1}_{\{n_y y + n_z z > c - n_x x_r\}}\, dydz \\ &+ \frac{y_r - cn_y}{3} \int\limits_{x_l}^{x_r} \int\limits_{z_l}^{z_r} \mathbf{1}_{\{n_x x + n_z z > c - n_y y_r\}}\, dxdz + \frac{z_r - cn_z}{3} \int\limits_{x_l}^{x_r} \int\limits_{y_l}^{y_r} \mathbf{1}_{\{n_x x + n_y y > c - n_z z_r\}}\, dxdy, \end{aligned}$$

where $[x_l, x_r] \times [y_l, y_r] \times [z_l, z_r]$ are the dimensions of the mixed cell under consideration. For each of the six integrals in this equation, the two-dimensional method that has been described above is applied, *i.e.* each double integral is replaced by an equation similar to (2.33). Note that the method in this section is mathematically more elegant (and on a parallel computer more efficient) than a case-by-case method for computing VOF fractions as is done in the original method of Youngs [96].

Near the Solid Boundary

The methods for computing the normal of the free surface and the VOF value for a given normal and plane constant require some modifications in the vicinity of the solid boundary.

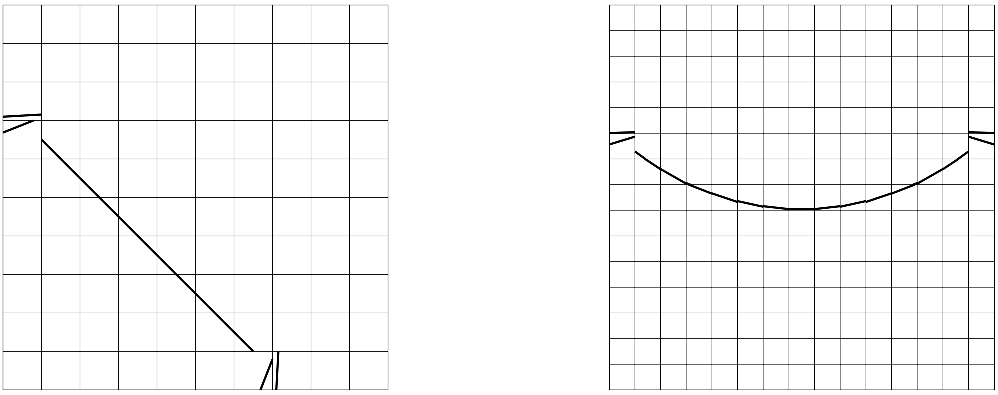


Figure 2.26: *Reconstruction of a line (left) and part of a circle (right). In B cells the value of F^s is set to zero and these cells are taken into account in system* (2.32).

The basic procedure for the computation of the normal is the same. However, near the solid boundary, some cells neighbouring the mixed cell under consideration may be entirely closed for flow ($F^b = 0$). A typical example is shown in figure 2.19, where the three cells in the left-hand column are B cells. In these cells the value of F^s is equal to zero and should not be used in the computation of the normal as is illustrated in figure 2.26. In this figure, a straight line (on the left) and part of a circle (on the right) are reconstructed on Cartesian grids with 10×10 and 15×15 cells respectively (the fluid is located below the line or circle). The value of F^s is equal to zero in B cells. Clearly, the reconstructed normal is not correct in mixed cells adjacent to B cells. If the contribution of B cells is removed from the system (2.32), then the reconstructed normals in mixed cells near the solid boundary are more accurate as is demonstrated in figure 2.27. Instead of ignoring B cells in computing the normal of the free surface, it is possible to take these B cells into account with suitable values of F^s. For example, if a straight line passes through the centre of the centre cell in figure 2.19 and is continued in the solid body, then, using the notation of figure 2.23, it follows that

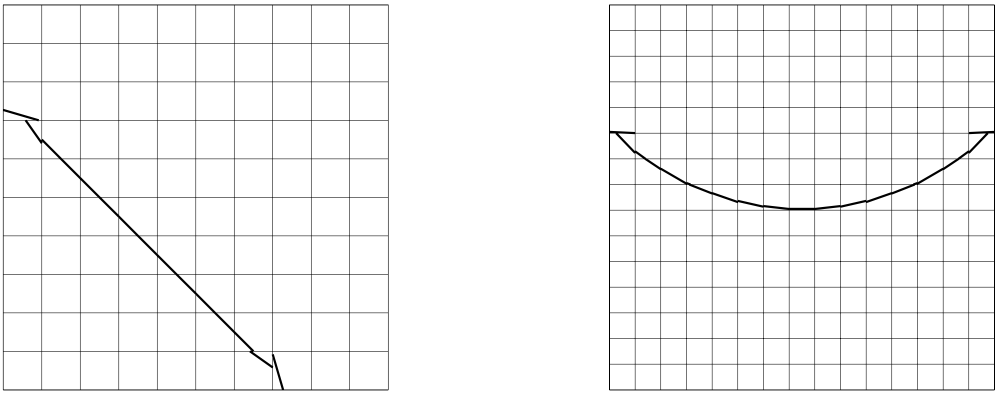


Figure 2.27: *Reconstruction of a line (left) and part of a circle (right). The contribution of B cells is removed from system* (2.32).

$$F^s_{nw} + F^s_{se} = 1, \quad F^s_{w} + F^s_{e} = 1, \quad \text{and} \quad F^s_{sw} + F^s_{ne} = 1.$$

This observation motivates the following choice for assigning VOF values to B cells: if cell (i, j, k) is a mixed cell wherein a normal has to be reconstructed and cell (l, m, n) is a B cell (and one of the 26 neighbours of the mixed cell), then the VOF value $F^s_{l,m,n}$ in this B cell is set to $1 - F^s_{i+i-l,j+j-m,k+k-n}$. The results of this method are shown in figure 2.28. From figures 2.27 and 2.28 it becomes clear that in some mixed cells near the solid boundary removing B cells from system (2.32) gives more accurate normals and in other mixed cells "mirrored" F^s values result in a higher accuracy. Closer inspection revealed that the former method is best applied in cells with a large value of F^s and the latter in the case of small F^s values.

For computing the plane constant, the same iterative method that has been described above is used. No changes are required in the vicinity of the solid boundary. However, at every iteration level in this method, the fractional volume of a mixed cell that is truncated by the free surface has to be computed. This was done by changing a volume integral into a carefully chosen boundary integral, such that the integral along the free surface

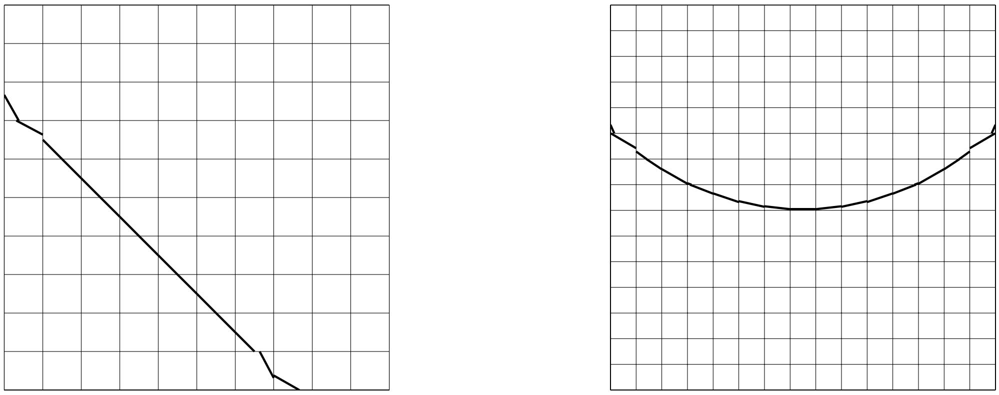


Figure 2.28: *Reconstruction of a line (left) and part of a circle (right). The contribution of B cells is taken into account, but "mirrored" values of F^s are assigned to these cells.*

vanished. If the mixed cell is cut by the solid body, this procedure is more difficult as is illustrated in figure 2.29 on the left. In this figure the dark-shaded region is not open for flow. The volume integral for computing the volume of Ω can be rewritten in such a way that the boundary integral along Γ_5 (the free surface) vanishes, but then the contribution of Γ_4 (the solid boundary) is not equal to zero. Thus, in this case, the area of Γ_4 needs to be computed, which is relatively difficult (compared to the computation of VOF values in mixed cells that are not cut by the solid body). Hence, in cut cells, the solid boundary is approximated such that it is aligned with one of the Cartesian directions as is shown in figure 2.29 on the right. The configuration in this figure is similar to the one in figure 2.25; only the value of x_e is changed.

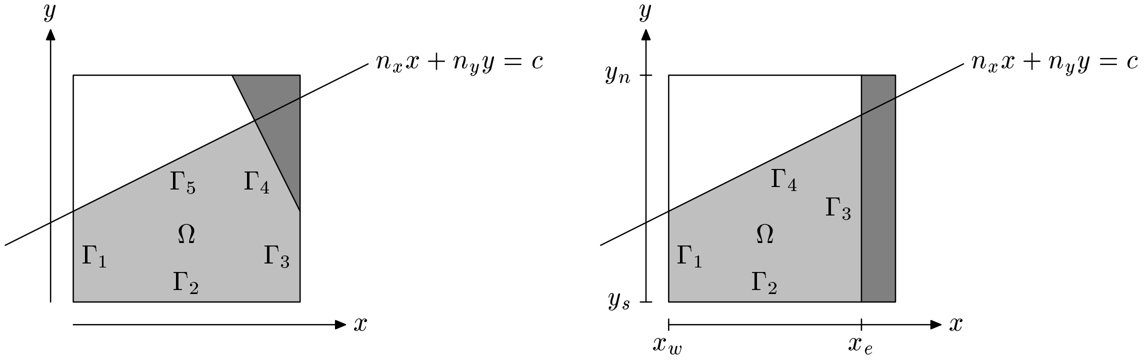


Figure 2.29: *Computation of the VOF value for a mixed cell that is cut by the solid boundary; the solid body is approximated by a line that is aligned with one of the Cartesian directions.*

2.4.2 Free-Surface Advection

Donor-Acceptor Method

Once the free surface has been reconstructed, the volume-of-fluid function F^s can be advected in time (in the remainder of this section the superscript s will be omitted for presentational reasons). Basically, this is done using the donor-acceptor method that has been developed by Hirt and Nichols [35]. At each cell face the amount of fluid, based on the face area, the liquid velocity, and the time step, that is transported from the donor cell to the acceptor cell is computed (see figure 2.30 on the left). In cells not close to the free surface, the flux δF_c across a cell face perpendicular to the x-direction is computed as

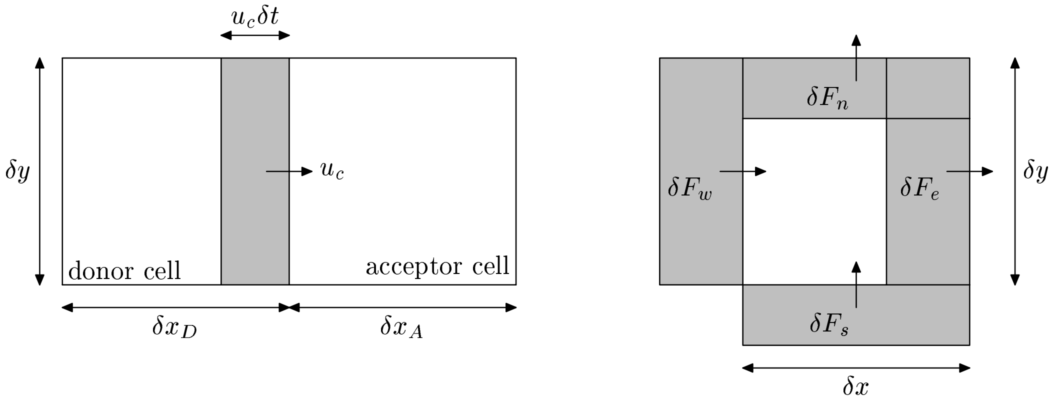


Figure 2.30: *Computation of the flux across a cell face using the donor-acceptor method.*

$$\delta F_c = u_c \delta t A_c^x \delta y, \tag{2.34}$$

where u_c is the normal velocity at the cell face, δt is the time step, and $A_c^x \delta y$ is the area of the part of the cell face that is open for flow. If for each cell face the flux has been computed (see figure 2.30 on the right), the VOF function is advected from time level n to $n+1$ by

$$F_c^b \delta x \delta y F_c^{n+1} = F_c^b \delta x \delta y F_c^n - \delta F_e - \delta F_n + \delta F_w + \delta F_s.$$

Although the basic principles of the donor-acceptor method, as described above, are simple, the computation of fluxes through cell faces near the free surface is more complicated since cells need not be completely filled with liquid here. Hence, the flux (2.34) may exceed the amount of fluid that is present in the donor cell. This is the cause of undershoots (below zero) and overshoots (above one) in the VOF function at the new time level. Hence, in the original VOF method, a flux near the free surface is computed as

$$|\delta F_c| = \min\left\{F_{AD}|u_c\delta t| + CF, F_D \delta x_D\right\}, \tag{2.35}$$

where

$$CF = \max\left\{(1 - F_{AD})\,|u_c\delta t| - (1 - F_D)\,\delta x_D, 0\right\}. \tag{2.36}$$

In these equations F_D and F_A denote the VOF values in the donor and acceptor cell respectively. The double subscript AD represents either the donor or the acceptor cell, depending on the orientation of the free surface relative to the direction of the flow. If the free surface is advected more or less normal to itself or if either the acceptor cell or the cell upstream of the donor cell is empty, then AD is set to A, otherwise AD is set to D. If the free surface is moving in tangential direction, setting AD to A would result in an unphysical steepening of the interface. The min operator in equation (2.35) prevents the fluxing of more fluid to the acceptor cell than the donor cell contains and the max operator in (2.36) accounts for an additional flux if the amount of void exceeds the amount available. More details are explained in [35].

Local Height Function

An important drawback of the original method by Hirt and Nichols is the creation of "flotsam" and "jetsam" — small bits of fluid that get (unphysically) separated from the main body of fluid (see for example figure 2.35 and the references mentioned in section 1.4) — during the advection of the free surface. Further, this method may result in a considerable gain or loss in mass (despite the min and max limiters in equations (2.35) and (2.36)) since, at the end of every time cycle, VOF values below zero are reset to zero and VOF values above one are reset to one. Therefore, a method was developed that overcomes these problems (see [46, 63] for alternative methods for removing flotsam and jetsam). Since flotsam and jetsam is created near the free surface, the advection method of Hirt and Nichols is adapted in S cells. In these cells the fluid is transported using the local height function that already has been introduced for computing the curvature of the free surface (see section 2.3.6). The idea is that not the individual fluxes at the cell faces belonging to the S cell are transported, but the height of the free surface in the row or column corresponding to the S cell is updated. This is illustrated in figure 2.31. If the local height function in the centre S cell in this figure is given by $S\left(x,y,t\right) = h\left(y,t\right) - x = 0$, then the advection equation (2.30) results in

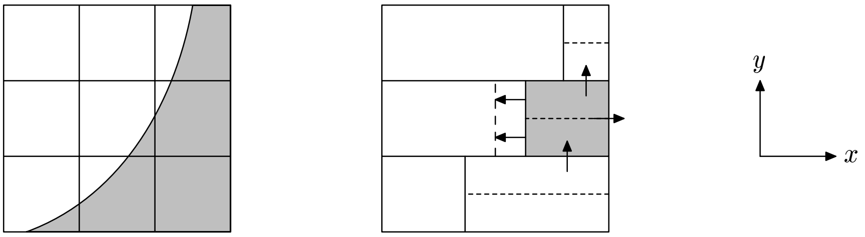


Figure 2.31: *Fluxes near the free surface are transported by changing the local height function.*

$$\frac{\partial h}{\partial t} = v\frac{\partial h}{\partial y} - u,$$

where u and v are the velocity components in x- and y-direction respectively.

Youngs' Method

The limiters in equations (2.35) and (2.36) of the original VOF method are necessary to prevent severe undershoots and overshoots in the VOF function and to prevent the free surface from diffusing. The main reason for this is the characterisation of the interface to be either horizontal or vertical (in two dimensions). With the piecewise-linear interface reconstruction that has been explained in section 2.4.1, a more accurate method for computing fluxes across cell faces near the free surface can be applied. This method was introduced by Youngs [96] and is illustrated in figure 2.32. Instead of fluxing an amount

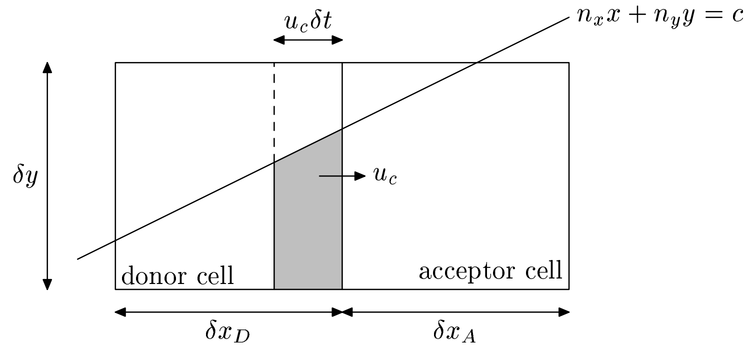


Figure 2.32: *Computation of the flux across a cell face using the piecewise-linear reconstruction of the interface.*

of $u_c \delta t A_c^x \delta y$, only the part of this volume that is inside the fluid ($n_x x + n_y y > c$) is fluxed (the shaded region in the figure). The region that has to be fluxed is computed with the same method that has been explained in the previous section; the volume integral is rewritten as a boundary integral along the boundary of the shaded region in such a way that the contribution of the free surface to this integral vanishes. The adaptation of the original VOF method to prevent the creation of flotsam and jetsam can also be applied in Youngs' method; the fluxes are computed as described above, however, the advection of the VOF function in S cells is performed using the local height function.

When the local height function is used for the advection of the free surface, the liquid mass is better conserved since fewer undershoots and overshoots in the VOF function occur. This important feature of the local height function is demonstrated in sections 2.5.1 and 2.5.2, where four advection methods — both the methods of Hirt and Nichols and of Youngs with and without the local height function — are compared with each other.

2.5 Results

2.5.1 Free-Surface Advection

An important part of the numerical model deals with the free-surface reconstruction and advection (see section 2.4). In this section some results regarding the advection of the free surface are presented. Four methods are compared with each other, namely Hirt and Nichols, Hirt and Nichols combined with the local height function, Youngs, and Youngs combined with the local height function (details of these methods have been discussed in section 2.4.2). The results in this section have been obtained with a prescribed velocity field; thus the Navier-Stokes equations are not solved. This makes it possible to test the free-surface reconstruction and advection only [42, 61].

Translation Test

First, some simple translation tests are performed. An initially circular or square body of fluid is placed in two different velocity fields. The first velocity field is aligned with the y-axis ($(u, v) = (0, 1)$). In the second velocity field the fluid is transported diagonally through the Cartesian grid ($(u, v) = (2, 1)$). In all the simulations, the fluid is transported in 1000 time steps over a distance that is approximately five times the diameter of the fluid. The number of grid cells along the diameter of the fluid is equal to 15. The results are shown in figure 2.33. From this figure it is clear that Youngs' method is more accurate than the method of Hirt and Nichols. Especially in the diagonal translation of a circle, Youngs' method produces qualitatively better results. Further, the methods with the use of a local height function perform somewhat worse than the methods without its use.

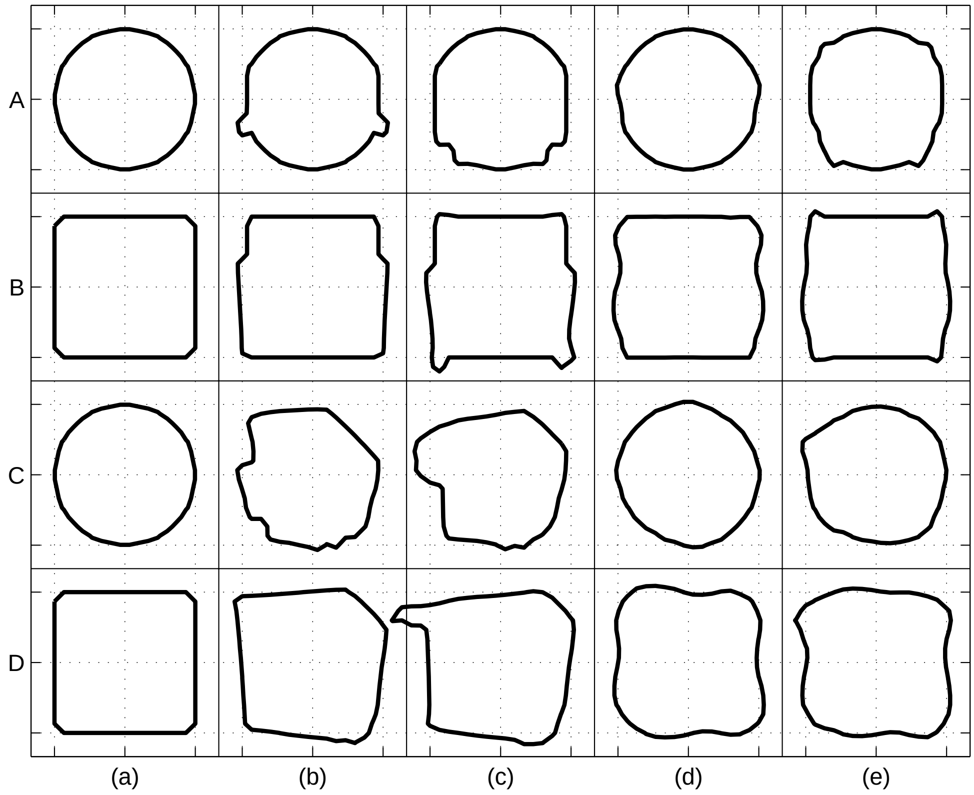


Figure 2.33: *Translation of a circle (rows A and C) and a square (rows B and D) in two different velocity fields* $(u, v) = (0, 1)$ *(rows A and B) and* $(u, v) = (2, 1)$ *(rows C and D). (a) Initial condition, (b) Hirt and Nichols without local height function, (c) Hirt and Nichols with local height function, (d) Youngs without local height function, and (e) Youngs with local height function.*

Rotation Test

A more difficult test case is the solid-body rotation of a fluid region. Again, an initially circular and square body of fluid is considered. The distance between the centre of rotation and the centre of mass of the fluid is approximately twice the diameter of the fluid region. In about 600 time steps the fluid is rotated 360 degrees. The number of grid cells along the diameter of the fluid is equal to 15. The results are shown in figure 2.34. The small drops of fluid that appear in column (a) actually belong to column (b). This flotsam and jetsam is typical for the original method of Hirt and Nichols. Clearly, Youngs'

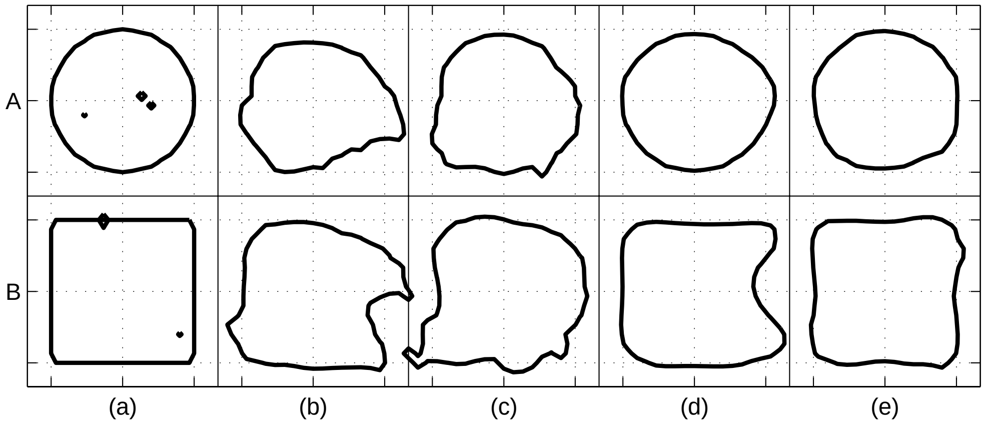


Figure 2.34: *Solid-body rotation of a circle (row A) and a square (row B). (a) Initial condition, (b) Hirt and Nichols without local height function, (c) Hirt and Nichols with local height function, (d) Youngs without local height function, and (e) Youngs with local height function.*

method produces better results than the method of Hirt and Nichols. However, also in Youngs' method the square region of fluid is quite distorted if the height function is not used. In the solid-body rotation the differences between the methods with and without the local height function are smaller than in the relatively simple translation tests. For the solid-body rotation of a square body of fluid, the use of a local height function even improves Youngs' method.

Single Vortex

The translation and rotation tests that have been discussed above are simple advection tests in the sense that (theoretically) the fluid topology does not change in time. Although these tests give some kind of guarantee that the advection method is correctly implemented, they do not give a definite indication of the capabilities of the method since in most practical applications the topology of the fluid changes considerably in time (*e.g.* breaking waves). Therefore, a more complex advection test, proposed by Rider and Kothe [61], is performed (see also [62]). A circle, with radius 0.15 and centred at $(0.50, 0.75)$, is placed in a square computational domain of unit dimensions. The circle is advected with a solenoidal velocity field $(u, v) = (-d\Phi/dy, d\Phi/dx)$, where

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

This velocity field causes the circular body of fluid to stretch and spiral around the centre of the domain as is shown in figure 2.35. The simulation is continued for T seconds. After $T/2$ seconds the flow field is reversed, such that at the end of the simulation, ideally, the fluid returns to its initial condition. In figure 2.35, snapshots of the simulation at $t = T/2$ and $t = T$ are shown for grids consisting of 60×60 and 100×100 cells and for simulation times $T = 2$ and $T = 4$. As for the translation and rotation tests, four different advection methods are studied. For the single vortex, the creation of flotsam and jetsam in the

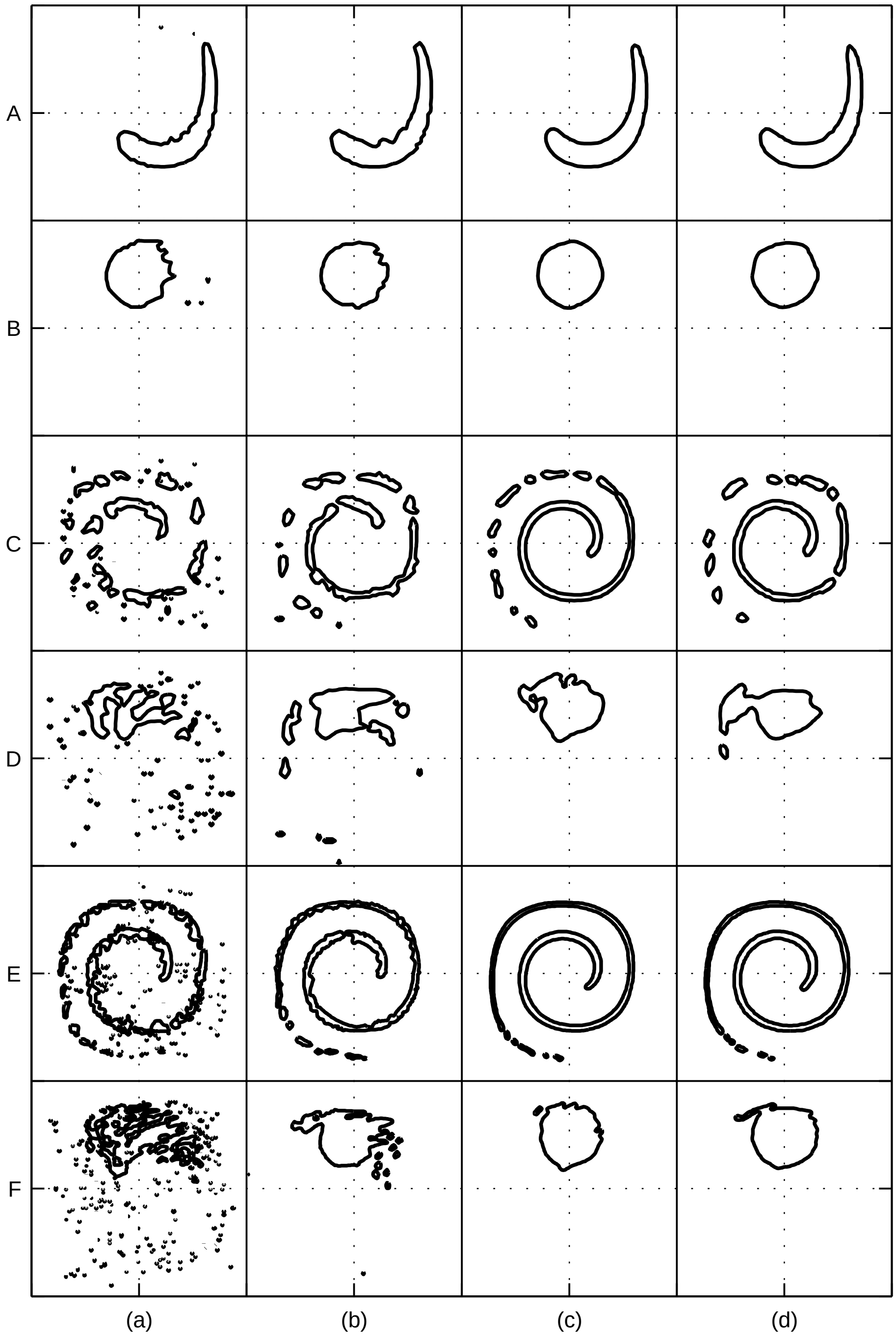


Figure 2.35: *Advection of a circular body of fluid placed in a single-vortex flow field. The maximum simulation time is equal to* $T = 4$ *except for rows A and B, where* $T = 2$*. Snapshots are shown at time* $T/2$ *(rows A, C, and E) and at time* T *(rows B, D, and F). The computational grid consists of* 60×60 *cells (rows A, B, C, and D) or* 100×100 *cells (rows E and F). (a) Hirt and Nichols without local height function, (b) Hirt and Nichols with local height function, (c) Youngs without local height function, and (d) Youngs with local height function.*

original method of Hirt and Nichols becomes very apparent. If a local height function is used in combination with this method, then the fluid remains much more coherent. Qualitatively, Youngs' method reconstructs the circle better than the method of Hirt and Nichols. Indeed, the original method of Youngs does not suffer from the creation of flotsam and jetsam. Hence, the use of a local height function does not affect Youngs' method much. Especially when the simulation time is long and a fine grid is used, Youngs' method reconstructs the circle better than the method of Hirt and Nichols. On the coarser grid the circle is heavily distorted for both the methods.

A more quantitative way of comparing the various advection methods is by examining the error in the VOF function F at time $t = T$ (see also [30, 61]). In table 2.1 the discrete error norm

$$\frac{1}{N}\sum_{i,j}\left|F_{i,j}\left(t=T\right)-F_{i,j}\left(t=0\right)\right|,$$

where N is the number of grid cells and the summation is over the grid cells, is presented for various values of N and T. For all the simulations corresponding to this table, a constant CFL number of 0.25 has been used. This means that on finer grids a smaller time step was chosen. Hence, the results in the table combine a grid-refinement study and a time-step analysis. Clearly, for all advection methods the error in the VOF function decreases if the grid is refined, although the number of time steps increases upon grid refinement. Except for one case, Youngs' method is superior compared to the method of Hirt and Nichols. Further, in almost all cases, the use of a local height function improves the original method of Hirt and Nichols; only when the number of time steps is small ($T = 0.5$, grids 32×32 and 64×64), the height function worsens the results. Moreover, the improvement in the method of Hirt and Nichols by using the height function is better on fine grids. For the method of Youngs the use of a local height function is less beneficiary; only when the number of time steps is very large or very small, the local height function improves the original method of Youngs. Moreover, the differences in the error between Youngs' method with and without a local height function is generally smaller than the differences between the method of Hirt and Nichols with and without the

T	grid	H&N −	H&N +	Youngs −	Youngs +
0.5	32×32	5.58×10^{-3}	6.11×10^{-3}	4.19×10^{-3}	4.01×10^{-3}
0.5	64×64	3.21×10^{-3}	3.64×10^{-3}	2.08×10^{-3}	2.33×10^{-3}
0.5	128×128	2.64×10^{-3}	2.14×10^{-3}	1.06×10^{-3}	1.34×10^{-3}
2.0	32×32	1.90×10^{-2}	1.07×10^{-2}	6.04×10^{-3}	7.23×10^{-3}
2.0	64×64	1.02×10^{-2}	5.13×10^{-3}	2.97×10^{-3}	3.19×10^{-3}
2.0	128×128	7.90×10^{-3}	3.31×10^{-3}	1.93×10^{-3}	1.63×10^{-3}
8.0	32×32	1.04×10^{-1}	8.39×10^{-2}	5.95×10^{-2}	1.10×10^{-1}
8.0	64×64	7.96×10^{-2}	6.05×10^{-2}	2.00×10^{-2}	4.49×10^{-2}
8.0	128×128	6.76×10^{-2}	3.17×10^{-2}	8.25×10^{-3}	8.02×10^{-3}

Table 2.1: *Error norm in the VOF function for the single-vortex advection test. Comparison of four advection methods: Hirt and Nichols (H&N) and Youngs, both without (−) and with (+) the use of a local height function. The CFL number is constant for all simulations.*

height function. In table 2.2 the order of convergence upon grid refinement and time-step reduction is studied. Although the variation in the ratios of the error norm is quite high,

T	grid	H&N −	H&N +	Youngs −	Youngs +
0.5	$32^2/64^2$	1.74	1.68	2.02	1.72
0.5	$64^2/128^2$	1.22	1.70	1.96	1.75
2.0	$32^2/64^2$	1.87	2.09	2.03	2.27
2.0	$64^2/128^2$	1.29	1.55	1.54	1.95
8.0	$32^2/64^2$	1.31	1.39	2.97	2.44
8.0	$64^2/128^2$	1.18	1.91	2.43	5.60

Table 2.2: *Order of convergence of error norm in the VOF function for the single-vortex advection test. Comparison of four advection methods: Hirt and Nichols (H&N) and Youngs, both without (–) and with (+) the use of a local height function.*

it is concluded that the original methods of Hirt and Nichols and of Youngs are first-order and second-order accurate respectively. This conclusion is drawn not only because of the results in table 2.2, but also because of the way the methods are constructed; while Hirt and Nichols reconstruct the interface by approximating it with horizontal and vertical line segments, the method of Youngs exploits a piecewise-linear reconstruction. The order of convergence of the former method appears to increase if it is combined with a local height function. For Youngs' method no definite conclusion can be made on the change of the order of convergence if a height function is used.

An important aspect of an advection method is the capability of conserving mass. Since undershoots and overshoots in the VOF function are cut off at the end of every time cycle (to fulfil the condition $0 \leq F^s \leq F^b$ at the beginning of a new time cycle), a considerable amount of mass may be gained or lost during a simulation. The local height

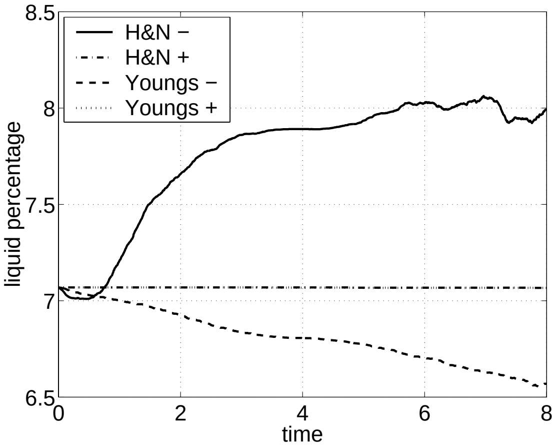


Figure 2.36: *Liquid percentage as a function of time for the single-vortex advection test (with $T = 8$ and a grid of 128×128 cells). Comparison of four advection methods: Hirt and Nichols (H&N) and Youngs, both without (–) and with (+) the use of a local height function.*

function was introduced to prevent large undershoots and overshoots in the VOF function. Hence, an advection method combined with a local height function for transporting fluxes across cell faces is expected to conserve mass better than the same advection method without the local height function. This is confirmed in figure 2.36, where the liquid percentage for the single-vortex simulation is shown as a function of time for the four different advection methods that have been studied in this section. Clearly, in the original method of Hirt and Nichols mass is gained due to undershoots in the VOF function and mass is lost in the original method of Youngs due to overshoots. If a local height function is used in combination with these methods, the liquid percentage remains constant in time.

2.5.2 Static Drop and Drop Oscillation

In this section results of simulation of fluid flow in a zero-gravity environment are presented. The liquid is not in contact with the solid body, whence only the numerical model for surface tension is validated (the numerical model for the contact angle is discussed in the next section). First, a static, circular liquid drop is studied. Then, the transformation of a square liquid drop into a circular drop under the influence of surface tension is discussed.

Static Drop

A circular body of fluid (in two dimensions), that is subjected to surface-tension forces only, is in static equilibrium. However, since the analytical VOF field need not correspond to the discrete VOF field of the steady-state solution and because of numerical errors in the boundary conditions applied at the free surface, spurious currents may occur [46, 60]. In the CSF method of Brackbill *et al.* [5], where also the motion of the surrounding air is resolved, these spurious currents are quite dramatic as they lead to an increase of the kinetic energy as has been demonstrated by Rudman [63]. In the left of figure 2.37, the evolution of the kinetic energy for the static drop is shown for the method of Hirt and Nichols and for Youngs' method, both in combination with a local height function for advecting fluid near the free surface. Clearly, in Youngs' method, the kinetic energy does not decrease; small oscillations at the free surface grow in time and the drop starts drifting away from its initial position. Apparently, this method is not able to damp small disturbances. The reason for this lies in the fact that the advection of the free surface is completely based on the linear reconstruction of the free surface in S cells. Due to small numerical errors, the reconstructed normal at the free surface is not symmetrical around the drop, whence the computed fluxes are not symmetrical. Hence, the distribution of the VOF field is not symmetrical at the next time step, which, in effect, results in an asymmetrical computation of the normal at the free surface, and so forth (see also [30], where it was concluded that the interface reconstruction is crucial for the overall performance of an advection algorithm). These problems do not occur in the method of Hirt and Nichols. Indeed, in this method the free surface is classified as being either horizontal or vertical, which is not sensitive to small asymmetries in the VOF field. For this method the kinetic energy tends to zero as is shown in the left of figure 2.37 for two different Ohnesorge numbers.

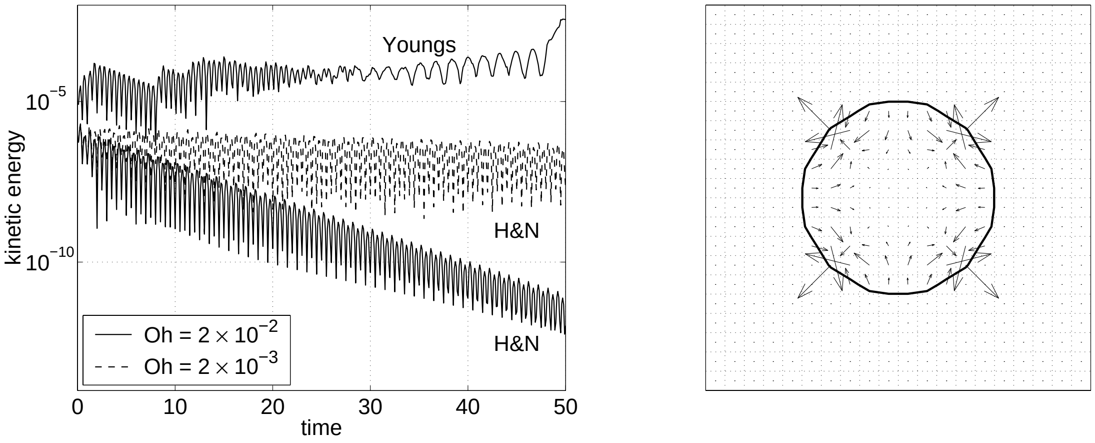


Figure 2.37: *Left: Evolution of the kinetic energy for the two-dimensional static drop for $Oh = 2 \times 10^{-2}$ and $Oh = 2 \times 10^{-3}$. Right: Snapshot of the free surface and vector plot of the velocity field after 10 time steps.*

Although in the method of Hirt and Nichols the kinetic energy decreases in time, spurious currents are present in this method as is shown in the right of figure 2.37. In this figure a snapshot of the free surface and a vector plot of the velocity field is shown after 10 time steps, clearly demonstrating the presence of spurious currents. For this simulation a circle with radius 0.25, centred in a square domain of unit dimensions, has been studied. The computational grid consisted of 20 × 20 cells and a time step of $\delta t = 10^{-2}$ was used. The Ohnesorge number $Oh = \mu/\sqrt{\rho\sigma R}$, indicating the ratio between viscous and surface-tension forces (R is a characteristic length scale, *e.g.* the radius of the circular body of fluid), was set to $Oh = 2 \times 10^{-2}$ and $Oh = 2 \times 10^{-3}$. Clearly, since Oh is proportional to the viscosity of the fluid, the kinetic energy is damped more for a higher value of Oh.

In figure 2.37 the free surface appears somewhat rippled. This is due to the post-processing software; the free surface is drawn as a contour line of the VOF field at a value 0.5. Although being more computationally expensive, a better way to show the

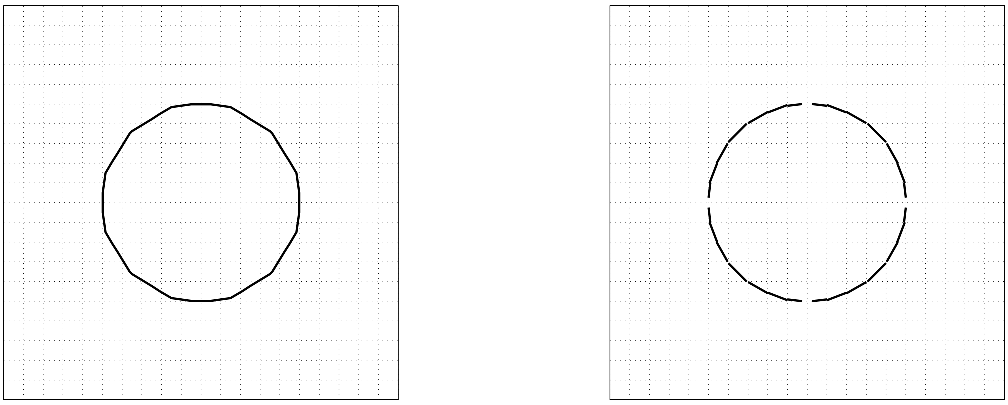


Figure 2.38: *Snapshot of the free surface drawn as a contour line of the VOF field (left) and as a piecewise-linear reconstruction (right).*

free surface is by plotting the reconstructed free surface (see section 2.4.1): in every computational cell that is cut by the free surface a line segment is drawn. The difference between the two methods is shown in figure 2.38. For convenience, however, most of the snapshots in the remaining part of this thesis show the free surface as a contour line of the VOF field.

The simulation of a static drop has been repeated in three dimensions. For this simulation the evolution of kinetic energy for two different Ohnesorge numbers is shown in figure 2.39. Although the kinetic energy decreases not so regular as in two dimensions, the simulations are stable: small disturbances are damped.

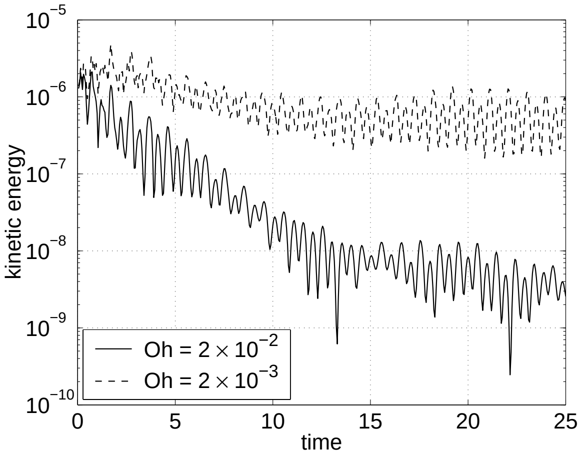


Figure 2.39: *Evolution of the kinetic energy for the three-dimensional static drop for* $Oh = 2 \times 10^{-2}$ *and* $Oh = 2 \times 10^{-3}$.

Drop Oscillation

If the initial shape of the fluid is not circular but square and the fluid is subjected to surface tension, then the fluid is not in equilibrium. Indeed, the curvature in the corners of the square is infinite and along the edges the curvature is equal to zero. Hence, the surface-tension force pulls the fluid from the corners towards the centre. Because of mass conservation and symmetry, the fluid at the centres of the edges starts flowing outward. After some time, the free surface resembles the initial configuration rotated about 45 degrees, whence the cycle starts over again. Of course, because of viscosity, the overshoots become smaller and smaller, such that, eventually, the fluid settles itself in a circular equilibrium position (see also [5, 55, 70, 72]).

Snapshots of the evolution of a square drop into a circular drop are shown in figure 2.40. For this simulation a square drop of dimensions $\frac{1}{2} \times \frac{1}{2}$ is placed in the centre of a square domain of unit dimensions. The grid consists of 40×40 cells and the time step is set to $\delta t = 10^{-3}$. The Ohnesorge number for this simulation is equal to $Oh = 0.1$. Note that in figure 2.40 the velocities are scaled with the largest velocity; in fact, the kinetic energy decreases to machine precision during the simulation.

This simulation has been performed both with the method of Hirt and Nichols and the method of Youngs. The snapshots in figure 2.40 correspond to the former method. For

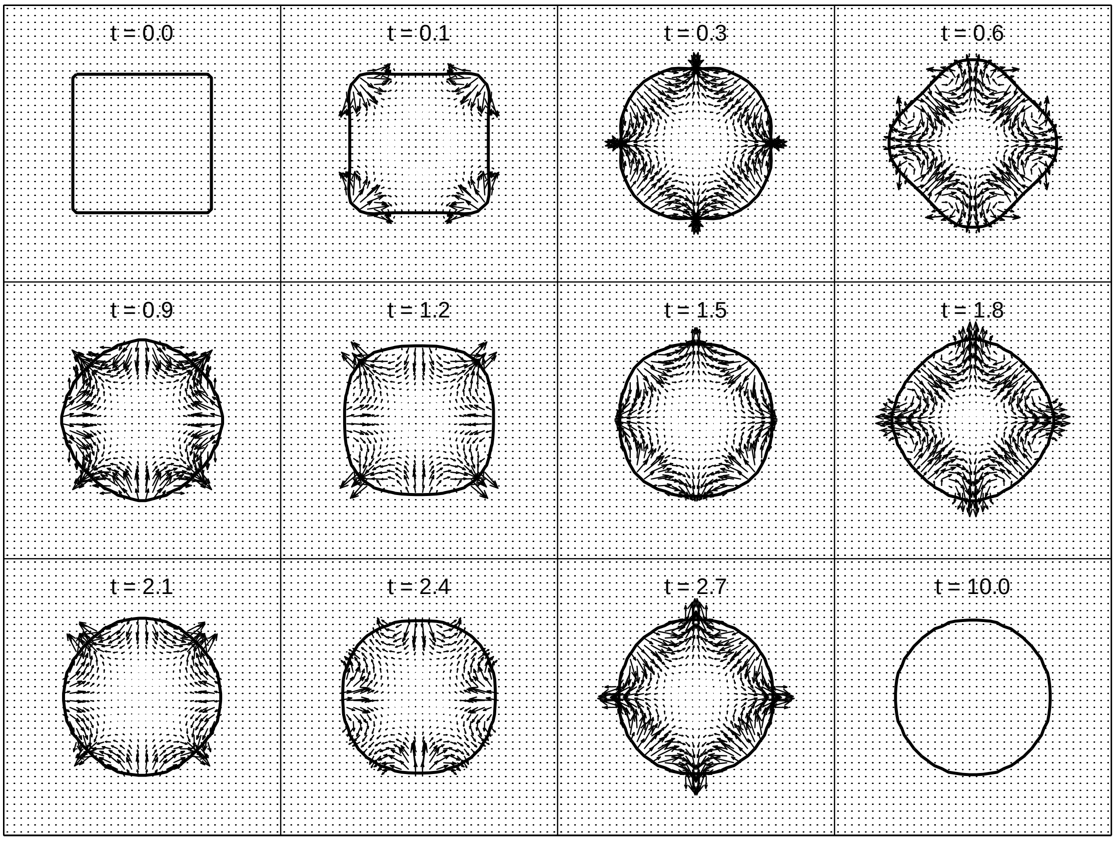


Figure 2.40: *Snapshots of the evolution of a square drop into a circular drop (computed with the method of Hirt and Nichols combined with a local height function). Shown are the shape of the free surface (solid lines) and the velocity field (arrows). In each snapshot the velocity is scaled with the largest velocity.*

this method the centre of mass of the liquid drop remains in the centre of the computational domain throughout the entire simulation. Small asymmetries in the VOF field and the velocity field (see for example the snapshot at time $t = 1.5$, where the relatively large velocity in the north has no counterpart in the east, south, or west) are damped. In the method of Youngs, the kinetic energy does not decrease to machine precision because the drop starts drifting away from the centre of the computational domain and, eventually, impacts with the solid boundary; see also the discussion for the static drop above.

Apart from the snapshots, also the diameter of the liquid drop along the horizontal centreline is studied. This is done in two and three dimensions on grids consisting of either 20 or 40 cells in every Cartesian direction. The results are shown in figure 2.41. Since the initial amount of fluid is known (and is constant in time because the local height function is used) and the shape of the free surface in the steady-state solution is either circular or spherical, the diameter of the liquid drop in the steady state can be compared to theory easily. This steady-state diameter is plotted in figure 2.41 as a dash-dotted, horizontal line. For the coarse-grid solutions the difference between the numerical and the theoretical steady-state diameter is 0.60% and 0.77% in two and three dimensions respectively. On the fine grid these errors equal 0.18% and 0.33% respectively.

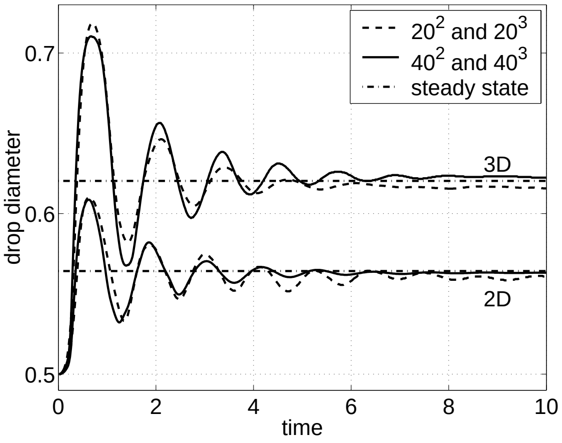


Figure 2.41: *Diameter of the liquid drop as a function of time in two and three dimensions for various grids. The dash-dotted lines indicate the theoretical drop diameters of the steady-state solution.*

2.5.3 Two-Dimensional Wall Adhesion in Square Cavity

In this section capillary flow in a (two-dimensional) square cavity is studied. Initially, the lower half of the cavity is filled with liquid. Surface tension is the only force present. A static contact angle θ is prescribed. The discrepancy between this contact angle and the initial angle between the free surface and the solid boundary drives the flow. Due to surface tension, the free surface will settle itself into an equilibrium configuration in which the mean curvature of the free surface is constant and the angle between the free surface and the solid boundary is equal to the prescribed contact angle; thus the shape of the free surface in the steady-state situation will be circular (see also [5, 55]).

Grid-Refinement Study

First, wall adhesion in a square container is simulated with $\theta = 45$ degrees. In figure 2.42 snapshots of the free surface are shown on various grids. For these simulations the Ohnesorge number was set to $Oh = 10^{-2}$. The snapshots are taken at time $t = 80$ when the fluid has reached a steady state (by this time the kinetic energy is close to machine precision). The bold line segments in figure 2.42 represent the reconstructed free surface and the thin, solid line shows the theoretical shape of the free surface. On the coarse grid of 20×20 cells a clear difference between simulation and theory is present; the grid is too coarse to predict the correct steady-state configuration of the free surface. On the finer grids of 40×40 and 80×80 cells the agreement is very good. A more quantitative comparison between simulation and theory is made in figure 2.43. In this figure, the liquid height at the wall and at the centreline of the cavity is plotted as a function of time and the theoretical liquid height in the steady-state solution is shown as a horizontal line. Again, the oscillation around the steady-state solution, which is typical for flows driven by surface tension, is present. From figure 2.43 it is clear that the error in the liquid height is larger at the solid wall than at the centreline of the cavity. The error in the steady-state liquid height at the wall is equal to 8.0%, 3.5%, and 1.5% for the

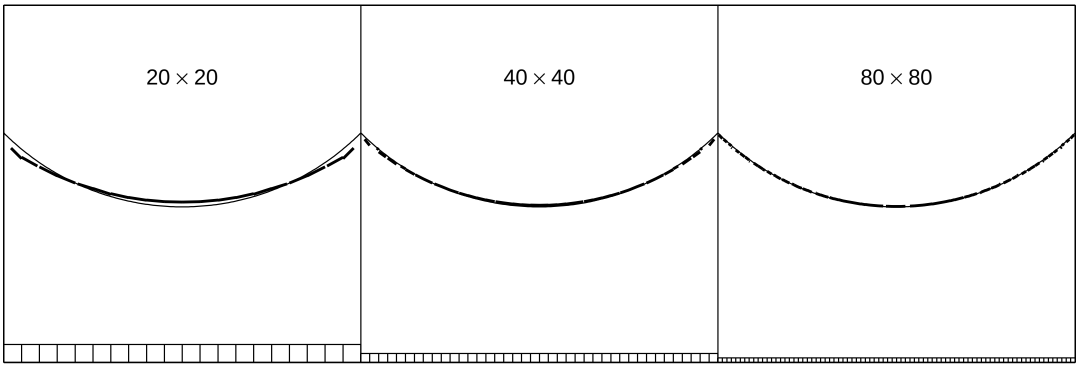


Figure 2.42: *Snapshots of the free surface in the steady state solution ($\theta = 45$) on grids of 20^2, 40^2, and 80^2 cells (a small part of the computational grid is shown in the bottom of the cavity). The bold line segments represent the reconstructed free surface. The solid thin line shows the theoretical steady state solution.*

coarse, medium, and fine grid respectively. Part of this error is due to the fact that, in the simulation, the liquid height at the wall is computed by adding the VOF values in the first (or last) column of the computational grid. This also explains the underprediction of the liquid height in the simulation. If this effect is taken into account, then the error in the liquid height on the coarse, medium, and fine grid is equal to 4.4%, 1.7%, and 0.54% respectively.

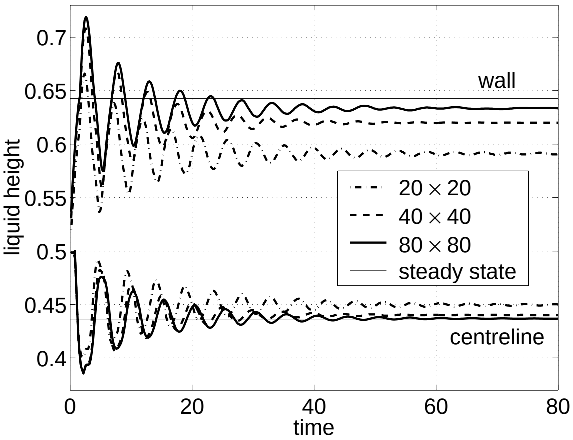


Figure 2.43: *Liquid height (for the simulation of wall adhesion in a square cavity ($\theta = 45$ and $Oh = 10^{-2}$)) at the wall and at the centreline as a function of time for grids consisting of 20^2, 40^2, and 80^2 cells.*

Variation of the Contact Angle

For checking the dependence of the free-surface shape on the contact angle, the same simulation as above is repeated with different contact angles. The steady-state solution of the free surface is shown in figure 2.44 for $\theta = 0$, 30, 60, 120, 150, and 180 degrees. Since the results on the medium grid gave adequately accurate results for $\theta = 45$, for all the simulations in this figure the grid consists of 40×40 cells. For post-processing reasons, the free surface is plot as a contour line of the VOF field at a value 0.5; hence the ripples in the free surface. Clearly, the method is capable of handling contact angles throughout the entire spectrum from 0 to 180 degrees; the agreement between the computed and the theoretical free-surface shape is very good.

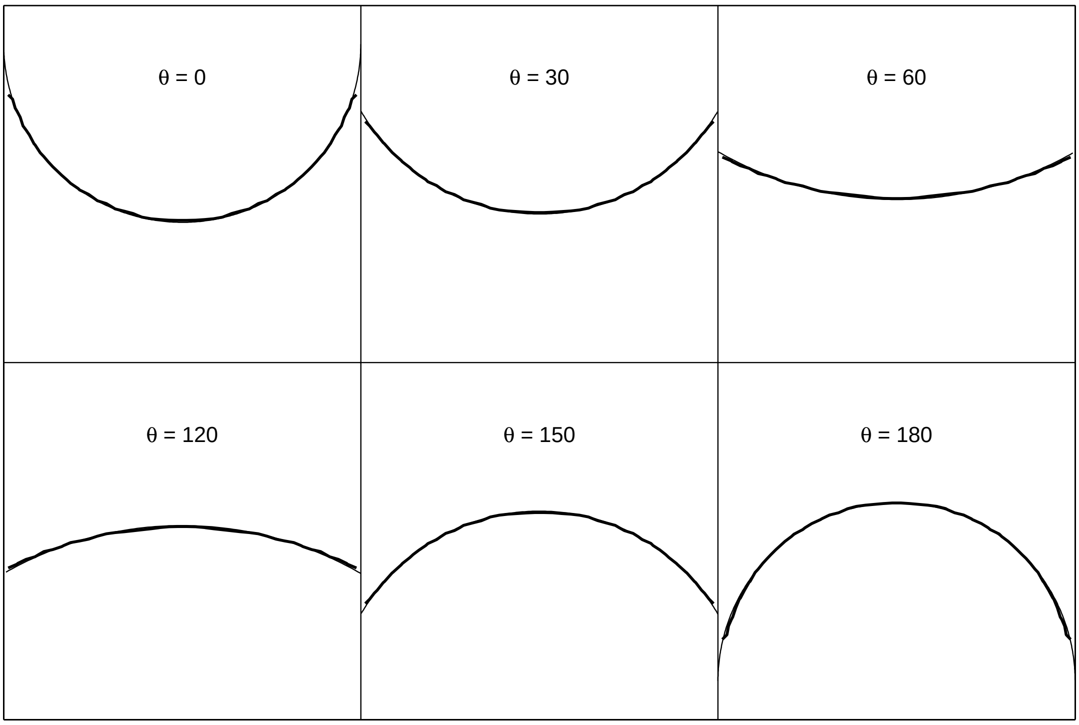


Figure 2.44: *Snapshots of the free surface in the steady-state solution for various contact angles. The bold lines show results of the simulation and the thin lines show theoretical results.*

Hirt and Nichols versus Youngs

Based on experiences in section 2.5.2, all the simulations in this section have been performed with the method of Hirt and Nichols in combination with a local height function. As a final comparison between the two methods for free-surface advection, the simulation in this section with contact angle $\theta = 30$ is repeated for both methods without the use of a local height function. Results of the simulation with the method of Hirt and Nichols are not presented here since these are almost identical to the ones presented before, where this method was combined with a local height function. This is due to the fact that, in simulations without massive change in topology of the free surface and moderate velocity gradients (like the simulations in this section), the original method of Hirt and Nichols gives similar results as the method extended with a local height function. The only difference is the slight increase or decrease of mass (for the simulations in this section in the order of 0.01%) due to small undershoots and overshoots in the VOF function.

For the original method of Youngs the results are quite dramatic. A few snapshots of the free surface for the simulation with contact angle $\theta = 30$ are shown in figure 2.45. Contrary to previous results in this section, the fluid reaches the top of the cavity (also when the value of the liquid viscosity is increased, fluid flows towards the top). Further, it is clear that symmetry is broken quite early in the simulation. It is concluded that the original method of Youngs, although it gives accurate results in simulations where the velocity field is prescribed (see section 2.5.1), gives worse results than the method of Hirt and Nichols in simulations with velocity fields that are computed from the Navier-Stokes equations.

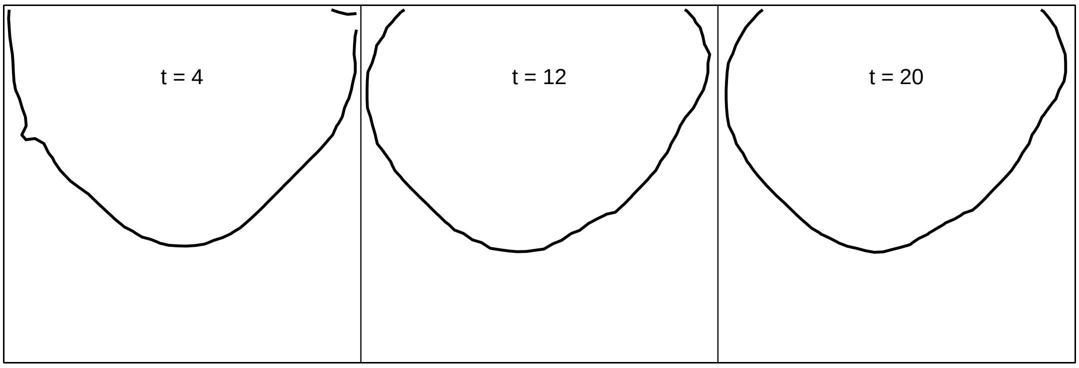


Figure 2.45: *Snapshots of the free surface for the method of Youngs without a local height function. The contact angle is equal to* $\theta = 30$ *and the number of grid cells is equal to* 40×40.

If a local height function is used in combination with Youngs' method, then the results improve a lot; the fluid stays coherent throughout the simulation and does not reach the top of the flow domain. However, contrary to results of the method of Hirt and Nichols, the results are not symmetrical. Moreover, in Youngs' method, the kinetic energy does not decrease to machine precision as is shown in figure 2.46.

Since the results of simulations, where the velocity field is solved from the Navier-Stokes equations, show that the method of Hirt and Nichols combined with a local height function for advecting the free surface gives the best results, all the simulations in the remaining part of this thesis are performed with this method.

2.5.4 Two-Dimensional Wall Adhesion in Circular Cavity

All the simulations in this chapter have been performed in simple geometries, *i.e.* in geometries of which the boundary coincides with Cartesian grid lines. These simple geometries were chosen to be able to focus the validation on the numerical models for surface tension, contact angle, and free-surface advection. In this section, for validating the treatment of cut cells, wall adhesion is studied in a circular cavity. The circle has a radius equal to unity. Initially, the lower half of the flow domain is filled with liquid. The contact angle is set to $\theta = 30$. Together with surface tension, this contact angle drives the flow.

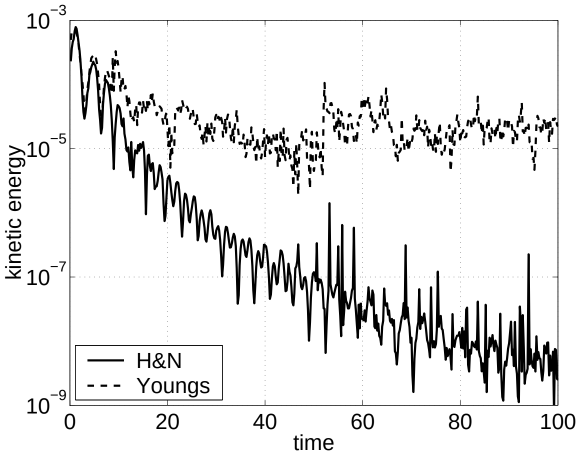


Figure 2.46: *Evolution of the kinetic energy in the wall-adhesion simulation ($\theta = 30$) for the method of Hirt and Nichols (H&N) and for Youngs' method, both in combination with a local height function.*

Grid-Refinement Study

In figure 2.47 snapshots (at time $t = 100$, when a steady state was reached) of the free surface are shown for grids consisting of 20×20, 40×40, and 80×80 cells. The Ohnesorge number for these simulations, based on the radius of the circular flow domain, was equal to $Oh = 10^{-2}$. In this figure, bold lines show results from the simulation, while thin lines correspond to the theoretical steady-state solution. As with the square cavity in figure 2.42, the grid consisting of 20×20 cells is too coarse to accurately resolve the liquid dynamics. The somewhat finer grid with 40×40 cells gives adequate results. The difference between theory and simulation is hardly visible on the finest grid with 80×80 cells.

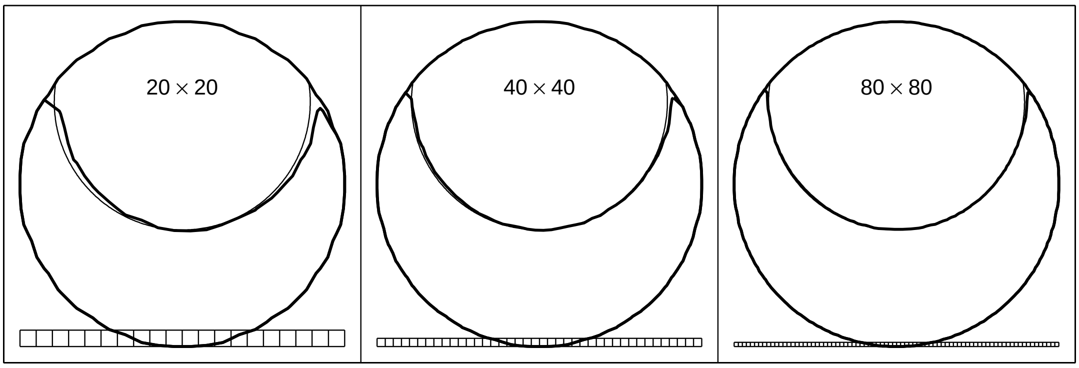


Figure 2.47: *Snapshots of the free surface in the steady-state solution ($\theta = 30$) on grids of 20^2, 40^2, and 80^2 cells (a small part of the computational grid is shown in the bottom of the flow domain). Bold lines show results from the simulation, thin lines show the theoretical steady-state solution.*

Staircase Geometry

To show the effect of taking into account cut cells instead of using a staircase discretisation of complex geometries, the simulation of wall adhesion in a circular cavity is repeated with a staircase approximation of the circular flow domain. This means that computational cells of which the centre lies outside the circular flow domain are B cells with a volume aperture $F^b = 0$. Volume apertures of interior cells are equal to $F^b = 1$. These cells are, depending on the presence of liquid, F, S, or E cells. For this simulation a grid with 80×80 cells was used. A snapshot of the free surface at time $t = 150$ is shown in the left of figure 2.48. Although the grid is quite fine, the free surface gets stuck in a corner of the staircase-shaped geometry. In order to show that this is not a temporary situation, the evolution of kinetic energy for this simulation is shown in the right of figure 2.48. Clearly, the configuration of the free surface at time $t = 150$ is almost identical to the steady state. This is confirmed by the circular shape of the free surface. Note that this solution is not in conflict with the prescription of a static contact angle of 30 degrees since in a non-smooth corner the contact angle can not be defined properly.

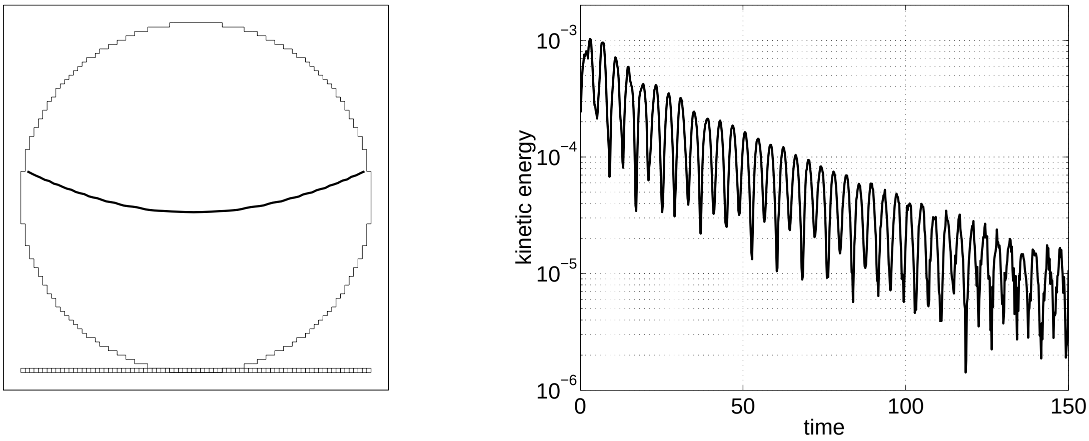


Figure 2.48: *Wall adhesion ($\theta = 30$) in a staircase approximation of the circular cavity. Left: Snapshot of the free surface at time $t = 150$. Right: Kinetic energy as a function of time.*

2.5.5 Three-Dimensional Wall Adhesion in Spherical Cavity

As a final example, wall adhesion in a (three-dimensional) spherical cavity is simulated. For this, a grid consisting of $40 \times 40 \times 40$ cells was used and the time step was set to $\delta t = 5 \times 10^{-4}$. Like in the previous simulations, the Ohnesorge number equaled $Oh = 10^{-2}$ and a contact angle $\theta = 30$ was prescribed. In figure 2.49 a three-dimensional snapshot of the steady-state solution is shown. To be able to see the free-surface shape, one quarter of the fluid is not drawn. Clearly, the free surface is spherical as theory predicts. However, it is not perfectly axisymmetric as is best seen at the contact line. The contact line is smooth in the direction of the Cartesian axes, while in the diagonal directions the free surface shows some ripples. This is due to the estimation of the contact line (see section 2.3.6), which is more difficult if the solid boundary is not aligned with a Cartesian direction.

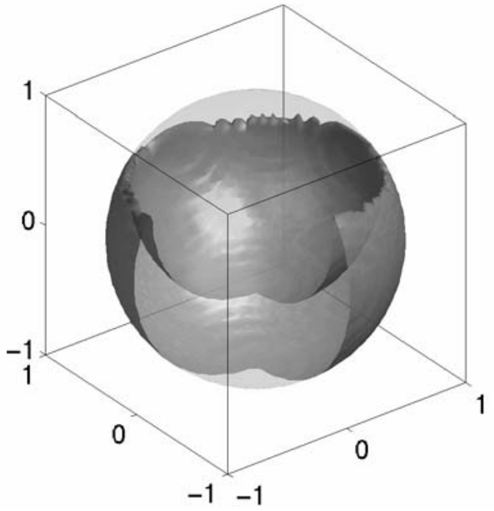


Figure 2.49: *Wall adhesion in a spherical cavity ($\theta = 30$). Snapshot of the steady-state solution. One quarter of the free surface has been removed to be able to see the interior free-surface shape.*

Chapter 3

Coupled Solid-Liquid Dynamics

In the previous chapter the motion of a liquid in a stationary container has been discussed; the motion of the container was not affected by the sloshing liquid. In this chapter the coupling of the solid-body motion and the motion of the liquid is studied.

3.1 Introduction

Usually, in practical applications, the motion of a container carrying liquid is not prescribed, but follows from an interaction with the liquid motion: the sloshing liquid induces a force and torque on the container, which, as a consequence, starts to accelerate. In return, this container motion influences the liquid motion, and so forth (see [15, 22, 64] for terrestrial examples of solid-liquid interaction).

The coupling of the solid-body dynamics and the liquid dynamics is the focus of this chapter. The coupling consists of two parts. The influence of the solid-body motion on the liquid dynamics is modelled as a virtual body force in the Navier-Stokes equations (section 3.2). The model for the solid-body dynamics is explained in section 3.3. The governing equations for the solid-body dynamics, influenced by the motion of the liquid, are studied in section 3.3.1. These equations are written in such a form that the discretisation results in a stable integration scheme for arbitrary liquid/solid mass ratios. The stability of the integration scheme is exemplified with a simple mass-spring model in section 3.3.2. The actual discretisation of the equations for the solid-body dynamics and the solution of the discretised equations is discussed in section 3.3.3 and section 3.3.4 respectively. In section 3.4 results are presented.

3.2 Liquid Dynamics

3.2.1 Virtual Body Force

In the first step towards the coupling of the solid-body dynamics and the liquid dynamics, it is explained how the motion of the liquid, influenced by the motion of the solid body in which the liquid is contained, is modelled. Hereto, the liquid velocity is considered in two reference frames: the velocity $\boldsymbol{v}$ of a liquid particle with respect to an inertial reference frame $\mathcal{O}'x'y'z'$ and the velocity $\boldsymbol{u}$ of the same liquid particle with respect to a moving reference frame $\mathcal{O}xyz$. The relation between $\boldsymbol{u}$ and $\boldsymbol{v}$ is given by

$$\frac{D\boldsymbol{v}}{Dt} = \dot{\boldsymbol{q}} + \dot{\boldsymbol{\omega}} \times \boldsymbol{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}) + \frac{D\boldsymbol{u}}{Dt} + 2\boldsymbol{\omega} \times \boldsymbol{u}, \tag{3.1}$$

where $\dot{\boldsymbol{q}} = d\boldsymbol{q}/dt + \boldsymbol{\omega} \times \boldsymbol{q}$ is the acceleration of the moving origin $\mathcal{O}$ with respect to the origin $\mathcal{O}'$ of the inertial reference frame, $\dot{\boldsymbol{\omega}} = d\boldsymbol{\omega}/dt + \boldsymbol{\omega} \times \boldsymbol{\omega} = d\boldsymbol{\omega}/dt$ is the angular acceleration of the moving reference frame, $\boldsymbol{\omega}$ is the angular velocity of the moving reference frame $\mathcal{O}xyz$, and $\boldsymbol{r}$ is the position of the liquid particle in $\mathcal{O}xyz$. The third and fifth term in the right-hand side of equation (3.1) represent the centrifugal and Coriolis acceleration respectively (see also [26]). Now, the Navier-Stokes equations read

$$\frac{D\boldsymbol{v}}{Dt} = -\frac{1}{\rho}\left(\nabla p - (\nabla \cdot \mu\nabla)\,\boldsymbol{u}\right) + \boldsymbol{F}, \tag{3.2}$$

or, equivalently, using equation (3.1)

$$\frac{D\boldsymbol{u}}{Dt} = -\frac{1}{\rho}\left(\nabla p - (\nabla \cdot \mu\nabla)\,\boldsymbol{u}\right) + \boldsymbol{F} + \boldsymbol{f}, \tag{3.3}$$

where

$$\boldsymbol{f} = -\dot{\boldsymbol{q}} - \dot{\boldsymbol{\omega}} \times \boldsymbol{r} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}) - 2\boldsymbol{\omega} \times \boldsymbol{u}. \tag{3.4}$$

Equation (3.3) has a form similar to equation (2.2), meaning that in the numerical model for the liquid dynamics only little change is required. Using Newton's third law, the extra term $\boldsymbol{f}$ in equation (3.3) can be seen as an acceleration due to a virtual body force: instead of actually moving the solid body in the numerical model, the fluid is subjected to an acceleration (equal in magnitude and opposite in sign) to account for the solid-body motion.

3.2.2 Example

For demonstrating the concept of a virtual body force for incorporating solid-body motion in the model for the liquid dynamics, an exemplifying simulation is studied in this section.

Consider a rectangular container with coordinates $[-10, 10] \times [-2, 2]$ in which a square of fluid with coordinates $[4, 6] \times [-1, 1]$ is placed. The fluid is not subjected to gravitational or capillary forces. The container is subjected to a sinusoidal motion; say that the horizontal component of the geometric centre at time t is located at $-5\cos(2\pi t) + 5$ (the vertical component is kept equal to zero). In this case, the fluid experiences no force (since it does not come in contact with the container) and therefore should stay at its initial position. However, in the numerical model not the solid body but the fluid is moved, whence the horizontal component of the centre of mass of the fluid should follow the path given by $5\cos(2\pi t)$ (with respect to the moving reference frame). The results in figure 3.1 confirm this. In this figure the path of the centre of mass of the fluid (with respect to the moving reference frame) is plotted as a function of time (dashed line). When the path of the container is added to this graph, the solid line appears, indicating that the fluid is at rest with respect to an inertial reference frame.

This simulation has been repeated in all Cartesian directions. Furthermore, similar simulations have been performed for validating the virtual body force in the case of

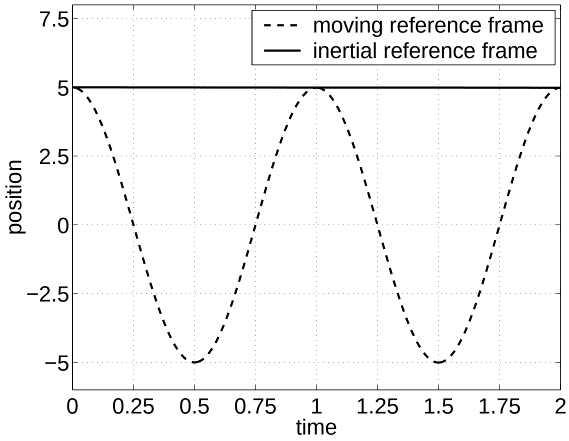


Figure 3.1: *Path of the centre of mass (horizontal component) of the fluid with respect to a moving reference frame and with respect to an inertial reference frame.*

angular motion. Results from these simulations are identical to figure 3.1, from which it is concluded that the concept of a virtual body force works properly and is correctly implemented (see also [78]).

3.3 Solid-Body Dynamics

3.3.1 Governing Equations

The model for the solid-body motion consists of an equation for linear momentum

$$m_s\dot{\boldsymbol{q}} + \dot{\boldsymbol{\omega}} \times m_s\overline{\boldsymbol{r}}_s + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times m_s\overline{\boldsymbol{r}}_s) = \boldsymbol{\mathcal{F}} + m_s\boldsymbol{F}, \tag{3.5a}$$

and an equation for angular momentum

$$m_s\overline{\boldsymbol{r}}_s \times \dot{\boldsymbol{q}} + \boldsymbol{I}_s\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}_s\boldsymbol{\omega} = \boldsymbol{\mathcal{T}} + m_s\overline{\boldsymbol{r}}_s \times \boldsymbol{F}. \tag{3.5b}$$

In these equations $\boldsymbol{q}$ and $\boldsymbol{\omega}$ are the unknown variables, representing the linear and angular velocity of the solid body respectively. The mass of the solid body is denoted by m_s. Further, $\boldsymbol{I}_s$ is the moment-of-inertia tensor and $\overline{\boldsymbol{r}}_s$ is the centre of mass of the solid body. Note that the latter two quantities are relative to the moving reference frame. The last terms in (3.5a) and (3.5b) represent the force and torque due to, for example, gravity. Finally, $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{T}}$ are respectively the force and torque that the fluid, via pressure (normal stress) and viscous effects (tangential stress), exerts on the boundary of the solid body, *i.e.*

$$\boldsymbol{\mathcal{F}} = \oint_{\partial V} (p\boldsymbol{I}_3 - \mu\nabla\boldsymbol{u}) \cdot \boldsymbol{n}\, dS,$$

$$\boldsymbol{\mathcal{T}} = \oint_{\partial V} (\boldsymbol{r} \times (p\boldsymbol{I}_3 - \mu\nabla\boldsymbol{u})) \cdot \boldsymbol{n}\, dS.$$

Here, $\boldsymbol{I}_3$ is the 3×3 identity matrix and $\boldsymbol{n}$ the outward-pointing normal on the boundary ∂V of the total volume V of the solid body. Using the divergence theorem, these integrals can be written as integrals over the volume V (see also [94]):

$$\begin{aligned}
\boldsymbol{\mathcal{F}} &= \int_V \nabla p - (\nabla \cdot \mu \nabla)\, \boldsymbol{u} \, dV = -\int_V \rho \left(\frac{D\boldsymbol{v}}{Dt} - \boldsymbol{F} \right) dV, \\
\boldsymbol{\mathcal{T}} &= \int_V \boldsymbol{r} \times \left(\nabla p - (\nabla \cdot \mu \nabla)\, \boldsymbol{u} \right) dV = -\int_V \rho \boldsymbol{r} \times \left(\frac{D\boldsymbol{v}}{Dt} - \boldsymbol{F} \right) dV.
\end{aligned}$$

In the last step the Navier-Stokes equations (3.2) were used. Note that the integration in the previous equations is not only over the liquid volume, but also over the void. In the void the density is set to zero, such that the void region does not contribute to the integral.

Direct discretisation of the system (3.5) would result in a method that is not stable for arbitrary liquid/solid mass ratios (this is demonstrated in section 3.4.1 and exemplified with a simple mass-spring model in section 3.3.2). Therefore, the system for the solid-body dynamics is rewritten first. Consider the first term on the right-hand side of equation (3.5a)

$$\boldsymbol{\mathcal{F}} = -\int_V \rho \left(\frac{D\boldsymbol{v}}{Dt} - \boldsymbol{F} \right) dV.$$

Using relation (3.1) for the liquid velocity $\boldsymbol{v}$ with respect to an inertial reference frame and the liquid velocity $\boldsymbol{u}$ with respect to a moving reference frame, the force $\boldsymbol{\mathcal{F}}$ is rewritten as

$$\boldsymbol{\mathcal{F}} = -\int_V \rho \left(\dot{\boldsymbol{q}} + \dot{\boldsymbol{\omega}} \times \boldsymbol{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}) + \frac{D\boldsymbol{u}}{Dt} + 2\boldsymbol{\omega} \times \boldsymbol{u} - \boldsymbol{F} \right) dV.$$

Since the linear velocity $\boldsymbol{q}$ and the angular velocity $\boldsymbol{\omega}$ of the solid body are constant over the volume V, this is equal to

$$\begin{aligned}
\boldsymbol{\mathcal{F}} = &- \dot{\boldsymbol{q}} \int_V \rho \, dV - \dot{\boldsymbol{\omega}} \times \int_V \rho \boldsymbol{r} \, dV - \boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times \int_V \rho \boldsymbol{r} \, dV \right) \\
&- \int_V \rho \left(\frac{D\boldsymbol{u}}{Dt} + 2\boldsymbol{\omega} \times \boldsymbol{u} - \boldsymbol{F} \right) dV.
\end{aligned}$$

In short, denoting the liquid mass with m_l and the centre of mass of the liquid with $\overline{\boldsymbol{r}}_l$ (with respect to the moving reference frame), this can be written as

$$\boldsymbol{\mathcal{F}} = -m_l \dot{\boldsymbol{q}} - \dot{\boldsymbol{\omega}} \times m_l \overline{\boldsymbol{r}}_l - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times m_l \overline{\boldsymbol{r}}_l) - \int_V \rho \left(\frac{D\boldsymbol{u}}{Dt} + 2\boldsymbol{\omega} \times \boldsymbol{u} - \boldsymbol{F} \right) dV.$$

Similarly, the torque $\boldsymbol{\mathcal{T}}$ on the right-hand side of equation (3.5b) can be rewritten, which results in

$$\boldsymbol{\mathcal{T}} = -m_l \overline{\boldsymbol{r}}_l \times \dot{\boldsymbol{q}} - \boldsymbol{I}_l \dot{\boldsymbol{\omega}} - \boldsymbol{\omega} \times \boldsymbol{I}_l \boldsymbol{\omega} - \int_V \rho \boldsymbol{r} \times \left(\frac{D\boldsymbol{u}}{Dt} + 2\boldsymbol{\omega} \times \boldsymbol{u} - \boldsymbol{F} \right) dV,$$

where $\boldsymbol{I}_l$ is the moment-of-inertia tensor of the liquid (with respect to the moving reference frame). Substituting these expressions for $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{T}}$ in the system (3.5) and rearranging terms gives an alternative form of the model for the solid-body dynamics, namely

$$m\dot{\boldsymbol{q}} + \dot{\boldsymbol{\omega}} \times m\overline{\boldsymbol{r}} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times m\overline{\boldsymbol{r}}) = -\int_V \rho \left(\frac{D\boldsymbol{u}}{Dt} + 2\boldsymbol{\omega} \times \boldsymbol{u} - \boldsymbol{F} \right) dV + m_s\boldsymbol{F}, \tag{3.8a}$$

$$m\overline{\boldsymbol{r}} \times \dot{\boldsymbol{q}} + \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} = -\int_V \rho\boldsymbol{r} \times \left(\frac{D\boldsymbol{u}}{Dt} + 2\boldsymbol{\omega} \times \boldsymbol{u} - \boldsymbol{F} \right) dV + m_s\overline{\boldsymbol{r}}_s \times \boldsymbol{F}. \tag{3.8b}$$

In these equations $m = m_s + m_l$ is the total mass, $\boldsymbol{I} = \boldsymbol{I}_s + \boldsymbol{I}_l$ the moment-of-inertia tensor of the coupled system, and $\overline{\boldsymbol{r}} = (m_s\overline{\boldsymbol{r}}_s + m_l\overline{\boldsymbol{r}}_l)/m$ the centre of mass of the coupled system. Note that the sloshing liquid makes the latter two quantities time dependent.

An important difference between the systems (3.5) and (3.8) is the distribution of the solid-body mass and the liquid mass over the left-hand side and right-hand side of the equations. In the former system the left-hand side contains the mass of the solid body, while on the other side the mass of the liquid, which is moving in the inertial reference frame, appears. Intuitively, one can see that solving this system iteratively, where the left-hand side and right-hand side are evaluated at the new and old time level respectively, can become unstable if the liquid mass is too large with respect to the mass of the solid body. In the latter system the total mass of the coupled system, *i.e.* the sum of the solid-body mass and the liquid mass, appears on the left-hand side. Now, the right-hand side contains the liquid mass in the moving reference frame. In this case, solving this system iteratively can be done in a stable manner since the liquid mass is always smaller than the total mass of the coupled system.

Before discussing the discretisation of (3.8), the stability of a mass-spring model, that exhibits similar characteristics as the model that has been discussed in this section, will be analysed.

3.3.2 Mass-Spring Model

Numerical Instability

Consider, as shown in figure 3.2, a system of two coupled masses m_s and m_l. For consistency these masses are referred to as the solid-body mass and the liquid mass respectively, although, in the present section, the liquid mass is a solid as well. The masses are connected by a spring, having a spring constant k and length ℓ. The system is partitioned

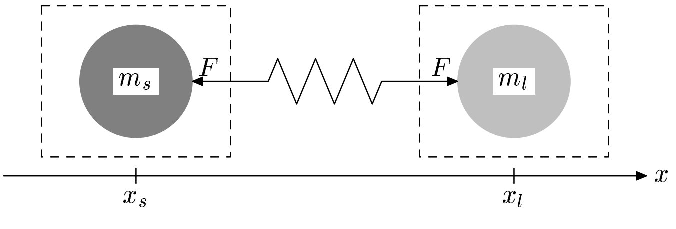


Figure 3.2: *Partitioning of mass-spring system (possibly unstable coupling).*

as indicated by the dashed rectangles, such that the solid-body mass is on one side and the liquid mass is on the other side. This partitioning corresponds to system (3.5). For the mechanical system in figure 3.2 the equations of motion read

$$m_s \ddot{x}_s = -F, \tag{3.9a}$$

$$m_l \ddot{x}_l = F, \tag{3.9b}$$

$$k\left(x_l - x_s - \ell\right) = -F. \tag{3.9c}$$

Although an analytical solution of this system is easily found, in this section the equations are solved iteratively for demonstrating the stability of the iteration process. Assuming that the solid body moves in reaction to the liquid forces, the left-hand side of equation (3.9a) is discretised at the new iteration level $n+1$, while the right-hand side is evaluated at the old level n. In the remaining two equations (3.9b) and (3.9c) all variables are evaluated at level $n+1$, which results in the following discrete version of system (3.9)

$$m_s \ddot{x}_s^{(n+1)} = -F^{(n)}, \tag{3.10a}$$

$$m_l \ddot{x}_l^{(n+1)} - F^{(n+1)} = 0, \tag{3.10b}$$

$$x_l^{(n+1)} + \frac{1}{k} F^{(n+1)} = x_s^{(n+1)} + \ell, \tag{3.10c}$$

where the iteration level is denoted with a superscript. From this system the variables x_s and x_l can be eliminated by differentiating (3.10c) twice with respect to time and then substituting equations (3.10a) and (3.10b) in this equation. After rearranging terms this gives

$$\frac{m_l}{k} \ddot{F}^{(n+1)} + F^{(n+1)} = -\frac{m_l}{m_s} F^{(n)}. \tag{3.11}$$

In the case of an infinitely stiff coupling, *i.e.* $k = \infty$, this iterative process is stable if and only if $m_l/m_s < 1$; the liquid mass must not exceed the mass of the solid body.

Obviously, in this simple model, if $m_l/m_s > 1$, the role of the liquid and solid body can be interchanged to make the iterative process stable. However, if the models for the solid-body dynamics and the liquid dynamics are very different, as is the case in the rest of this thesis, this solution is not an option.

Stable Numerical Coupling

Of course, the stability problems encountered above can be overcome by solving the system (3.9) simultaneously. However, as pointed out before, this would be very inconvenient since in practice the models for the solid-body dynamics and liquid dynamics are solved with very different algorithms. Therefore, inspired by the quasi-simultaneous method developed for aerodynamical flow [12, 77], the dynamics of the solid body and the liquid are partitioned in such a way that the numerical method is stable for arbitrary liquid/solid mass ratios. This is achieved by adding a term $m_l \ddot{x}_s^{(n+1)}$ to the left-hand side of equation (3.10a). The same term is added to the right-hand side of this equation, but here with iteration index (n). Thus, equation (3.10a) is changed to

$$\left(m_s + m_l\right) \ddot{x}_s^{(n+1)} = -F^{(n)} + m_l \ddot{x}_s^{(n)}. \tag{3.12}$$

This equation is combined with equations (3.10b) and (3.10c), which, after eliminating x_s and x_l, results in

$$\frac{m_l}{k}\ddot{F}^{(n+1)} + F^{(n+1)} = \frac{m_l}{m_s + m_l}\frac{m_l}{k}\ddot{F}^{(n)}. \tag{3.13}$$

Compared to equation (3.11) the right-hand side has changed. The important difference is the factor $m_l/(m_s + m_l)$, which is always less than one, on the right-hand side of equation (3.13) instead of a factor m_l/m_s in (3.11).

Graphically, the partitioning of the coupled system that is shown in equation (3.12) can be illustrated as in figure 3.3. The idea is that a "frozen" state of the liquid is treated simultaneously with the solid body (the left-hand side of equation (3.12) and the left-hand rectangle in figure 3.3). The difference between the "frozen" liquid behaviour and the real liquid dynamics, oscillations around the frozen state, appears on the right-hand side of equation (3.12) and in the right-hand rectangle in figure 3.3. This quasi-simultaneous method makes the coupling stable and corresponds to system (3.8); here, also, the solid-body mass and the liquid mass are treated simultaneously as is reflected on the left-hand side of (3.8). In this system the "frozen" state is the position of the liquid with respect to the inertial reference frame; oscillations around this state correspond to the liquid dynamics with respect to the moving reference frame (as described by the velocity $\boldsymbol{u}$).

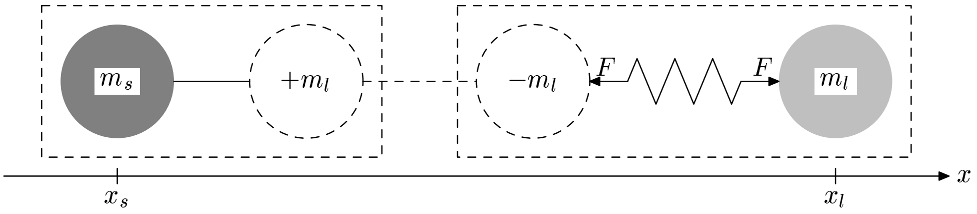


Figure 3.3: *Partitioning of mass-spring system (stable coupling).*

Stability Analysis

Apart from an intuitive approach, the stability of the two partitioning methods in this section can also be analysed mathematically. Hereto, equations (3.10a) and (3.12) are unified in a single equation

$$(m_s + \gamma m_l)\,\ddot{x}_s^{(n+1)} = -F^{(n)} + \gamma m_l \ddot{x}^{(n)},$$

where γ is a parameter indicating the type of partitioning; $\gamma = 0$ corresponds to a possibly unstable partitioning and $\gamma = 1$ to the stable partitioning. This equation is combined with (3.10b) and (3.10c) as before, which gives

$$\frac{m_l}{k}\ddot{F}^{(n+1)} + F^{(n+1)} = -\left(1-\gamma\right)\beta F^{(n)} + \gamma\beta\frac{m_l}{k}\ddot{F}^{(n)},$$

where a parameter $\beta = \beta\left(\gamma\right) = m_l/\left(m_s + \gamma m_l\right)$ has been introduced. After substitution of a discretised version of the second-order derivative with time step δt, *e.g.*

$$\ddot{F}^{(n)} = \frac{F^{(n)} - 2F^{(n-1)} + F^{(n-2)}}{\delta t^2},$$

and rearranging terms this results in

$$(\alpha+1)\,F^{(n+1)}+(\alpha\beta\,(1-\gamma)-\beta\gamma-2)\,F^{(n)}+(2\beta\gamma+1)\,F^{(n-1)}-\beta\gamma F^{(n-2)}=0,$$

where $\alpha = k\,\delta t^2/m_l$. The roots λ of this third-order difference equation satisfy

$$\alpha\lambda^3+\alpha\beta\,(1-\gamma)\,\lambda^2+(\lambda-\beta\gamma)\,(\lambda-1)^2=0.$$

For $\alpha = 0$ the roots are equal to $\lambda_{1,2} = 1$ and $\lambda_3 = \beta\gamma$. The latter root is always less than one since either $\gamma = 0$, whence $\lambda_3 = 0$, or $\gamma = 1$, whence $\lambda_3 = \beta = m_l/\,(m_s + m_l) < 1$. Hence, both the partitioning methods are stable for $\alpha = 0$. In the case $\alpha \to \infty$, two of the roots vanish, the other root equals $\lambda_3 = -\beta\,(1-\gamma)$. This root vanishes as well if $\gamma = 1$, making the second partitioning method stable for $\alpha \to \infty$. However, if $\gamma = 0$, then $\lambda_3 = -\beta = -m_l/m_s$. Thus, the first partitioning method, for $\alpha \to \infty$, is unstable if $m_l > m_s$; a result that has been noticed before.

3.3.3 Discretised Equations

Based on the stability of the different partitionings of the mass-spring model in the previous section, system (3.8) (rather than (3.5)) is the preferred model for the solid-body dynamics. Indeed, after discretisation, system (3.5) (where the left-hand side is evaluated at the new time level and the right-hand side at the old time level) corresponds to equations (3.10a) and (3.10b) in the sense that the solid-body mass appears on the left and the liquid mass on the right of the equality. On the other hand, in system (3.8), the total mass of the coupled system appears on the left-hand side, which corresponds to equation (3.12). In section 3.4.1 more details on the stability of systems (3.5) and (3.8) can be found.

In this section the discretisation of system (3.8) is discussed in more detail. The temporal discretisation of the equations for linear momentum (3.8a) and angular momentum (3.8b) is straightforward. Both the linear acceleration $d\boldsymbol{q}/dt$ and the angular acceleration $d\boldsymbol{\omega}/dt$ are discretised at the new time level (indicated by a superscript $n+1$). The linear and angular velocity are discretised at the old time level (indicated by a superscript n). Since the model for the solid-body dynamics is applied after the model for the liquid dynamics has been completed (in the same time step), all the quantities from the liquid model carry a superscript $n+1$. This results in

$$m\frac{d\boldsymbol{q}}{dt}^{n+1}+\frac{d\boldsymbol{\omega}}{dt}^{n+1}\times m\overline{\boldsymbol{r}}^{n+1}=\boldsymbol{\mathcal{L}}^n, \tag{3.14a}$$

$$m\overline{\boldsymbol{r}}^{n+1}\times\frac{d\boldsymbol{q}}{dt}^{n+1}+\boldsymbol{I}^{n+1}\frac{d\boldsymbol{\omega}}{dt}^{n+1}=\boldsymbol{\mathcal{A}}^n, \tag{3.14b}$$

where

$$\boldsymbol{\mathcal{L}}^n=-m\boldsymbol{\omega}^n\times\boldsymbol{q}^n-\boldsymbol{\omega}^n\times\left(\boldsymbol{\omega}^n\times m\overline{\boldsymbol{r}}^{n+1}\right)-\int\limits_V\rho\boldsymbol{\mathcal{U}}^{n+1}\,dV+m_s\boldsymbol{F}^{n+1}, \tag{3.15a}$$

$$\boldsymbol{\mathcal{A}}^n=-m\overline{\boldsymbol{r}}^{n+1}\times(\boldsymbol{\omega}^n\times\boldsymbol{q}^n)-\boldsymbol{\omega}^n\times\boldsymbol{I}^{n+1}\boldsymbol{\omega}^n-\int\limits_V\rho\boldsymbol{r}^{n+1}\times\boldsymbol{\mathcal{U}}^{n+1}\,dV+m_s\overline{\boldsymbol{r}}_s\times\boldsymbol{F}^{n+1}. \tag{3.15b}$$

In these equations $\boldsymbol{\mathcal{U}}^{n+1} = D\boldsymbol{u}^{n+1}/Dt + 2\boldsymbol{\omega}^n \times \boldsymbol{u}^{n+1} - \boldsymbol{F}^{n+1}$. Note that the volume V is the volume of the solid body (recall that the volume integral was deduced, via the divergence theorem, from a boundary integral over the boundary of the solid body). Hence, this volume V does not carry a superscript since it does not depend on time. However, in that part of the solid body where liquid is absent, the integrand vanishes because the density is taken to be equal to zero in the void region.

The implementation of the non-integral terms in (3.14) and (3.15) is trivial. The implementation of the remaining terms, although being far from difficult, requires some attention. All these terms are of the form $\int_V \rho\varphi^n \, dV$ and are discretised as

$$\int\limits_V \rho\varphi^n \, dV \doteq \sum_{i,j,k} \rho\varphi^n_{i,j,k} F^s_{i,j,k} \delta x_i \delta y_j \delta z_k, \tag{3.16}$$

where the summation is over all the computational cells, φ is some scalar variable (*e.g.* Du/Dt), $F^s_{i,j,k}$ is the volume-of-fluid function in cell (i, j, k), and $\delta x_i \delta y_j \delta z_k$ is the volume of this cell. The variable φ is evaluated in cell centres. This means that staggered velocities have to be averaged before they can be evaluated. Note that also the centre of mass $\overline{\boldsymbol{r}}_l$ and the moment-of-inertia tensor $\boldsymbol{I}_l$ of the liquid are integral terms since

$$m_l \overline{\boldsymbol{r}}_l = \int\limits_V \rho \boldsymbol{r} \, dV \quad \text{and} \quad \boldsymbol{I}_l \boldsymbol{\psi} = \int\limits_V \rho \boldsymbol{r} \times (\boldsymbol{\psi} \times \boldsymbol{r}) \, dV,$$

where $\boldsymbol{\psi}$ is some vector (*e.g.* $d\boldsymbol{\omega}/dt$) the moment-of-inertia tensor operates on. Therefore, these quantities are computed using (3.16) as well.

3.3.4 Solution Method

In matrix form the 6 × 6 linear system (3.14) for the unknown vectors $(d\boldsymbol{q}/dt)^{n+1}$ and $(d\boldsymbol{\omega}/dt)^{n+1}$ reads

$$\begin{pmatrix} m & 0 & 0 & 0 & m\overline{r}_z & -m\overline{r}_y \\ 0 & m & 0 & -m\overline{r}_z & 0 & m\overline{r}_x \\ 0 & 0 & m & m\overline{r}_y & -m\overline{r}_x & 0 \\ 0 & -m\overline{r}_z & m\overline{r}_y & I_{xx} & -I_{xy} & -I_{xz} \\ m\overline{r}_z & 0 & -m\overline{r}_x & -I_{xy} & I_{yy} & -I_{yz} \\ -m\overline{r}_y & m\overline{r}_x & 0 & -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix} \begin{pmatrix} dq_x/dt \\ dq_y/dt \\ dq_z/dt \\ d\omega_x/dt \\ d\omega_y/dt \\ d\omega_z/dt \end{pmatrix} = \begin{pmatrix} \mathcal{L}_x \\ \mathcal{L}_y \\ \mathcal{L}_z \\ \mathcal{A}_x \\ \mathcal{A}_y \\ \mathcal{A}_z \end{pmatrix}. \tag{3.17}$$

For presentational reasons the superscripts have been dropped (it will be clear that the matrix and right-hand side are evaluated at time level n, while the unknown vector contains quantities at time level $n + 1$). Further, the components of a vector $\boldsymbol{\psi}$ are denoted with subscripts x, y, and z, *i.e.* $\boldsymbol{\psi} = (\psi_x, \psi_y, \psi_z)^T$, and the moment-of-inertia tensor has components

$$\boldsymbol{I} = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix}.$$

By using Gaussian elimination, the lower-left 3×3 block of the matrix in (3.17) is eliminated, whence a decoupled system for $d\boldsymbol{\omega}/dt$ arises, namely

$$\left(\begin{array}{ccc|c} I_{xx} - m\overline{r}_y^2 - m\overline{r}_z^2 & -I_{xy} + m\overline{r}_x\overline{r}_y & -I_{xz} + m\overline{r}_x\overline{r}_z & \mathcal{A}_x + \overline{r}_z\mathcal{L}_y - \overline{r}_y\mathcal{L}_z \\ -I_{xy} + m\overline{r}_x\overline{r}_y & I_{yy} - m\overline{r}_x^2 - m\overline{r}_z^2 & -I_{yz} + m\overline{r}_y\overline{r}_z & \mathcal{A}_y - \overline{r}_z\mathcal{L}_x + \overline{r}_x\mathcal{L}_z \\ -I_{xz} + m\overline{r}_x\overline{r}_z & -I_{yz} + m\overline{r}_y\overline{r}_z & I_{zz} - m\overline{r}_x^2 - m\overline{r}_y^2 & \mathcal{A}_z + \overline{r}_y\mathcal{L}_x - \overline{r}_x\mathcal{L}_y \end{array}\right). \tag{3.18}$$

Note that during this process no divisions (in particular through zero) take place. Using Steiner's theorem, it follows that the 3×3 matrix in (3.18) is the moment-of-inertia tensor relative to the centre of mass of the coupled system. The 3×3 linear system for $d\boldsymbol{\omega}/dt$ is solved with Gaussian elimination followed by backward substitution. Once the solution for $d\boldsymbol{\omega}/dt$ is known, the solution for $d\boldsymbol{q}/dt$ can be found using backward substitution in (3.17).

In the model for the solid-body dynamics and also in the Navier-Stokes equations (3.3) the linear and angular velocity of the solid body appears. This means that $d\boldsymbol{q}^{n+1}/dt$ and $d\boldsymbol{\omega}^{n+1}/dt$ have to be integrated in time, which is done using a fourth-order Runge-Kutta method. Hereto, system (3.17) is written as

$$\frac{d\boldsymbol{y}}{dt} = \boldsymbol{g}\left(t, \boldsymbol{y}\right), \tag{3.19}$$

where $\boldsymbol{y} = \left(\boldsymbol{q}, \boldsymbol{\omega}\right)^T$ and $\boldsymbol{g} = \boldsymbol{M}^{-1}\left(\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{A}}\right)^T$. The matrix $\boldsymbol{M}$ is an abbreviation for the matrix that appears in (3.17). Equation (3.19) is integrated in time from level n to level $n+1$ by

$$\boldsymbol{y}^{n+1} = \boldsymbol{y}^n + \frac{1}{6}\delta t\left(\boldsymbol{S}_1 + 2\boldsymbol{S}_2 + 2\boldsymbol{S}_3 + \boldsymbol{S}_4\right),$$

where the four stages $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_4$ are defined as

$$\begin{aligned} \boldsymbol{S}_1 &= \boldsymbol{g}\left(t^n, \boldsymbol{y}^n\right), \\ \boldsymbol{S}_2 &= \boldsymbol{g}\left(t^n + \frac{1}{2}\delta t, \boldsymbol{y}^n + \frac{1}{2}\delta t\boldsymbol{S}_1\right), \\ \boldsymbol{S}_3 &= \boldsymbol{g}\left(t^n + \frac{1}{2}\delta t, \boldsymbol{y}^n + \frac{1}{2}\delta t\boldsymbol{S}_2\right), \\ \boldsymbol{S}_4 &= \boldsymbol{g}\left(t^n + \delta t, \boldsymbol{y}^n + \delta t\boldsymbol{S}_3\right). \end{aligned}$$

Note that the explicit time dependence in the right-hand side of (3.19) (*i.e.* the first component of the function $\boldsymbol{g}$) is entirely due to the liquid model. Since the models for the liquid dynamics and the solid-body dynamics are treated separately, it is most unconvenient to evaluate the position and velocity of the liquid at the intermediate time level $n + \frac{1}{2}$. Moreover, the quantities of the liquid model are already known at time level $n+1$. Hence, in the Runge-Kutta stages $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_4$, the first component of $\boldsymbol{g}$ is frozen at time level $n+1$.

In the simulations that have been performed with the method presented in this chapter, the solid-body dynamics turns out to be more sensitive to small disturbances than the model for the liquid dynamics, hence requiring either a smaller time step or a more accurate time-integration method. Since the latter option has been implemented, in practice, it is possible to set the time step for the solid-body model equal to the one used in the liquid model. Should this not be sufficient, then it is possible to take multiple Runge-Kutta time steps during one single time step in the liquid model.

Now that $d\boldsymbol{q}/dt$, $d\boldsymbol{\omega}/dt$, $\boldsymbol{q}$, and $\boldsymbol{\omega}$ are known at the new time level, one computational cycle has been finished.

3.4 Results

3.4.1 Free Fall

Stable Method

As a first example, the free fall of a rectangular tank, partially filled with liquid, is simulated. With this simulation some basic validation is performed. Consider a hollow, rectangular tank with dimensions $1 \times 2 \times 4$. The mass of the tank is set to $m_s = 12$. From these parameters the moment-of-inertia tensor of the tank, relative to the centre of mass of the tank, can be computed, namely $\boldsymbol{I}_s = \operatorname{diag}(36, 25, 9)$. The lower half (in z-direction) of the tank is filled with liquid having a density of $\rho = 6$. Hence, the liquid mass is equal to $m_l = 24$ and the total mass equals $m = 36$. The acceleration $\boldsymbol{F}$ due to gravity is set to $(0, 0, -10)^T$.

During the first 0.05 seconds of the simulation (on a computational grid of $10 \times 20 \times 40$ cells) the tank is kept at its initial position, whence the pressure settles itself at hydrostatic values. After 0.05 seconds, the tank, together with the liquid, is released. This, of course, causes the coupled system to fall in the negative z-direction with an acceleration equal to gravitational acceleration. Also, the pressure becomes equal to the ambient pressure $p_0 = 0$ everywhere in the fluid since the coupled system is in a free fall. These results are confirmed in figure 3.4. On the left in this figure, the pressure is plotted at equidistant locations along the centreline in z-direction. At the bottom of the tank the pressure is equal to $p = 6 \cdot 10 \cdot 2 = 120$. Taking into account that the centre of the first computational

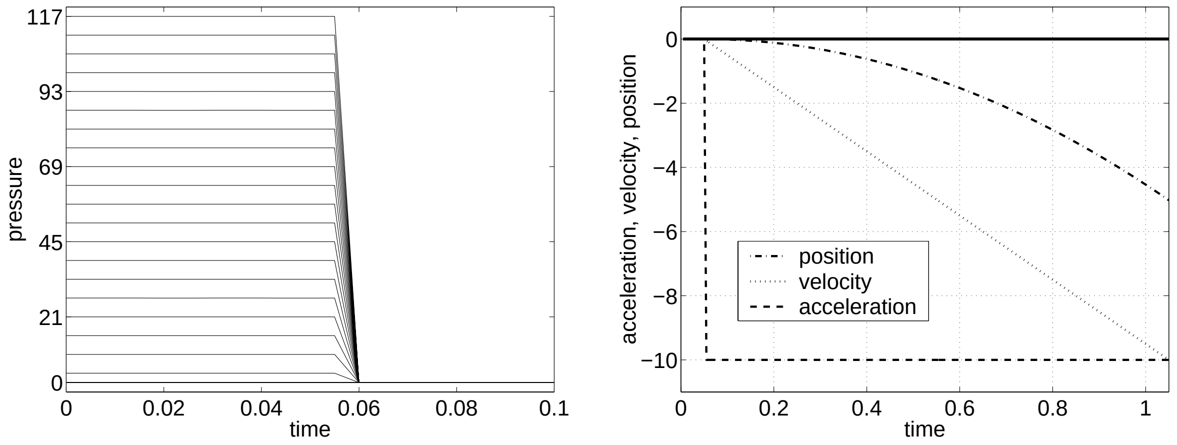


Figure 3.4: *Left: Pressure at equidistant locations along the z-centreline of the tank as a function of time. Right: Acceleration (dashed), velocity (dotted), and position (dash-dotted) in z-direction of the coupled system as a function of time. The solid line represents the same quantities in x- and y-direction.*

cell lies half a mesh size (which is $\delta z/2 = 0.05$) above the bottom of the tank, the maximum value of $p = 117$ in figure 3.4 can be explained. The same reasoning applies to the centre of the last cell below the free surface, where the pressure is equal to $p = 3$ (at the free surface the pressure has a value of $p = p_0 = 0$). Since the tank is kept at its initial position until $t = 0.05$, the pressure in the tank does not change in this interval. Also, one time step after $t = 0.05$ (the time step is set to $\delta t = 0.005$), the pressure has not changed since, in the liquid model, the virtual body force (3.4) is discretised at the old time level at which time the tank was not moving. In the next time step, at $t = 0.06$, the fluid "feels" that the tank is falling with an acceleration of $d\boldsymbol{q}/dt = (0, 0, -10)^T$, whence the pressure in the fluid becomes zero everywhere. On the right in figure 3.4 the motion of the tank is plotted. Most of the quantities, namely the angular acceleration, velocity, and position in all three spatial directions and the linear acceleration, velocity, and position in x- and y-direction are equal to zero during the entire simulation. The linear acceleration in z-direction changes from 0 to -10 at $t = 0.05$ as expected. Also the linear velocity and position in z-direction, showing linear and quadratic behaviour, are plotted. Note that, for presentational reasons, the initial position of the tank is set to zero in this figure.

Of course, the results from this simulation should be independent of the choice for the moving reference frame. To check this, the previous simulation is performed for two different cases. In the first case, the origin of the moving reference frame is equal to the centre of mass of the coupled system. In the second case, the origin is chosen such that the centre of mass of the tank is located at $\overline{\boldsymbol{r}}_s = (1, 2, 4)^T$. In the first case, system (3.17) is diagonal since $\overline{\boldsymbol{r}} = (0, 0, 0)^T$ and the moment-of-inertia tensor of the coupled system is diagonal (the coordinate system is aligned with the principal axes of the coupled system). Hence, in this case, system (3.18) is equal to the lower-right block of (3.17). In the second case, $\overline{\boldsymbol{r}} = (1, 2, 4/3)^T$ and $\boldsymbol{I}$ is not diagonal. Hence, system (3.17) is not diagonal and Gaussian elimination is needed to achieve a diagonal system (3.18) (again, the coordinate system is aligned with the principal axes). The 3×3 systems for $d\boldsymbol{\omega}/dt$ for these cases, taken from the simulation and rounded to two decimal places, are

$$\left(\begin{array}{ccc|c} 59.96 & 2.76E-08 & 9.59E-07 & -2.74E-04 \\ 2.76E-08 & 42.96 & -7.88E-07 & 4.10E-06 \\ 9.59E-07 & -7.88E-07 & 18.96 & -5.64E-05 \end{array}\right)$$

and

$$\left(\begin{array}{ccc|c} 59.96 & 9.77E-04 & 7.63E-04 & 2.62E-03 \\ 9.77E-04 & 42.96 & 1.89E-03 & -7.63E-04 \\ 7.63E-04 & 1.89E-03 & 18.96 & -5.64E-05 \end{array}\right).$$

respectively. Analytically, in both cases, system (3.18) equals

$$\left(\begin{array}{ccc|c} 60 & 0 & 0 & 0 \\ 0 & 43 & 0 & 0 \\ 0 & 0 & 19 & 0 \end{array}\right).$$

The numerical results agree closely with theory; minor differences are due to the numerical integration method (3.16) for computing the centre of mass and the moment-of-inertia tensor of the liquid. This simulation has been repeated for liquid/solid mass ratios ranging from 0.1 to 100. For all mass ratios the method gave stable results identical to figure 3.4 (only the value of the pressure in the fluid differed because of the change in the density of the fluid).

Unstable Method

The previous simulation has been performed with the possibly unstable method (3.5) as well. For this method the acceleration of the coupled system with a liquid/solid mass ratio of 1.2 is shown in figure 3.5. If the liquid/solid mass ratio is further decreased to one, then the unstable method gives the same results as the stable method. This can be explained by analysing equations (3.5a) and (3.8a). In the absence of angular motion these equations reduce to

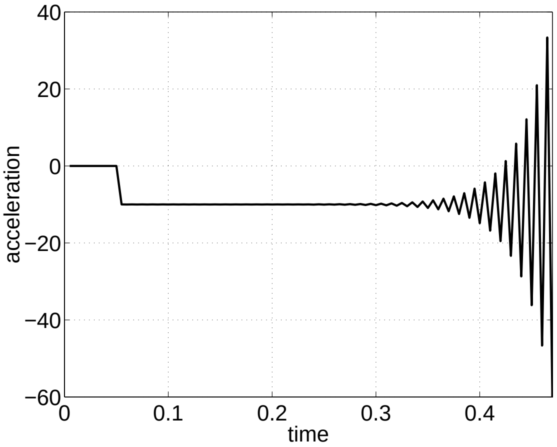


Figure 3.5: *Acceleration in z-direction of the coupled system as a function of time as predicted by the unstable method.*

$$m_s \frac{d\boldsymbol{q}}{dt} = -\int\limits_V \rho \left(\frac{D\boldsymbol{v}}{Dt} - \boldsymbol{F} \right) dV + m_s \boldsymbol{F}$$

and

$$m \frac{d\boldsymbol{q}}{dt} = -\int\limits_V \rho \left(\frac{D\boldsymbol{u}}{Dt} - \boldsymbol{F} \right) dV + m_s \boldsymbol{F},$$

respectively, where, using equation (3.1), the relation between $\boldsymbol{u}$ and $\boldsymbol{v}$ is given by $D\boldsymbol{v}/Dt = d\boldsymbol{q}/dt + D\boldsymbol{u}/Dt$. These equations are discretised by evaluating the left-hand side at the new time level $n+1$ and the right-hand side at the old time level n, *i.e.*

$$m_s \frac{d\boldsymbol{q}}{dt}^{n+1} = -\int\limits_V \rho \left(\frac{D\boldsymbol{v}^n}{Dt} - \boldsymbol{F} \right) dV + m_s \boldsymbol{F}$$

and

$$m \frac{d\boldsymbol{q}}{dt}^{n+1} = -\int\limits_V \rho \left(\frac{D\boldsymbol{u}^n}{Dt} - \boldsymbol{F} \right) dV + m_s \boldsymbol{F}.$$

If both the liquid velocity $\boldsymbol{u}$ with respect to the moving reference frame and the pressure p vanish throughout the liquid (as is the case in this section), then these equations simplify to

$$m_s \frac{d\boldsymbol{q}}{dt}^{n+1} = -m_l \frac{d\boldsymbol{q}}{dt}^{n} + m_l \boldsymbol{F} + m_s \boldsymbol{F} \quad \text{and} \quad m \frac{d\boldsymbol{q}}{dt}^{n+1} = m_l \boldsymbol{F} + m_s \boldsymbol{F},$$

which is rewritten as

$$\frac{d\boldsymbol{q}}{dt}^{n+1} = -\frac{m_l}{m_s} \left(\frac{d\boldsymbol{q}}{dt}^{n} - \boldsymbol{F} \right) + \boldsymbol{F} \quad \text{and} \quad \frac{d\boldsymbol{q}}{dt}^{n+1} = \boldsymbol{F}.$$

Clearly, the solution for both formulations is $d\boldsymbol{q}/dt \equiv \boldsymbol{F}$. However, small disturbances in the solution for $d\boldsymbol{q}/dt$ at time level n are in the first formulation amplified with a factor m_l/m_s (this factor is visible in figure 3.5, where $m_l/m_s = 1.2$), which becomes unstable if the liquid/solid mass ratio exceeds unity in correspondence with the numerical results (see also the analysis of the mass-spring model in section 3.3.2).

3.4.2 Empty Tank

Another method for validating the model for the solid-body dynamics is by studying the motion of a (rectangular) tank that is not filled with liquid. If the moving origin coincides with the centre of mass of the tank, *i.e.* $\overline{\boldsymbol{r}}_s = (0,0,0)^T$, and the acceleration $\boldsymbol{F}$ is set to zero, then system (3.8) simplifies to

$$\begin{aligned} m_s \frac{d\boldsymbol{q}}{dt} &= -m\boldsymbol{\omega} \times \boldsymbol{q}, \\ \boldsymbol{I}_s \frac{d\boldsymbol{\omega}}{dt} &= -\boldsymbol{\omega} \times \boldsymbol{I}_s \boldsymbol{\omega}. \end{aligned}$$

Note that, because of the absence of liquid, m and $\boldsymbol{I}$ have been replaced by m_s and $\boldsymbol{I}_s$ respectively. The initial conditions are chosen as $\boldsymbol{q}(0) = (0,0,0)^T$ and $\boldsymbol{\omega}(0) = \left(\omega_x^0, \omega_y^0, \omega_z^0\right)^T$. The former initial condition implies that $\boldsymbol{q}$ is identically equal to zero. Since the moving origin was chosen to coincide with the centre of mass of the tank, the moment-of-inertia tensor of the tank is diagonal, say $\boldsymbol{I}_s = \text{diag}\left(I_{xx}, I_{yy}, I_{zz}\right)$. Hence, the following three equations remain

$$\frac{d\omega_x}{dt} = \frac{I_{yy} - I_{zz}}{I_{xx}} \omega_y \omega_z, \quad \frac{d\omega_y}{dt} = \frac{I_{zz} - I_{xx}}{I_{yy}} \omega_x \omega_z, \quad \frac{d\omega_z}{dt} = \frac{I_{xx} - I_{yy}}{I_{zz}} \omega_x \omega_y. \tag{3.21}$$

Harmonic Solutions

Obviously, the tank keeps rotating with the initial angular velocity if two components of $\boldsymbol{\omega}(0)$ are set to zero. Hence, the initial angular velocity is set to $\boldsymbol{\omega}(0) = (2,0,1)^T$. Further, the moment-of-inertia tensor (relative to the centre of mass) is set to $\boldsymbol{I}_s = \text{diag}(1, 1, \pi/2 + 1)$ (note that also in the case $I_{xx} = I_{yy} = I_{zz}$ the tank keeps rotating with the initial angular velocity). With these conditions the solution of (3.21) can be found analytically, namely

$$\omega_x(t) = 2\cos\left(\frac{\pi}{2}t\right), \quad \omega_y(t) = 2\sin\left(\frac{\pi}{2}t\right), \quad \omega_z(t) = 1.$$

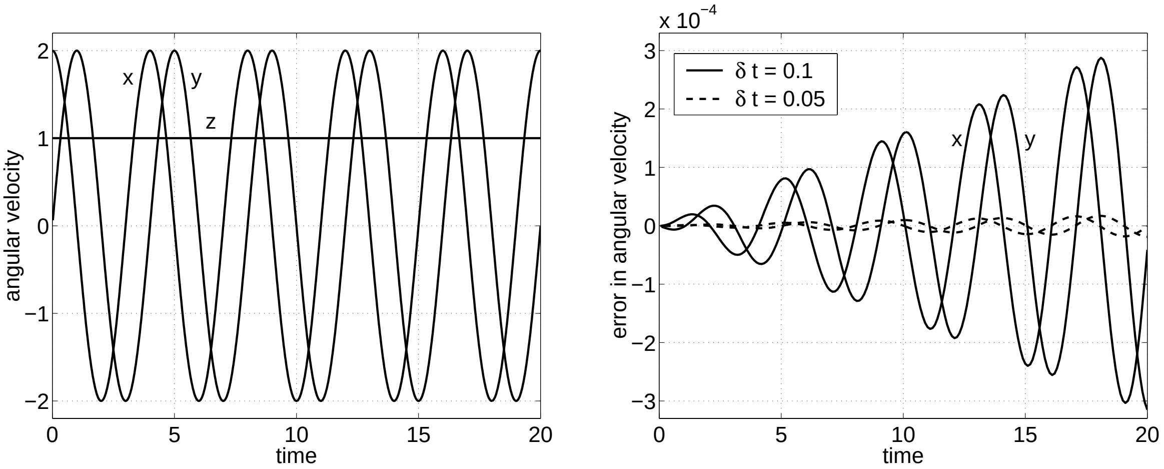


Figure 3.6: *Left: Evolution of the angular velocity around the three Cartesian axes for an empty tank with* $\boldsymbol{I}_s = \mathrm{diag}\,(1, 1, \pi/2 + 1)$ *and* $\boldsymbol{\omega}\,(0) = (2, 0, 1)^T$*. Right: Evolution of the error in the angular velocity around the* x*- and* y*-axis for two different time steps,* $\delta t = 0.1$ *(large error) and* $\delta t = 0.05$ *(small error).*

Results of the numerical simulation are shown in figure 3.6. On the left, lines indicated with x, y, and z show the numerical solutions for ω_x, ω_y, and ω_z respectively. On the right, the error in the solutions for ω_x and ω_y is plotted for two different time steps, $\delta t = 0.1$ and $\delta t = 0.05$. Due to the fourth-order Runge-Kutta time integration, the error in the simulation corresponding to $\delta t = 0.1$ is 16 times larger than the one corresponding to $\delta t = 0.05$.

Nonharmonic Solutions

If I_{xx}, I_{yy}, and I_{zz} are pairwise unequal, then (3.21) is more difficult to solve. Hence, only the conservation of kinetic energy is checked in this case, which is given by

$$E_{\text{kin}} = \frac{1}{2} I_{xx}\omega_x^2 + \frac{1}{2} I_{yy}\omega_y^2 + \frac{1}{2} I_{zz}\omega_z^2.$$

On the left in figure 3.7 the numerical solutions for ω_x, ω_y, and ω_z are shown in the case that $\boldsymbol{I}_s = \mathrm{diag}\,(1, 2, 4)$ and $\boldsymbol{\omega}\,(0) = (3, 2, 1)^T$. In this figure the kinetic energy minus the initial kinetic energy is plotted as a dashed line, which is clearly equal to zero during the entire simulation. A nice way to visualise the angular velocity is shown on the right in figure 3.7, where the projection $\boldsymbol{\omega}/|\boldsymbol{\omega}|$ of the angular velocity on the unit sphere is plotted. In the bounding box of this figure the projections of the normalised angular velocity in the xy-, xz-, and yz-plane are plotted.

3.4.3 Full Tank

In this section the motion of the same container as in the previous section is studied, *i.e.* with mass $m_s = 12$ and moment-of-inertia tensor $\boldsymbol{I}_s = \mathrm{diag}\,(1, 2, 4)$. However, now the container is completely filled with liquid. A simple calculation shows that the moment-of-inertia tensor of the liquid in this case is equal to $\boldsymbol{I}_l = \frac{1}{12} m_l\, \mathrm{diag}\left(\frac{7}{2} - 2\sqrt{2}, \frac{3}{2}, \frac{7}{2} - \sqrt{2}\right)$.

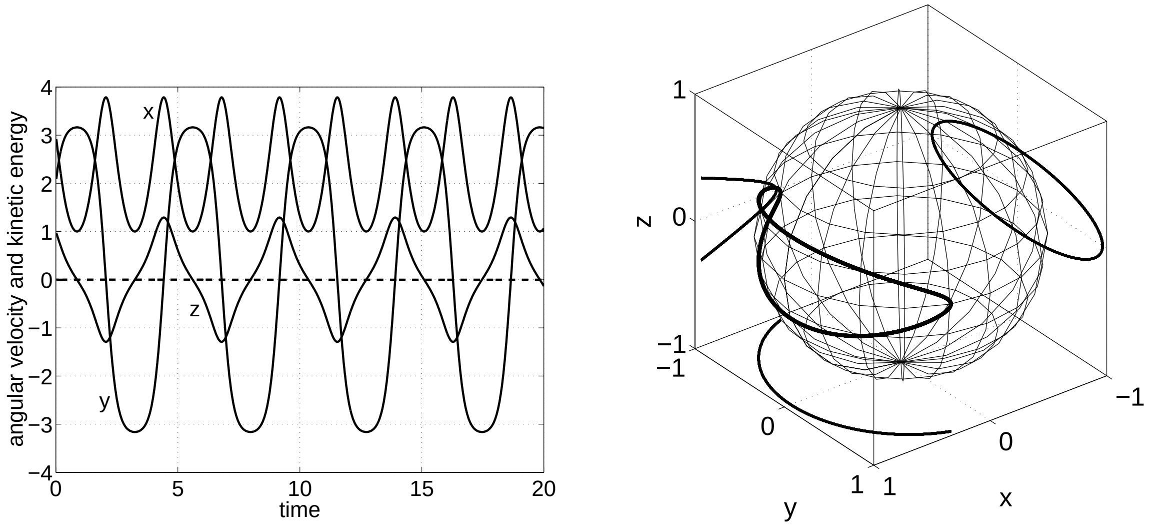


Figure 3.7: *Left: Evolution of the angular velocity (solid lines) around the three Cartesian axes for an empty tank with* $\boldsymbol{I}_s = \operatorname{diag}(1,2,4)$ *and* $\boldsymbol{\omega}(0) = (3,2,1)^T$. *The dashed line represents the kinetic energy minus the initial kinetic energy of the container. Right: Projection of the angular velocity on the unit sphere.*

The presence of fluid leads to a totally different motion of the container compared to the results in the previous section. Indeed, the coupled system settles itself in a state of minimum kinetic energy, which corresponds to rotation around the axis with maximum moment of inertia (the fluid is responsible for damping the rotation around the x- and y-axis). Thus, in the steady-state solution, the system rotates around the z-axis only. The final angular velocity around this axis follows from conservation of angular momentum. If the initial angular velocity is, again, chosen as $\boldsymbol{\omega}(0) = (3,2,1)^T$ and the liquid mass is set to $m_l = 120$ (*i.e.* a liquid/solid mass ratio of 10), then, theoretically, the final angular velocity around the z-axis is equal to 1.93. The results for this particular simulation are shown in figure 3.8. On the left, the angular velocities around the x-, y-, and z-axis are plotted versus time as well as the theoretically predicted final angular velocity around the z-axis. On the right, the projection of the angular velocity on the unit sphere is plotted. From these figures the transition to rotation around a single axis becomes very clear. The numerical simulation predicts a final angular velocity around the stable moment-of-inertia axis in close correspondence with the theoretical value; the difference is less than half a percent.

Grid-Refinement Study and Time-Step Analysis

Now that the container is filled with liquid, a grid-refinement study and a time-step analysis can be performed. The above simulation is repeated on grids consisting of $20 \times 20 \times 20$, $40 \times 40 \times 40$, $60 \times 60 \times 60$, and $80 \times 80 \times 80$ computational cells. The results are shown in figure 3.9 (for presentational reasons only the angular velocity around the x-axis on the four different grids is considered). On the right in this figure the solutions on subsequent grids are subtracted from each other, giving an indication of the error in the numerical solution. Clearly, the error decreases when the grid is refined. Note that

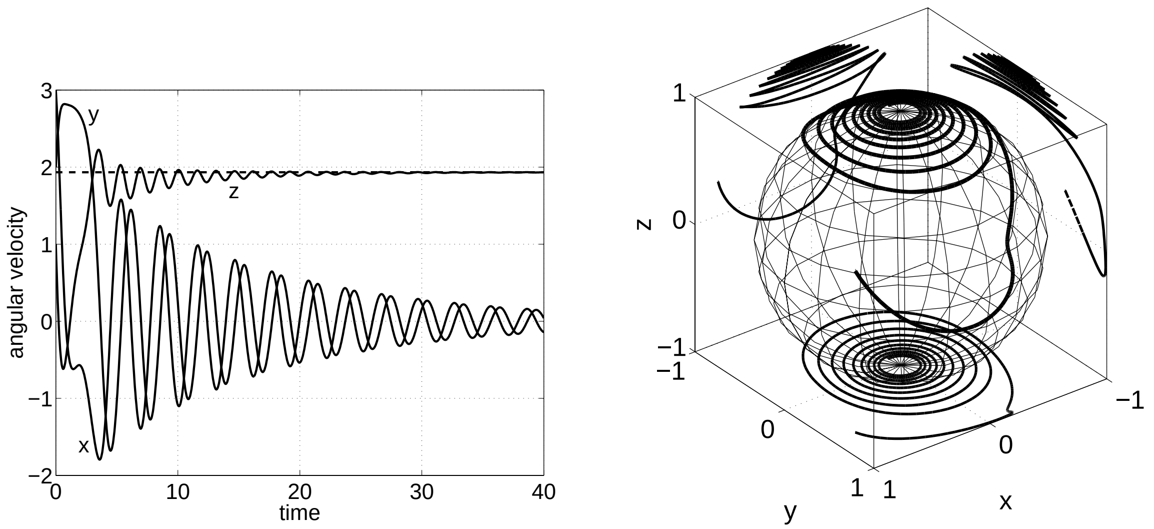


Figure 3.8: *Left: Evolution of the angular velocity (solid lines) around the three Cartesian axes for a completely filled tank. The dashed line represents the theoretically predicted final angular velocity around the z-axis. Right: Projection of the angular velocity on the unit sphere.*

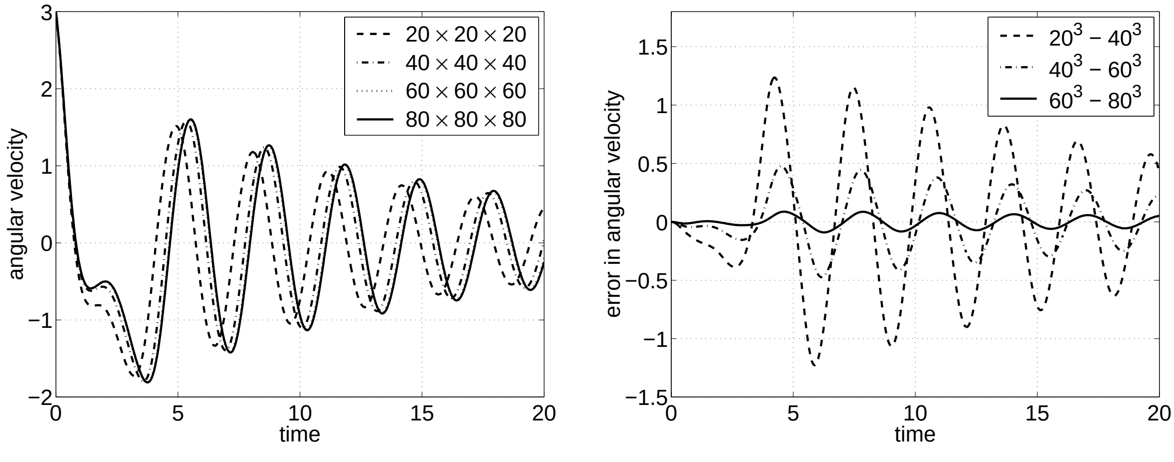


Figure 3.9: *Left: Angular velocity around x-axis of a full tank for various grids. Right: Indication of the error in the angular velocity given by the difference in the solutions on subsequent grids.*

the large size of the error can be explained by the phase difference between the solutions on the various grids.

All the simulations in this section have been performed with a time step of $\delta t = 10^{-3}$. In figure 3.10 the previous simulation is repeated on a grid of $40 \times 40 \times 40$ cells, but with different time steps, namely $\delta t = 5 \times 10^{-3}$, $\delta t = 1 \times 10^{-3}$, and $\delta t = 2 \times 10^{-4}$. The right-hand figure shows that the error in the angular velocity decreases when the time step is reduced.

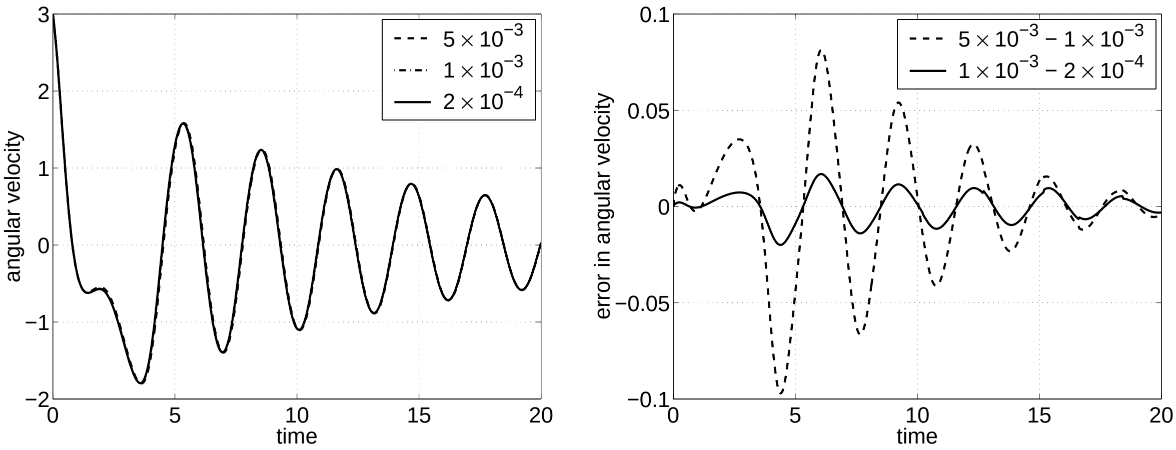


Figure 3.10: *Same as figure 3.9 but now for different time steps.*

Liquid/Solid Mass Ratio

Since the grid-refinement study and the time-step analysis revealed that the method is consistent, it is possible to examine how the coupled system that has been examined in this section reacts to a change in physical parameters, such as the liquid/solid mass ratio. In the previous simulations the liquid/solid mass ratio was equal to 10; the liquid mass and the solid-body mass were set to $m_l = 120$ and $m_s = 12$ respectively. If the liquid/solid mass ratio is increased (by increasing the density of the fluid), the fluid has more influence on the motion of the coupled system, whence the system settles itself faster into a steady state. On the other side, if the liquid/solid mass ratio is decreased, the system bears more resemblance to the motion of an empty tank. This phenomenon is illustrated on the left in figure 3.11, where the angular velocity of the coupled system is shown for liquid/solid mass ratios of 0.1, 1, 10, and 100. Note that the angular velocity converges to different values for different mass ratios, which is caused by a change in the liquid mass and thus the moment-of-inertia tensor. For presentational reasons only a detail of the time history of the angular velocity is shown in this figure. In the actual simulation the computation was continued until the amplitude of the oscillations in the angular velocity around the z-axis was less than 10^{-2} (for the simulation with a mass ratio of 0.1 this happened after approximately 400 seconds simulation time). On the right in figure 3.11 the computed angular velocity in the steady-state solution is compared to theory. Only the absolute value of the angular velocity is considered since the coupled system may be rotating in either positive or negative direction in the body-fixed coordinate system. The difference between simulation and theory is less than half a percent for all four mass ratios.

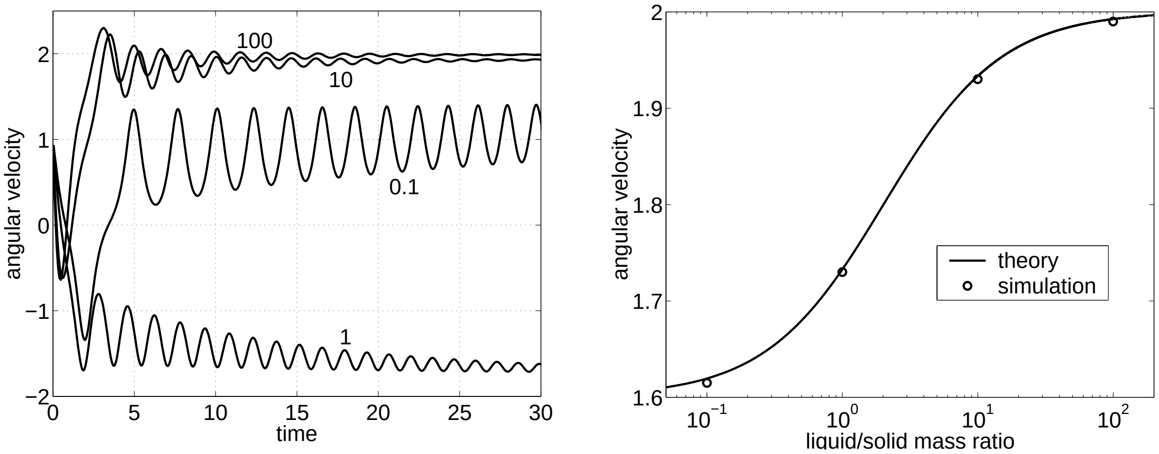


Figure 3.11: *Left: Evolution of the angular velocity around z-axis of a full tank for various liquid/solid mass ratios. Right: Comparison of the theoretical and numerical final angular velocity around z-axis.*

Although in this section only angular motion has been presented, in all the simulations the linear motion (which should be equal to zero) was computed from equation (3.8a). The linear acceleration turned out to be in the order of 10^{-6} for all the simulations in this section; the tolerance that was used in the computations. Thus, also the linear acceleration is predicted correctly in these simulations.

3.4.4 Partially Filled Tank

After studying an empty tank and a completely filled tank, in this section the intermediate case, a partially filled tank, is discussed. As an example a flat-spin motion is simulated. The same rectangular container that has been studied before is rotating around the x-axis with angular velocity $\omega_x = 5$ (initially, both ω_y and ω_z are equal to zero). The container is filled with fluid for $|y| \geq c$, where c determines the filling ratio. Since $I_{xx} < I_{yy} < I_{zz}$, rotation around the x-axis and z-axis is stable. However, because of the viscous liquid, kinetic energy is dissipated. Hence, in the steady state, the system will be rotating around the z-axis only, which corresponds to a state of minimum kinetic energy. This phenomenon — transition from rotation around the axis with minimum moment of inertia to rotation around the axis with maximum moment of inertia — is known as a flat spin. The concept of a flat spin is illustrated in figure 3.12. In the computation, the filling ratio was set to 65% and the density of the fluid was chosen such that the liquid/solid mass ratio equals $m_l/m_s = 10$. In figure 3.13 the angular velocities around the Cartesian axes are shown as a function of time. This figure clearly shows that some time (more than ten seconds) is needed before the transition is triggered. Once rotation around the y- and z-axis is excited, the change of rotation axis takes place in about ten seconds, although quite some time passes before the angular velocities around the x- and y-axis decrease to zero.

In figure 3.14 a more detailed study of the steady rotation around the x-axis at the beginning and around the z-axis at the end of the simulation is shown. In this figure snapshots of the free surface (solid lines) are drawn; on the left the results from the

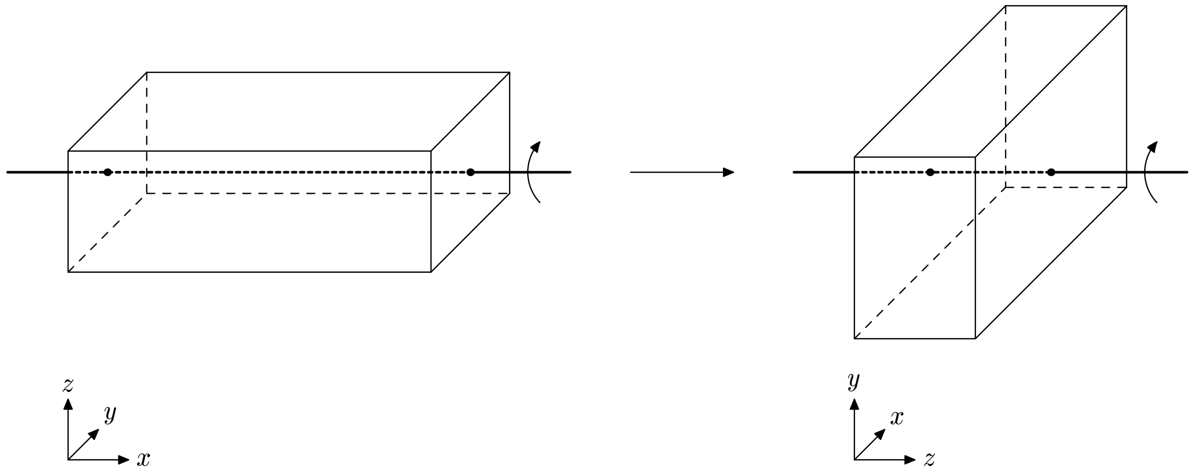


Figure 3.12: *Flat spin; change of rotation around axis with minimum moment of inertia to rotation around axis with maximum moment of inertia.*

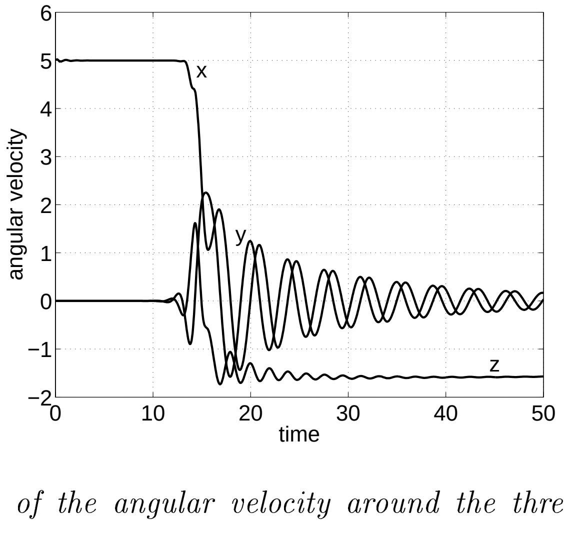


Figure 3.13: *Evolution of the angular velocity around the three Cartesian axes for a partially filled tank.*

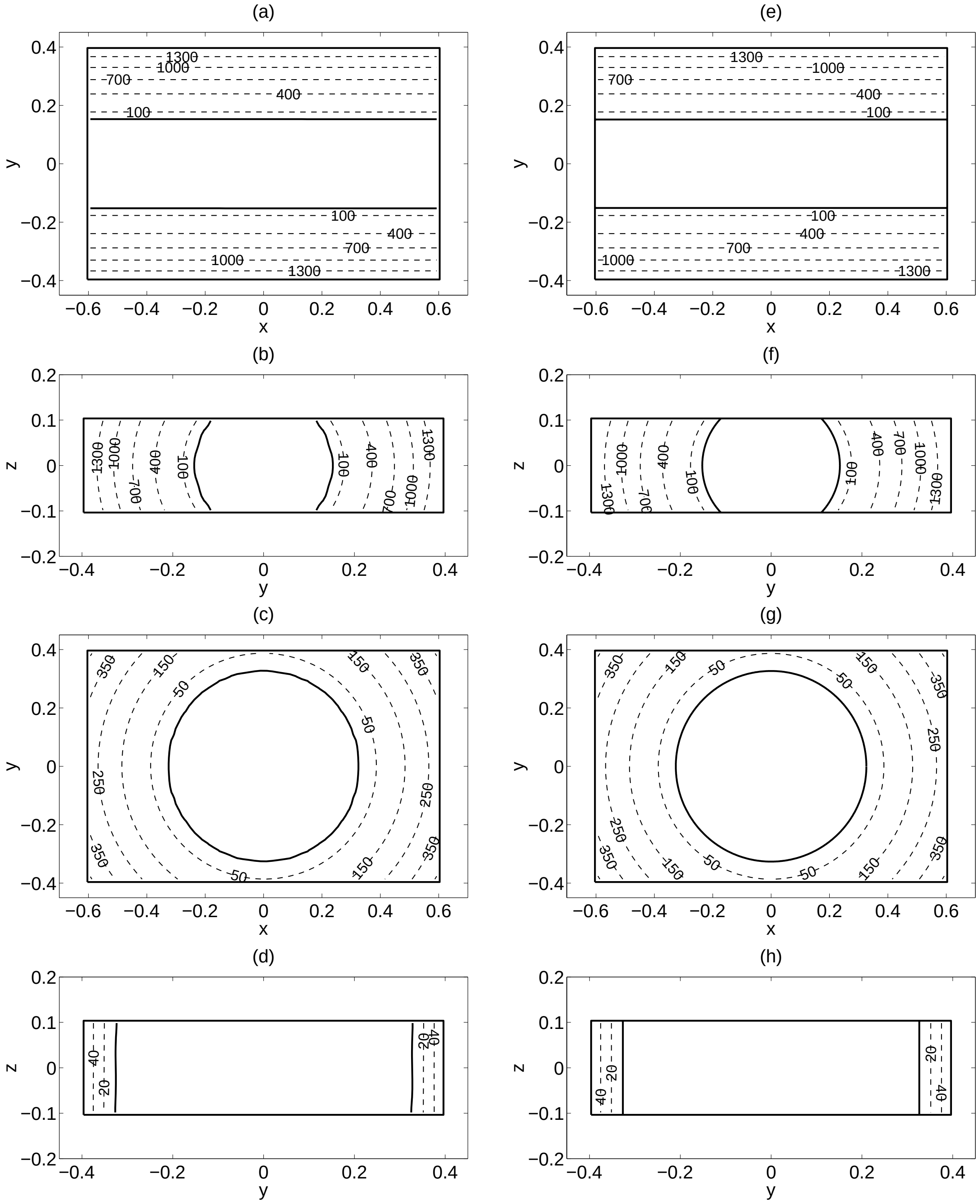


Figure 3.14: *Snapshots of the free surface (solid lines) and pressure contours (dashed lines) for the flat-spin motion in the xy- and yz-plane at two different time levels, namely at the beginning (top four) and at the end (bottom four) of the flat spin. On the left results from the simulation, on the right results based on theory.*

simulation and on the right the corresponding plots based on theory are shown. The results in two different cross sections, namely in the xy- and the yz-plane, and at two different time levels, namely at the beginning and at the end of the simulation, are presented. This figure clearly shows that initially the system is rotating around the x-axis and finally around the z-axis. Indeed, in the xy-plane the shape of the free surface changes from a straight line into a circle, while in the yz-plane opposite behaviour takes place.

When the system is rotating around one axis only, *i.e.* at the beginning and at the end of the simulation, it is easy to compare the pressure contours to theory since in this case the pressure contours are concentric circles around the axis of rotation. If the pressure at the free surface is equal to zero, then the pressure p at distance R from the centre of rotation is equal to

$$p = \frac{1}{2}\rho\,\omega^2\left(R^2 - r^2\right),$$

where ρ is the density of the fluid, ω is the angular velocity, and r is the distance between the free surface and the centre of rotation. This theory is compared to results from the numerical simulation in figure 3.14 as well, where the pressure contours are plotted as dashed lines. Obviously, the correspondence between simulation and theory is very good.

In figure 3.15 three-dimensional snapshots of a flat-spin simulation are shown, for which the filling ratio was decreased to 40%. In the upper-left snapshot, the container is rotating steadily around the axis with minimum moment of inertia. In the second snapshot, the beginning of the flat spin can be seen; liquid starts to flow due to rotation around the y- and z-axis. Some intermediate snapshots show the violent behaviour of the transition. Finally, in the lower-right snapshot, the angular velocities around the x- and y-axis have vanished and the container is rotating steadily around the axis with maximum moment of inertia.

3.4.5 The Wet Satellite Model Experiment

For studying the dynamics of liquid-filled spacecraft, the Wet Satellite Model (WSM) experiment was performed in 1992 [86]. Following the launch of the MASER 5 rocket on April 9, 1992, a small spacecraft, the Ejectable Ballistometer (EB), was separated from the MASER payload. After separation, the EB continued on its own trajectory until impact with earth after 370 seconds. Two photographs of the launch of the EB are shown in figure 3.16. During its lifetime the EB performed a flat-spin manoeuvre (see also the previous section).

The EB consists of an experiment tank that contains approximately 2.7 kilograms of water, which is about 50% of the total volume of the tank. The experiment tank is an annular cylinder of which the inner and outer radius are equal to 0.125 m and 0.145 m respectively and the height is equal to 0.323 m. The dry mass of the EB is 18.8 kg and its moment-of-inertia tensor is a diagonal matrix $\mathrm{diag}\,(0.25, 0.38, 0.37)$ kgm^2. Since the annular width of the tank is much smaller than the axial and azimuthal dimensions of the tank, the radial velocity of the liquid can be neglected with respect to the axial and azimuthal velocities. Also, the radius of curvature in the radial direction is much smaller than the radius of curvature in the axial and azimuthal directions. Hence, the free surface will deform in the axial and/or azimuthal directions mainly and the flow can be assumed

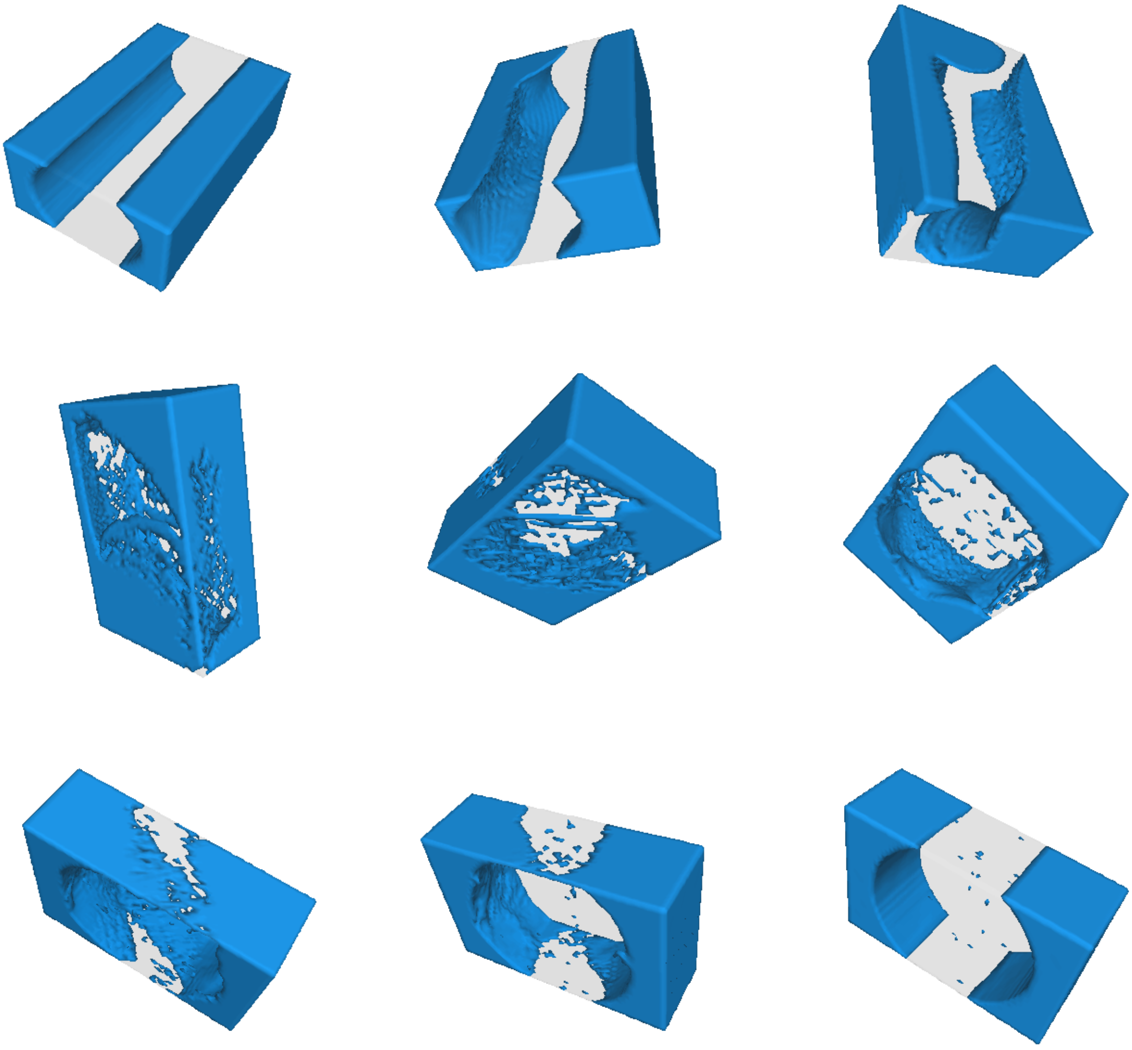


Figure 3.15: *Three-dimensional snapshots of the flat-spin simulation with a filling ratio of 40%.*

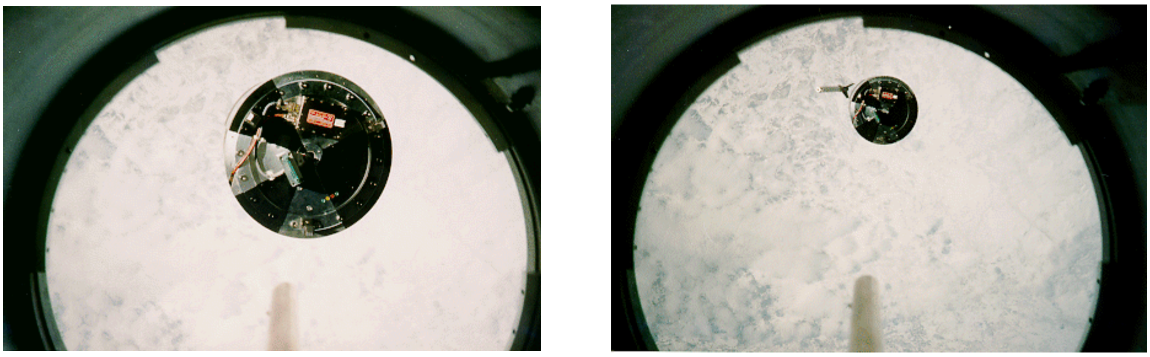


Figure 3.16: *Photographs of the separation from MASER of the Ejectable Ballistometer taken during the Wet Satellite Model experiment (these photographs were taken from http://www.nlr.nl/public/hosted-sites/mgcd/EXPMNT/A-015.html).*

to be two-dimensional. This simplification to a two-dimensional liquid-dynamics problem was an important feature of the WSM experiment since in 1992 no numerical models were known that could simulate a three-dimensional analogue of the WSM experiment.

When the EB was released from the MASER payload, its axis of symmetry was aligned with the direction of separation push. Hence, all the liquid inside the experiment tank was initially located in the bottom half of the tank. The initial motion of the EB was induced by transferring angular momentum from a flywheel. The flywheel axis was aligned with the axis of symmetry, *i.e.* the axis with minimum moment of inertia. Thus, the initial rotation of the EB was around this axis, whence a flat-spin transition could be expected. The amount of angular momentum transferred from the flywheel had been calibrated on ground, such that the initial angular velocity around the axis with minimum moment of inertia was known accurately.

A simulation of the WSM experiment has been performed with the method that has been presented in this thesis (although liquid sloshing has been studied in an EB-like geometry [85], ComFlo is the first method that simulates the WSM experiment). As in the experiment, the lower half of the annulus was filled with liquid and an initial angular velocity of 1.2 radians per second around the cylindrical axis was prescribed. For this simulation a grid consisting of 60 × 60 × 60 cells was used and the time step, which was automatically adjusted based on the CFL number, varied from $\delta t = 2.5 \times 10^{-4}$ to $\delta t = 4 \times 10^{-3}$. The total simulation time was 370 seconds corresponding to the experiment time. Results from the simulation and the actual experiment are shown in figure 3.17.

Although the geometry of the EB is not very suitable for a method based on Cartesian grids — a cylindrical grid would be more appropriate —, the qualitative agreement between experiment and simulation is satisfactory. Indeed, the interval in which the angular velocity around the x-axis decreases from its initial value to zero is of the same order (approximately 200 seconds) in both the simulation and the experiment. Further, the frequencies of the oscillations in the angular velocities around the y- and z-axis in the simulation are comparable with those in the experiment. In particular the high-frequency oscillations in the first half and the low-frequency oscillations during the second half of the flat spin appear both in the experiment and the simulation. Also, the amplitude of the oscillations in the simulation is of the same magnitude as in the experiment. In the experiment, after about 200 seconds, the angular velocity around the y-axis is ap-

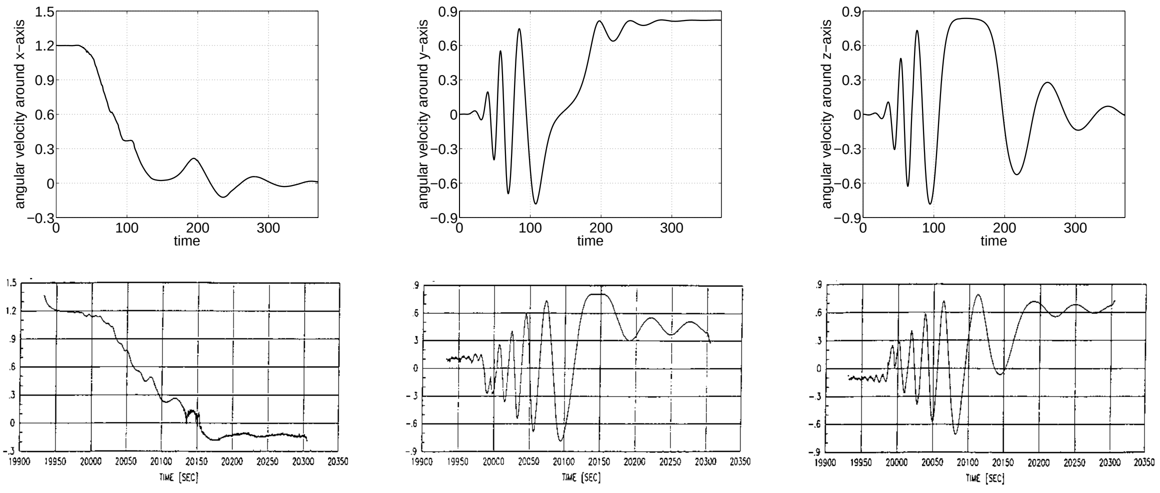


Figure 3.17: *Angular velocities around x- (left), y- (middle), and z-axis (right) in the Wet Satellite Model experiment. Comparison of simulation (top) to experiment (bottom). Note that the simulation time index 0 corresponds to an experiment time index of 19950 approximately.*

proximately constant for almost 40 seconds. This phenomenon is also present in the simulation, however, not in the angular velocity around the y-axis but around the z-axis and somewhat earlier than in the experiment.

In the simulation the angular velocity around the z-axis vanishes in the final stage of the flat spin. This behaviour is not visible in the experiment, where the EB seems to settle itself into a steady state in which it is rotating both around the y- and z-axis. Since the rotation axis in the steady state depends on the moment-of-inertia tensor of the coupled system and thus on the liquid distribution, apparently, the free-surface configuration that is predicted by the simulation does not agree with the experiment. Unfortunately, the free-surface configuration was not monitored during the WSM experiment. Hence, no further conclusions about this disagreement between simulation and experiment can be drawn.

Chapter 4

Summary and Conclusions

In this thesis a numerical method has been presented for simulating fluid flow under micro-gravity conditions and for the simulation of coupled solid-liquid dynamics. The numerical model has been implemented in a computer program called ComFlo with which all the simulations in this thesis have been performed.

Computational Grid

In chapter 2 a numerical method for discretising the Navier-Stokes equations on a three-dimensional, Cartesian grid was introduced. On a Cartesian grid it is relatively easy (compared to an unstructured grid) to advect a free surface. Further, grid generation becomes a trivial task compared to the generation of a body-fitted grid, which can be extremely time consuming.

For incorporating complex geometries the cut-cell technique was used, meaning that the boundary of the solid body cuts the Cartesian grid, hereby creating small, irregular grid cells. The representation of a complex geometry on a Cartesian grid was accomplished by introducing apertures, which indicate that part of a cell and its cell faces may be closed for flow. Simulation of wall-adhesion flow in a circular cylinder showed that use of the cut-cell technique is crucial for simulating fluid flow in complex geometries. Indeed, the same simulation in a staircase approximation of the circular cylinder gave deviant and incorrect results.

Discretisation Method

The finite-volume method was used for the discretisation of the Navier-Stokes equations. Thus, a weak formulation of the governing equations was the starting point for the numerical method. In the discrete approximations of the divergence and gradient operators apertures were included to account for complex geometries. The discretisation was done such that the discrete operators for convection and diffusion have the same symmetry properties as their continuous counterparts, *i.e.* the discrete operator for convection is a skew-symmetric matrix and the discrete operator for diffusion is a symmetric, negative-definite matrix. Further, the discrete gradient operator is equal to minus the transpose of the discrete divergence operator. From these symmetry properties it follows that, in the absence of external forces, the kinetic energy of the flow can not grow, making the numerical method extremely robust.

The discrete Navier-Stokes equations seem quite complex because of the inclusion of apertures and the incorporation of stretched grids. However, when applied to a staircase-shaped geometry on a uniform grid, the spatial discretisation reduces to a familiar, second-order finite-difference discretisation.

Despite the presence of small cells, no severe limitations on the time step were encountered. In fact, it can be proven that the discretisation of the convective terms that was exploited in this thesis leads to a time-step restriction that is based on the mesh size of the Cartesian grid; small cells do not make this restriction more stringent. Diffusive terms do strengthen the time-step restriction. However, by using a lower limit for the size of control volumes for discretising diffusion, this limitation is kept within bounds. Effectively, this precaution shifts the solid boundary over a small distance (smaller than the size of a computational cell) before applying the no-slip boundary condition.

Free-Surface Advection

Another focal point of chapter 2 is the numerical method for simulating interface flow. Two methods for interface advection were studied, namely the method of Hirt and Nichols and of Youngs. In the method of Hirt and Nichols the interface is reconstructed piecewise constant (thus, aligned with the Cartesian directions). The reconstructed interface is advected using limiters to prevent large undershoots and overshoots in the VOF function and to prevent unwanted steepening or smearing of the interface. In the method of Youngs a piecewise-linear reconstruction of the interface is computed. Based on the reconstructed interface, fluxes are computed which are transported unchanged from a donor cell to an acceptor cell. It was shown that both advection methods have severe problems with mass conservation. Moreover, the method of Hirt and Nichols creates a lot of flotsam and jetsam. The problems encountered in both advection methods can be solved by introducing a local height function near the free surface. In every surface cell, locally a height function is defined in a 3×3 (two dimensions) or $3 \times 3 \times 3$ (three dimensions) block of grid cells. The fluxes that have been computed at cell faces are not directly transported from a donor cell to an acceptor cell, but instead the value of the local height function is updated. This method for advecting the free surface conserves mass rigorously and creates far less flotsam and jetsam.

In fluid-flow problems with a prescribed velocity field, the use of a local height function improves the method of Hirt and Nichols a lot, while Youngs' method is less beneficiary. In these problems Youngs' method (with or without a height function) performs superior. However, in simulations where the velocity field is solved from the Navier-Stokes equations, Youngs' method (with or without a height function) gives worse results than the method of Hirt and Nichols combined with a local height function. This is due to the linear reconstruction of the interface. For this, the normal of the interface is predicted using the VOF field around the cell under consideration. Since the velocity field is computed up to a user-defined accuracy, small errors occur in the distribution of the VOF field, for which the computation of the normal, and thus the entire advection algorithm, is very sensitive. Typically, Youngs' method loses symmetry quickly in symmetrical flow problems. The method of Hirt and Nichols is not sensitive to small variations in the VOF field since the interface is reconstructed piecewise constant. Therefore, the method of Hirt and Nichols combined with a local height function is preferred in this thesis.

Micro-Gravity

In this thesis, capillary forces, such as surface tension and wall adhesion, could not be neglected since a numerical model was needed for the simulation of liquid dynamics in space. Therefore, the boundary conditions at the free surface required much attention. Especially, a correct implementation of the boundary conditions for the pressure and the contact angle is crucial for obtaining accurate results. Indeed, in the various wall-adhesion simulations described in this thesis, the contact angle and surface tension drive the flow and determine the steady-state solution. The pressure at the free surface depends on its curvature, which is computed using the local height function that was introduced earlier for advecting the free surface. By means of grid refinement and comparing results from simulations (with various contact angles) to theory, the numerical method has been validated. In two dimensions, both in simple and complex geometries, the results agree closely. The three-dimensional simulation of wall adhesion in a sphere showed that the free surface is not perfectly axisymmetric. However, the steady-state solution is adequately predicted considering the fact that all the information in these simulations comes from the contact line, which is a one-dimensional subset of the flow domain.

Coupled Dynamics

The model that has been developed in chapter 2 was extended in chapter 3 by coupling the liquid dynamics with the dynamics of the solid body in which the liquid is contained. This required a model for the solid-body dynamics in which terms appear representing the force that the sloshing liquid induces on the solid body. Further, the model for the liquid dynamics was extended with a virtual body force accounting for the solid-body motion.

The governing equations for the solid-body dynamics are naturally written with respect to an inertial reference frame. Hence, also the force due to the sloshing liquid is written with respect to this reference frame. However, discretisation of these equations results in a numerical method that is unstable if the liquid mass becomes too large with respect to the mass of the solid body. This is demonstrated in a simulation of the free fall of a liquid-filled container. This stability problem was solved by writing the sloshing force as a sum of two contributions. The first contribution is the force resulting from linear and angular accelerations of the fluid, due to motion of the solid body. The second contribution is the force due to accelerations of the fluid with respect to the moving reference frame. The former contribution is then treated simultaneously with the motion of the solid body. Discretisation of this alternative mathematical model for the solid-body dynamics results in a numerical model that is stable for arbitrary liquid/solid mass ratios. This unconditional stability was exemplified with a mass-spring model and demonstrated with simulations where the liquid/solid mass ratio ranged from 0.1 to 100.

Flat Spin

An important example of coupled solid-liquid dynamics is the flat-spin motion of a (partially) liquid-filled spacecraft. In this case the coupled system is initially rotating around its axis with minimum moment of inertia. From physics it is known that a free-flying body can rotate uniformly around one of the three principal moment-of-inertia axes only;

rotation around the axis with intermediate moment of inertia is unstable, while rotation around the other two axes is stable. Thus, the initial condition (rotation around the axis with minimum moment of inertia) can be a steady state. However, if damping in the coupled system occurs (*e.g.* due to the viscous liquid), kinetic energy is lost. Hence, in this case, rotation around the axis with maximum moment of inertia (corresponding to a state of minimum kinetic energy) is the steady state in which the coupled system settles itself. A flat-spin motion of a three-dimensional, rectangular container partially filled with liquid has been simulated. This simulation showed that the numerical method can predict a flat spin correctly. A quantitative comparison of a flat spin was made by simulating the Wet Satellite Model (WSM) experiment, which was performed in space in 1992. In fact, ComFlo is the first computational fluid dynamics method that performed a simulation of the WSM experiment. Although the geometry of the WSM (a thin annulus) is not very suitable for a numerical method based on Cartesian grids, results of the simulation agree adequately with the experiment.

Epilogue

From the results of the various simulations that have been discussed in this thesis (and from the many more simulations that have been performed in developing the numerical method, but are not reported here), it can be concluded that the present method is capable of simulating liquid dynamics under micro-gravity conditions and can predict the coupled dynamics of liquid-filled spacecraft. The method is very robust and applicable for simulation of terrestrial fluid-flow problems as well. However, for further (physical) understanding of liquid behaviour in space — especially with respect to contact-line dynamics — and validation of numerical methods, experiments remain extremely valuable. Hence, the forthcoming launch of the experiment satellite SloshSat FLEVO will hopefully be a success and help in the further development of micro-gravity computational fluid dynamics.

Bibliography

[1] A.S. Almgren, J.B. Bell, P. Colella, and T. Marthaler. A cell-centered Cartesian grid projection method for the incompressible Euler equations in complex geometries. In *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference*, San Diego, CA, June 19–22, 1995. AIAA-95-1743-CP.

[2] N. Ashgriz and J.Y. Poo. FLAIR: flux line segment model for advection and interface reconstruction. *J. Comp. Phys.*, 93:449–468, 1991.

[3] V. Babu and S. Korpela. Numerical solution of the incompressible three-dimensional Navier-Stokes equations. *Comput. & Fluids*, 23(5):675–691, 1994.

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

[5] J.U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modelling surface tension. *J. Comp. Phys.*, 100:335–354, 1992.

[6] M. Bussman, S. Chandra, and J. Mostaghimi. Modeling the splash of a droplet impacting a solid surface. *Phys. Fluids*, 12(12):3121–3132, 2000.

[7] D. Calhoun and R.J. LeVeque. A Cartesian grid finite-volume method for the advection-diffusion equation in irregular geometries. *J. Comp. Phys.*, 157:143–180, 2000.

[8] K.-H. Chen and R.H. Pletcher. Simulation of three-dimensional liquid sloshing flows using a strongly implicit calculation procedure. *AIAA J.*, 31(5):901–910, 1993.

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

[10] S. Chen, D.B. Johnson, P.E. Raad, and D. Fadda. The surface marker and micro cell method. *Int. J. Numer. Meth. Fluids*, 25:749–778, 1997.

[11] D.K. Clarke, M.D. Salas, and H.A. Hassan. Euler calculations for multielement airfoils using Cartesian grids. *AIAA J.*, 24(3):353–358, 1986.

[12] E.G.M. Coenen. *Viscous-inviscid interaction with the quasi-simultaneous method for 2D and 3D aerodynamic flow.* PhD thesis, University of Groningen, The Netherlands, 2001.

[13] A.B. Cortes and J.D. Miller. Numerical experiments with the lid driven cavity flow problem. *Comput. & Fluids*, 23(8):1005–1027, 1994.

[14] 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 T. Miloh and G. Zilman, editors, *Proceedings 15th International Workshop on Water Waves and Floating Bodies*, Caesarea, Israel, February 27 – March 1, 2000.

[15] E.F.G. van Daalen, K.M.T. Kleefsman, J. Gerrits, H.R. Luth, and A.E.P. Veldman. Anti-roll tank simulations with a volume of fluid (VOF) based Navier-Stokes solver. In *Proceedings 23rd Symposium on Naval Hydrodynamics*, Val de Reuil, France, September 17–22, 2000.

[16] F.T. Dodge, S.T. Green, and M.W. Cruse. Analysis of small-amplitude low gravity sloshing in axisymmetric tanks. *Microgravity Sci. Technol.*, IV(4):228–234, 1991.

[17] M Dreyer, A. Delgado, and H.-J. Rath. Fluid motion in capillary vanes under reduced gravity. *Microgravity Sci. Technol.*, V(4):203–210, 1993.

[18] M Dreyer, A. Delgado, and H.-J. Rath. Capillary rise of liquid between parallel plates under microgravity. *J. Colloid Interface Sci.*, 163:158–168, 1994.

[19] J.K. Dukowicz. Efficient volume computation for three-dimensional hexahedral cells. *J. Comp. Phys.*, 74:493–496, 1988.

[20] G. Fekken, A.E.P. Veldman, and B. Buchner. Simulation of green-water loading using the Navier-Stokes equations. In J. Piquet, editor, *Proceedings 7th International Conference on Numerical Ship Hydrodynamics*, pages 6.3–1–6.2–12, Nantes, France, July 19–22, 1999.

[21] D.E. Fyfe, E.S. Oran, and M.J. Fritts. Surface tension and viscosity with Lagrangian hydrodynamics on a triangular mesh. *J. Comp. Phys.*, 76:349–384, 1988.

[22] J. Gerrits, G.E. Loots, G. Fekken, and A.E.P. Veldman. Liquid sloshing on earth and in space. In B. Šarler, C.A. Brebbia, and H. Power, editors, *Moving Boundaries V*, pages 111–120. WIT Press, Southampton, UK, 1999.

[23] J. Gerrits and A.E.P. Veldman. Numerical simulation of coupled liquid-solid dynamics. In E. Oñate, G. Bugeda, and B. Suárez, editors, *Proceedings ECCOMAS 2000*, Barcelona, Spain, September 11–14, 2000. Paper 575.

[24] J. Gerrits and A.E.P. Veldman. Transient dynamics of containers partially filled with liquid. In B. Šarler and C.A. Brebbia, editors, *Moving Boundaries VI*, pages 63–72. WIT Press, Southampton, UK, 2001.

[25] J. Glimm, J.W. Grove, X.L. Li, and D.C. Tan. Robust computational algorithms for dynamic interface tracking in three dimensions. *SIAM J. Sci. Comput.*, 21(6):2240–2256, 2000.

[26] H.P. Greenspan. *The theory of rotating fluids.* Cambridge University Press, 1969.

[27] 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. Comp. Phys.*, 152:423–456, 1999.

[28] P. Hansbo. Lagrangian incompressible flow computations in three dimensions by use of space-time finite elements. *Int. J. Numer. Meth. Fluids*, 20:989–1001, 1995.

[29] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 8(12):2182–2189, 1965.

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

[31] 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*, 35:151–172, 2001.

[32] P. He, M. Salcudean, I.S. Gartshore, and P. Nowak. Multigrid calculation of fluid flows in complex 3D geometries using curvilinear grids. *Comput. & Fluids*, 25(4):395–419, 1996.

[33] B.T. Helenbrook, L. Martinelli, and C.K. Law. A numerical method for solving incompressible flow problems with a surface of discontinuity. *J. Comp. Phys.*, 148:366–396, 1999.

[34] D.W. Hewett. The embedded curved boundary method for orthogonal simulation meshes. *J. Comp. Phys.*, 138:585–616, 1997.

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

[36] C.W. Hirt and J.P. Shannon. Free-surface stress conditions for incompressible-flow calculations. *J. Comp. Phys.*, 2:403–411, 1968.

[37] J.Y. Huang. Moving coordinates methods and applications to the oscillations of a falling slender body. In B. Šarler and C.A. Brebbia, editors, *Moving Boundaries VI*, pages 73–82. WIT Press, Southampton, UK, 2001.

[38] B. Iannotta. Slosh, rattle and roll. *New Scientist*, pages 32–35, 27 May 2000.

[39] D.B. Johnson, P.E. Raad, and S. Chen. Simulation of impacts of fluid free surfaces with solid boundaries. *Int. J. Numer. Meth. Fluids*, 19:153–176, 1994.

[40] F.Y. Kafka and E.B. Dussan. On the interpretation of dynamic contact angles in capillaries. *J. Fluid Mech.*, 95:539–565, 1979.

[41] F.J. Kelecy and R.H. Pletcher. The development of a free surface capturing approach for multidimensional free surface flows in closed containers. *J. Comp. Phys.*, 138:939–980, 1997.

[42] S. Kim and H.C. No. Second-order model for free surface convection and interface reconstruction. *Int. J. Numer. Meth. Fluids*, 26:79–100, 1998.

[43] D.B. Kothe. Perspective on Eulerian finite volume methods for incompressible interfacial flows. In H.C. Kuhlmann and H.-J. Rath, editors, *Free Surface Flows*, pages 267–331. Springer Verlag, New York, NY, USA, 1997.

[44] D.B. Kothe and R.C. Msjolsness. RIPPLE: a new model for incompressible flows with free surfaces. *AIAA J.*, 30(11):2692–2700, 1992.

[45] D.B. Kothe, W.J. Rider, Mosso S.J., Brock J.S., and J.I. Hochstein. Volume tracking of interfaces having surface tension in two and three dimensions. Technical Report AIAA 96-0859, Los Alamos National Laboratory, 1996. Presented at the 34th Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, January 1996.

[46] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti. Modelling merging and fragmentation in multiphase flows with SURFER. *J. Comp. Phys.*, 113:134–147, 1994.

[47] C.M. Lemos. Higher-order schemes for free-surface flows with arbitrary configurations. *Int. J. Numer. Meth. Fluids*, 23:545–566, 1996.

[48] R.J. LeVeque. Cartesian grid methods for flow in irregular regions. In K.W. Morton and M.J. Baines, editors, *Numerical methods for fluid dynamics III*, pages 375–382. Clarendon Press, Oxford, UK, 1988.

[49] P.Y. Liang. Numerical method for calculation of surface tension flows in arbitrary grids. *AIAA J.*, 29(2):161–167, 1991.

[50] W. Lin and C.J. Chen. Automatic grid generation of complex geometries in Cartesian co-ordinates. *Int. J. Numer. Meth. Fluids*, 28:1303–1324, 1998.

[51] N. Lock, M. Jaeger, M. Medale, and R. Occelli. Local mesh adaptation technique for front tracking problems. *Int. J. Numer. Meth. Fluids*, 28:719–736, 1998.

[52] H. Miyata. Finite-difference simulation of breaking waves. *J. Comp. Phys.*, 65:179–214, 1986.

[53] K. Morinishi. A finite difference solution of the Euler equations on non-body fitted Cartesian grids. *Comput. & Fluids*, 21(3):331–344, 1992.

[54] T. Nakayama and K. Washizu. Nonlinear analysis of liquid motion in a container subjected to forced pitching oscillation. *Int. J. Numer. Meth. Engrg.*, 15:1207–1220, 1980.

[55] S.E. Navti, K. Ravindran, C. Taylor, and R.W. Lewis. Finite element modelling of surface tension effects using a Lagrangian-Eulerian kinematic description. *Comput. Methods Appl. Mech. Engrg.*, 147:41–60, 1997.

[56] R.B. Pember, J.B. Bell, P. Colella, W.Y. Crutchfield, and M.L. Welcome. An adaptive Cartesian grid method for unsteady compressible flow in irregular regions. *J. Comp. Phys.*, 120:278–304, 1995.

[57] D.H. Peregrine. The effect of entrained air in violent water wave impacts. *J. Fluid Mech.*, 325:377–397, 1996.

[58] M. Perić, R. Kessler, and Scheuerer G. Comparison of finite-volume numerical methods with staggered and colocated grids. *Comput. & Fluids*, 16(4):389–403, 1988.

[59] J.Y. Poo and N. Ashgriz. A computational method for determining curvatures. *J. Comp. Phys.*, 84:483–491, 1989.

[60] S. Popinet and S. Zaleski. A front-tracking algorithm for accurate representation of surface tension. *Int. J. Numer. Meth. Fluids*, 30:775–793, 1999.

[61] W.J. Rider and D.B. Kothe. Reconstructing volume tracking. *J. Comp. Phys.*, 141:112–152, 1998.

[62] M. Rudman. Volume-tracking methods for interfacial flow calculations. *Int. J. Numer. Meth. Fluids*, 24:671–691, 1997.

[63] M. Rudman. A volume-tracking method for incompressible multifluid flows with large density variations. *Int. J. Numer. Meth. Fluids*, 28:357–378, 1998.

[64] W. Rumold. Modeling and simulation of vehicles carrying liquid cargo. *Multibody System Dynamics*, 5:351–374, 2001.

[65] Z.A. Sabeur, J.E. Cohen, J.R. Stephens, and A.E.P. Veldman. Investigation on free-surface flow oscillatory impact pressures with the Volume-of-Fluid method. In M.J. Baines, editor, *Numerical Methods for Fluid Dynamics VI*, pages 493–498. Will Print, Oxford, UK, 1998.

[66] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech.*, 31:567–603, 1999.

[67] R. Scardovelli and S. Zaleski. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *J. Comp. Phys.*, 164:228–237, 2000.

[68] T.M. Shih, C.H. Tan, and B.C. Hwang. Effects of grid staggering on numerical schemes. *Int. J. Numer. Meth. Fluids*, 9:193–212, 1989.

[69] 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(4):1165–1191, 1999.

[70] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. An improved level set method for incompressible two-phase flows. *Comput. & Fluids*, 27(5–6):663–680, 1998.

[71] 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. Comp. Phys.*, 162:301–337, 2000.

[72] M. Sussman and P. Smereka. Axisymmetric free boundary problems. *J. Fluid Mech.*, 341:269–294, 1997.

[73] E.Y. Tau. A second-order projection method for the incompressible Navier-Stokes equations in arbitrary domains. *J. Comp. Phys.*, 115:147–152, 1994.

[74] O. Ubbink and R. Issa. A method for capturing sharp fluid interfaces on arbitrary meshes. *J. Comp. Phys.*, 153:26–50, 1999.

[75] H.S. Udaykumar, H.-C. Kan, W. Shyy, and R. Tran-Son-Tay. Multiphase dynamics in arbitrary geometries on fixed Cartesian grids. *J. Comp. Phys.*, 137:366–405, 1997.

[76] O.V. Vasilyev. High order finite difference schemes on non-uniform meshes with good conservation properties. *J. Comp. Phys.*, 157:746–761, 2000.

[77] A.E.P. Veldman. New, quasi-simultaneous method to calculate interacting boundary layers. *AIAA J.*, 19:79–85, 1981.

[78] A.E.P. Veldman and M.E.S. Vogels. Axisymmetric liquid sloshing under low-gravity conditions. *Acta Astronautica*, 11(10/11):641–649, 1984.

[79] R.W.C.P. Verstappen and A.E.P. Veldman. A symmetry-preserving Cartesian grid method for turbulent flow. To be published.

[80] R.W.C.P. Verstappen and A.E.P. Veldman. Data-parallel solution of the incompressible Navier-Stokes equations. In P. Wesseling, editor, *High performance computing in fluid dynamics*, pages 237–260. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[81] R.W.C.P. Verstappen and A.E.P. Veldman. Spectro-consistent discretisation of Navier-Stokes: a challenge to RANS and LES. *J. Engrg. Math.*, 34:163–179, 1998.

[82] R.W.C.P. Verstappen and A.E.P. Veldman. Numerical computation of a viscous flow around a circular cylinder on a Cartesian grid. In E. Oñate, G. Bugeda, and B. Suárez, editors, *Proceedings ECCOMAS 2000*, Barcelona, Spain, September 11–14, 2000. Paper 423.

[83] J.A. Viecelli. A method for including arbitrary external boundaries in the MAC incompressible fluid computing technique. *J. Comp. Phys.*, 4:543–551, 1969.

[84] J.A. Viecelli. A computing method for incompressible flows bounded by moving walls. *J. Comp. Phys.*, 8:119–143, 1971.

[85] M.E.S. Vogels. A numerical method for the simulation of liquid-solid body dynamics. Technical Report MP 87030, National Aerospace Laboratory NLR, 1988. Presented at the 12th IMACS World Congress on Scientific Computation, Paris, France, July 1988.

[86] J.P.B. Vreeburg. The wet satellite model experiment. Technical Report TP 94196, National Aerospace Laboratory NLR, 1994.

[87] J.P.B. Vreeburg. Simulation of liquid dynamics onboard Sloshsat FLEVO. Technical Report TP 99236, National Aerospace Laboratory NLR, 1999.

[88] J.P.B. Vreeburg. Diagnosis of water motion in the Sloshsat FLEVO tank. Technical Report TP 2000-061, National Aerospace Laboratory NLR, 2000. Presented at the 5th International Astronautical Congress, Amsterdam, The Netherlands, October 1999.

[89] J.P.B. Vreeburg and A.E.P. Veldman. Transient and sloshing motions in an unsupported container. In R. Monti, editor, *Physics of fluids in microgravity*, pages 293–321. Gordon and Breach Publishing, 2002.

[90] M.W. Williams, D.B. Kothe, and E.G. Puckett. Accuracy and convergence of continuum surface tension models. Technical Report LA-UR-98-2268, Los Alamos National Laboratory, 1998.

[91] G. Wölk, M. Dreyer, H.-J. Rath, and M.M. Weislogel. Damped oscillations of a liquid/gas surface upon step reduction in gravity. *J. Spacecraft and Rockets*, 34(1):110–117, 1997.

[92] D.J. Wood, D.H. Peregrine, and T. Bruce. Wave impact on a wall using pressure-impulse theory. *J. Waterway, Port, Coastal, and Ocean Engng.*, pages 182–190, July/August 2000.

[93] H. Worth. NEAR team recovers mission after faulty engine burn. Available at http://near.jhuapl.edu/news/articles/99jan29_1, January 1999.

[94] F. Xiao. A computational model for suspended large rigid bodies in 3D unsteady viscous flows. *J. Comp. Phys.*, 155:348–379, 1999.

[95] G. Yang, D.M. Causon, and D.M. Ingram. Cartesian cut-cell method for axisymmetric separating body flows. *AIAA J.*, 37(8):905–911, 1999.

[96] D.L. Youngs. An interface tracking method for a 3D Eulerian hydrodynamics code. Technical Report AWRE/44/92/35, Atomic Weapons Research Establishment, April 1987.

[97] D. de Zeeuw and K.G. Powell. An adaptively refined Cartesian mesh solver for the Euler equations. *J. Comp. Phys.*, 104:56–68, 1993.

Samenvatting

Dynamica van vloeistofgevulde ruimtevaartuigen

Ruimtevaartprojecten zijn vaak geldverslindend: de bouw en lancering van een kleine satelliet kost al gauw tientallen miljoenen euro. Het verloren gaan van een ruimtevaartuig is dan ook een hoogst onwenselijke gebeurtenis. Daarom wordt er bij de bouw van een satelliet weinig aan het toeval overgelaten; er worden hoge veiligheidseisen gesteld en met talloze rampscenario's wordt rekening gehouden. Dit is noodzakelijk, omdat de vluchtleiding na de lancering slechts een beperkte controle over een satelliet heeft.

Helaas is anticipatie op problemen niet altijd mogelijk, omdat met betrekking tot de dynamica van ruimtevaartuigen een aantal zaken nog niet goed begrepen is. Een cruciale rol hierbij speelt de aanwezigheid van vloeistoffen (bijvoorbeeld brandstof) aan boord van een satelliet.

Ten eerste is het gedrag van vloeistoffen in een gewichtsloze omgeving anders dan op aarde. Op aarde worden vloeistoffen onder invloed van de zwaartekracht naar het zwaartepunt van de aarde getrokken en bij afwezigheid van andere krachten zal het vloeistofoppervlak in de rusttoestand horizontaal zijn. In de ruimte, waar de invloed van de zwaartekracht te verwaarlozen is, spelen capillaire krachten een belangrijke rol. Dit zijn krachten die geconcentreerd zijn op het vloeistofoppervlak en zorgen voor een zekere weerstand tegen het opbreken van het oppervlak en proberen de grootte van het oppervlak te minimaliseren (denk aan een zeepbel). Op de plek waar het vloeistofoppervlak in aanraking is met een vaste wand zorgen capillaire krachten ervoor dat de vloeistof bij de wand omhoog kruipt (of, afhankelijk van vloeistofeigenschappen en type wand, zich van de wand verwijdert); het vloeistofoppervlak en de vaste wand snijden elkaar onder een zekere hoek, de zogenaamde contacthoek. Dit effect is ook op aarde waarneembaar: in een glas water is duidelijk te zien dat het wateroppervlak schuin tegen het glas omhoog staat.

Ten tweede beïnvloeden de beweging van de satelliet en de zich aan boord bevindende klotsende vloeistof elkaar. Dit is als volgt te begrijpen: Ten einde een satelliet in een juiste baan te krijgen en te houden, zijn af en toe koerscorrecties nodig. Deze worden bereikt door het afvuren van stuurraketjes, hetgeen niet alleen de satelliet maar ook de vloeistof in beweging zet. De klotsende vloeistof, op zijn beurt, veroorzaakt krachten op de satelliet en beïnvloedt diens beweging, enzovoort. Deze complexe interactie tussen de vloeistofdynamica en de dynamica van het ruimtevaartuig (kortweg dynamische interactie genoemd) is moeilijk te voorspellen en kan vervelende gevolgen hebben, zoals in 1998 is gebeurd met de NEAR satelliet (NEAR is een acroniem voor Near Earth Asteroid Rendezvous). Deze satelliet werd gelanceerd met de bedoeling in een baan rond de asteroïde Eros te geraken en vervolgens ook een landing op dit rotsblok van $13 \times 13 \times 33$ kubieke kilometer te maken. Onderweg naar Eros werd de satelliet tijdens een koerscorrectie uit voorzorg door de boordcomputer uitgeschakeld, omdat interne sensoren versnellingen van de satelliet registreerden die groter waren dan de veiligheid toeliet. Na nader onderzoek werden deze excessieve versnellingen toegeschreven aan het klotsen van de brandstof. Uiteindelijk heeft het tuimelen van de satelliet een vertraging van ruim een jaar in de missie veroorzaakt; de landing op Eros vond plaats op 12 februari 2001.

Ook op aarde zijn voorbeelden van dynamische interactie te vinden. Denk bijvoorbeeld aan tankwagens op de weg die gevuld met vloeistof een aanzienlijk langere remweg hebben. Aan boord van olietankers wordt juist dankbaar gebruik gemaakt van de effecten van dynamische interactie. In zogenaamde antislingertanks, mits goed ontworpen, zorgt klotsende vloeistof ten gevolge van het slingeren van het schip voor een stabiliserende werking.

Theorie, experiment en simulatie

Van oudsher is het experiment de voornaamste methode om stroming van vloeistoffen te bestuderen. Tot op de dag van vandaag zijn experimenten zeer populair. Een nieuw vliegtuigontwerp wordt dan ook nog steeds maandenlang onderworpen aan windtunneltesten. Echter, experimenten met als doel het gedrag van vloeistoffen in de ruimte beter te begrijpen zijn zeer schaars. In de eerste plaats is het natuurlijk erg lastig om deze experimenten op aarde uit te voeren, omdat de zwaartekracht in de weg zit. Dit is weliswaar te omzeilen door te gaan experimenteren in een valtoren of in een vliegtuig dat een paraboolvlucht maakt, maar dan nog is een toestand zonder invloed van zwaartekracht slechts enkele seconden haalbaar. In de tweede plaats is het daadwerkelijk experimenteren in de ruimte vaak kostbaar en in ieder geval zeer omslachtig.

Toch komen experimenten in de ruimte voor en een bekend experiment dat in 1992 heeft plaatsgevonden is het zogenaamde Wet Satellite Model (WSM) experiment. Tijdens dit experiment werd een kleine satelliet, de zogenaamde Ejectable Ballistometer (EB), gelanceerd vanuit een MASER 5 raket en raakte vervolgens in een vrije val van 370 seconden. De EB bestond uit een cilindrische, ringvormige tank die voor de helft gevuld was met water. Tijdens de lancering werd de EB een roterende beweging om de as van de cilinder meegegeven. Uit de fysica is bekend dat een roterend lichaam evolueert naar een toestand van minimale kinetische energie. In het geval van de EB is dit een rotatie om een as loodrecht op de initiële rotatieas. Dit betekent dat enige tijd na de lancering de satelliet om al zijn assen begon te draaien en hierbij ook de aan boord zijnde vloeistof in beweging zette. Gedurende het verdere verloop van het experiment werd de rotatie om de as van de cilinder gedempt door de viskeuze vloeistof en settelde het systeem zich in een stationaire toestand. De overgang van rotatie rond de initiële as naar rotatie rond een andere as wordt een flat spin genoemd en is een bekend voorbeeld van dynamische interactie. Het WSM experiment heeft veel waardevolle data opgeleverd met betrekking tot dit complexe verschijnsel.

Halverwege de negentiende eeuw kwam er een methode bij om onderzoek te doen naar het gedrag van vloeistoffen. Rond deze tijd schreven Navier (1823) en Stokes (1845) de wiskundige vergelijkingen op die de stroming van een vloeistof beschrijven. Deze zijn gebaseerd op een aantal fysische behoudswetten, namelijk behoud van massa, impuls en energie. Ondanks de ogenschijnlijk eenvoudige vorm van de Navier-Stokes vergelijkingen, zijn ze alleen in sterk vereenvoudigde gevallen met pen en papier op te lossen.

In de twintigste eeuw is er een derde methode bijgekomen om de stroming van vloeistoffen beter te leren begrijpen, namelijk die van de numerieke simulatie. Hierbij wordt getracht de Navier-Stokes vergelijkingen, op een benaderende manier, met behulp van de computer op te lossen. Hiertoe wordt het gebied waarin de vloeistof kan stromen verdeeld in kleine cellen; hoe meer cellen des te nauwkeuriger het resultaat (en des te langer duurt de berekening). In elk van die cellen wordt een behoudswet voor massa, impuls en energie opgesteld. Die van massabehoud is eenvoudig te begrijpen: de totale hoeveelheid vloeistof die door de rand van een cel naar binnen stroomt, moet gelijk zijn aan de totale hoeveelheid vloeistof die door de rand naar buiten stroomt. Tezamen vormen alle behoudswetten voor alle cellen een groot stelsel vergelijkingen, waaruit met behulp van de computer fysische grootheden van de vloeistof, zoals druk, snelheid en temperatuur, worden opgelost.

Dit proefschrift

In dit proefschrift wordt een methode beschreven om eerder genoemde stromingsproblemen met behulp van een computer te simuleren. Het proefschrift is opgesplitst in twee delen: in het eerste deel wordt een numeriek model ontwikkeld voor de simulatie van vloeistofstromingen in een gewichtsloze omgeving en in het tweede deel wordt dit model uitgebreid voor het simuleren van dynamische interactie. De inhoud van beide delen wordt hieronder in meer detail beschreven.

Het proefschrift begint met het formuleren van de Navier-Stokes vergelijkingen. Zoals al eerder vermeld, worden deze vergelijkingen in het vakgebied van de numerieke stromingsleer benaderend opgelost door het rekengebied (het gebied waarin de vloeistof zich beweegt) op te delen in cellen. In dit proefschrift is gekozen voor kubusvormige cellen (de kubusvormige cellen vormen samen een zogenaamd Cartesisch rekenrooster). Het voordeel van kubussen is dat het rekenrooster een overzichtelijke, regelmatige ordening bezit; elke kubus is op een zelfde manier geöriënteerd en heeft evenveel buren als alle andere roostercellen. Dit maakt de boekhouding in het computerprogramma relatief eenvoudig (vergeleken met een rekenrooster dat niet netjes geordend is).

Nadeel van een Cartesisch rekenrooster is dat het niet altijd goed aansluit bij de geometrie van het rekengebied. Als het rekengebied bestaat uit ronde vormen, zoals bij brandstoftanks aan boord van satellieten vaak het geval is, dan snijdt de rand van het rekengebied schuin door de kubussen in het rekenrooster. Hiermee moet rekening worden gehouden bij het opstellen van de discrete (d.w.z. voor elke kubus) vergelijkingen voor massa-, impuls- en energiebehoud.

Een belangrijk onderdeel van het numeriek model is het bijhouden van de positie van het vloeistofoppervlak. In dit proefschrift wordt hiervoor gebruik gemaakt van de zogenaamde VOF methode (VOF is een acroniem voor Volume Of Fluid). Bij deze methode wordt in elke roostercel een getal geïntroduceerd, de zogenaamde VOF functie, dat aangeeft welke fractie van die cel gevuld is met vloeistof; 0 is helemaal leeg, 1 is helemaal vol en een waarde tussen 0 en 1 geeft aan dat de desbetreffende cel een deel van het vrije oppervlak bevat (zo'n roostercel wordt een oppervlaktecel genoemd). Het daadwerkelijke oppervlak wordt vervolgens gereconstrueerd aan de hand van de VOF functie. Dit kan heel grof door te zeggen dat in elke oppervlaktecel het vrije oppervlak evenwijdig loopt aan het Cartesisch rooster (in dit geval zijn er in elke cel dus drie mogelijke oriëntaties van het oppervlak). Deze methode is geïntroduceerd door Hirt en Nichols. Een meer verfijnde methode, de methode van Youngs, staat toe dat het oppervlak in een oppervlaktecel ook schuin ten opzichte van het rekenrooster mag staan. Echter, deze ogenschijnlijk meer nauwkeurige methode blijkt in de praktijk erg gevoelig voor kleine verstoringen in de VOF functie, zoals wordt aangetoond in dit proefschrift. Daarom is voor de methode van Hirt en Nichols gekozen om het oppervlak te traceren.

Helaas is de originele methode van Hirt en Nichols vrij onnauwkeurig, omdat er ten gevolge van numerieke fouten nogal eens vloeistof wordt verloren of gewonnen. Bovendien raken er tijdens het transporteren van het oppervlak veel druppels los van de bulk van de vloeistof. Deze problemen, het ontbreken van massabehoud en loslatende druppels, zijn opgelost door de transportmethode in de buurt van het vrije oppervlak aan te passen. Deze gemodificeerde methode van Hirt en Nichols geeft, met name in simulaties waar de vloeistof heftig klotst, aanzienlijk betere resultaten.

Het vrije oppervlak speelt nog op een andere manier een belangrijke rol in het numeriek model. In een omgeving zonder invloed van de zwaartekracht kunnen capillaire krachten aan het vloeistofoppervlak namelijk niet verwaarloosd worden. Wiskundig gezien betekenen capillaire krachten dat de druk op het vloeistofoppervlak een functie is van de kromming van dit oppervlak. Om deze kromming nauwkeurig te kunnen bepalen is een precieze beschrijving van de positie van het oppervlak vereist, waarbij in oppervlaktecellen nabij de vaste wand de contacthoek additionele informatie verschaft.

In het tweede deel van het proefschrift wordt het numeriek model uitgebreid ten behoeve van simulatie van dynamische interactie. Hiertoe moet een model worden opgesteld voor de dynamica van het rekengebied. Dit model bestaat uit vergelijkingen voor translatie en rotatie en bevat termen die de krachten ten gevolge van de klotsende vloeistof representeren. Het is vrij natuurlijk om deze vergelijkingen te formuleren in een inertiaal coördinatenstelsel (d.w.z. een stelsel dat niet met het rekengebied meebeweegt). Deze aanpak heeft echter een groot nadeel: het oplossen van de zo verkregen vergelijkingen met behulp van een computer is instabiel als de vloeistofmassa te groot is in vergelijking met de massa van het rekengebied. Daarom wordt het model voor de dynamica van het rekengebied zodanig herschreven, dat de termen ten gevolge van de klotsende vloeistof geformuleerd zijn in het coördinatenstelsel dat met het rekengebied meebeweegt. Middels deze herschrijving kunnen de vergelijkingen voor de dynamica van het rekengebied voor willekeurige massaverhoudingen numeriek worden opgelost zonder stabiliteitsproblemen.

Het model dat in het proefschrift wordt besproken, is geïmplementeerd in een computerprogramma genaamd ComFlo. Met dit programma zijn veel simulaties gedaan om te testen of alle aspecten van het model goed functioneren. Een representatief deel van deze simulaties is beschreven in dit proefschrift en uit de resultaten mag geconcludeerd worden dat ComFlo in staat is stromingen in een omgeving zonder zwaartekrachtsinvloeden, waarbij dynamische interactie een rol speelt, te simuleren. Ter afsluiting is ook de flat spin van het eerder genoemde Wet Satellite Model experiment gesimuleerd.

Epiloog

Ondanks het vermogen van ComFlo om capillaire stromingen en dynamische interactie te simuleren, blijft er nog veel onderzoek nodig in dit vakgebied. Zo zijn er nog open vragen over de fysica van de contacthoek en is het nog steeds een probleem om de positie van ruimtevaartuigen met klotsende brandstof aan boord te regelen. Numerieke simulatiemethoden, zoals beschreven in dit proefschrift, zullen een belangrijke rol spelen bij het vinden van antwoorden op deze vragen, maar ook experimenten blijven van grote waarde.

Eén experiment dat op het programma staat en nauw aansluit bij dit proefschrift is het SloshSat FLEVO experiment (FLEVO is een acroniem voor Facility for Liquid Experimentation and Verification in Orbit). SloshSat is een experimenteersatelliet, gebouwd door het Nationaal Lucht- en Ruimtevaartlaboratorium NLR, die hopelijk binnen niet al te lange tijd vanuit de Space Shuttle gelanceerd zal worden. Doel van het SloshSat experiment is het onderzoeken van het gedrag van vloeistoffen in een gewichtsloze omgeving en het bestuderen van dynamische interactie. De lancering van SloshSat stond oorspronkelijk gepland voor 1999 en de resultaten van het experiment hadden ter validatie moeten dienen voor het model dat in dit proefschrift is beschreven. Dit is door herhaald uitstel van het experiment helaas niet mogelijk gebleken, maar zeker is dat de toekomstige vlucht van SloshSat waardevolle data zal opleveren en op die manier zal bijdragen aan een beter fysisch begrip en de verdere ontwikkeling van numerieke simulatiemethoden.

Dankwoord

Al weer ruim vier maanden geleden ben ik uit Groningen vertrokken om met mijn huidige baan in Utrecht te beginnen. Maar nog steeds kom ik regelmatig in het noorden; natuurlijk omdat er nog een aantal zaken geregeld moet worden met betrekking tot mijn aanstaande promotie, maar vooral om vrienden op te zoeken die allemaal op hun eigen manier bij hebben gedragen aan de totstandkoming van dit proefschrift.

Hiervoor wil ik in de eerste plaats mijn promotor, Arthur Veldman, bedanken. Arthur, in de afgelopen jaren heb ik met ontzettend veel plezier met je samengewerkt aan de ontwikkeling van ComFlo. In die jaren hebben we heel veel gepraat; niet alleen over het onderzoek, maar ook over talloze andere onderwerpen, zoals muziek, de juiste kleur blauw voor de poster en onze vier-letter persoonlijkheden. Je deur stond altijd open; kenmerkend voor de informele, vriendschappelijke manier waarop jij met je collega's omgaat. Arthur, enorm bedankt voor je enthousiasme, het vertrouwen dat je in me had en de vrijheid die je me hebt gegeven in mijn onderzoek. Je bent voor een groot deel verantwoordelijk voor de fantastische tijd die ik in het IWI heb gehad. Ik hoop dat ik in de toekomst nog regelmatig contact met je zal hebben.

De leden van de beoordelingscommissie, Hendrik Hoogstraten, Hans Kuipers en Piet Wesseling, en de referenten, Roel Verstappen en Jan Vreeburg, wil ik bedanken voor het zorgvuldig lezen van dit proefschrift en het geven van nuttige adviezen, hetgeen heeft geleid tot een bevredigender eindresultaat.

De SRON wil ik bedanken voor het financieren van het onderzoek dat in dit proefschrift is beschreven en het J.M. Burgerscentrum bedank ik voor hun financiële bijdrage aan mijn reis naar Ljubljana en de gelegenheden om met andere promovendi in de stromingsleer kennis te maken. Het NLR wil ik bedanken voor het beschikbaar stellen van de foto die op de omslag van dit proefschrift pronkt.

Een aanzienlijk deel van mijn tijd in het IWI heb ik achter de computer gezeten. Vanzelfsprekend waren er van tijd tot tijd soft- en hardware problemen. Gelukkig kon ik hiermee altijd terecht bij Doeke de Vries, Kees Visser, Peter Arendz en Harm Paas, waarvoor ik ze hartelijk bedank. Jan Kraak wil ik bedanken voor zijn hulp bij het visualisatieproces. De TM-ers bedank ik voor de prettige sfeer in de groep en voor de gezellige (maar helaas spaarzame) uitjes. Ook wil ik op deze plek Ed van Daalen bedanken voor zijn enthousiasme en de prettige samenwerking.

In de afgelopen jaren heb ik veel samengewerkt met studenten tijdens hun stage-en/of afstudeeronderzoek. Jos Dijkstra, Erwin Loots, Bart Lamers, Bernard Bos, Geert Fekken, Johan Bos, Theresa Kleefsman, Simone De Kleermaeker, Bram de Bruin, Joris Steneker en Norbert Douchin, ik wil jullie bedanken voor de prettige samenwerking en de bijdrage die jullie hebben geleverd aan de ontwikkeling van ComFlo. Erwin, Geert en Theresa, veel succes met (de afronding van) jullie promotieonderzoek.

De sfeer onder de AIO's is mij de afgelopen jaren bijzonder goed bevallen. Niet alleen maakten we onze dagelijkse gang naar de ACLO voor de lunch en dronken we koffie/thee in de RC-kantine. We gingen ook (in steeds wisselende samenstelling) eten bij de ACLO, squashen, tennissen, fitnessen, hardlopen, kanoën, karten, uit eten, naar de film, Catan en andere bordspellen spelen en op vakantie. Ena (ik mis je als kamergenoot), Erwin, Geert, Edith, Barteld, Joost, Theresa, Conny, Marc, Martijn, Jasper, Maint, Robert, Jun, Wim, Robert, Gert-Jan en Roland, bedankt voor jullie vriendschap en alle leuke dingen die we hebben gedaan. Jullie hebben mijn leven enorm verrijkt en ik hoop dat jullie dat blijven doen, ook al zien we elkaar nu wat minder vaak.

Tenslotte bedank ik mijn ouders Teun en Roelie, mijn broer René, en Joop en Truus voor alle steun, hulp en liefde die ze mij altijd hebben gegeven.

Jeroen Gerrits
Utrecht, 21 oktober 2001