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.

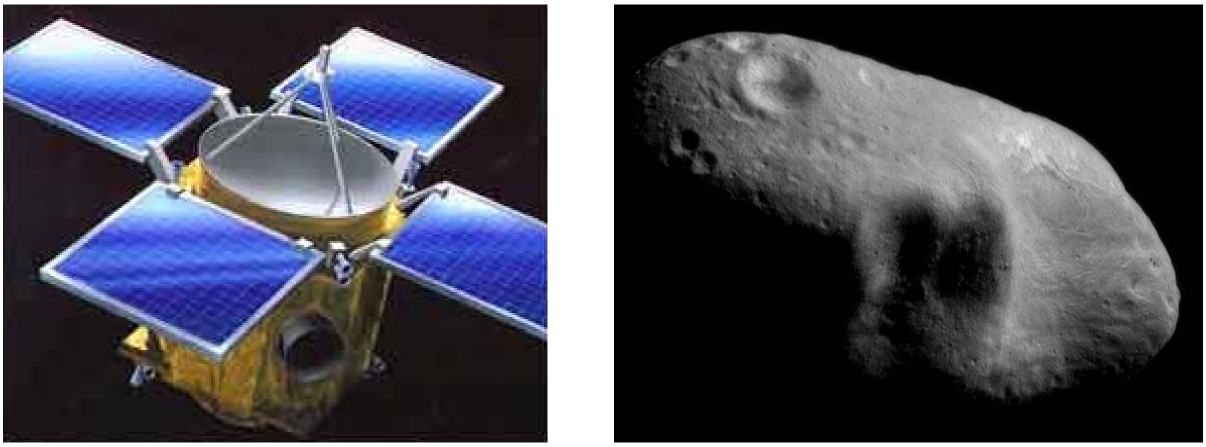

Figure 1.1: *Left: Mock-up of the Near Earth Asteroid Rendezvous (NEAR) spacecraft. Right: Picture of asteroid 433 Eros taken by the NEAR spacecraft. These pictures were taken from http://www.space.com/php/multimedia/imagegallery/archive.php and http://antwrp.gsfc.nasa.gov/apod/ap010211.html respectively.*

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

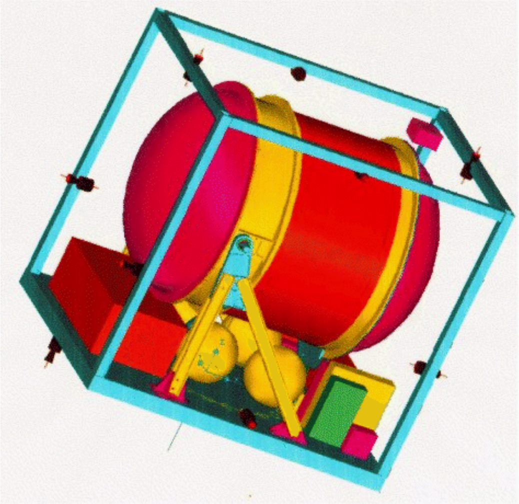

Figure 1.2: *Mock-up of the experiment satellite SloshSat FLEVO.*

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,

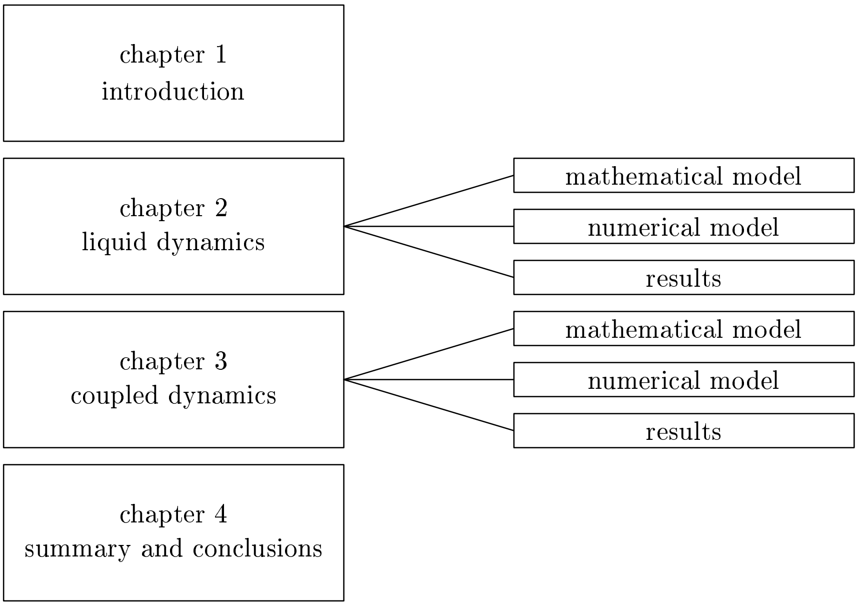

Figure 1.3: *Outline of thesis.*

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.