Direct Numerical Simulations of Gas-Liquid Multiphase Flows

Grétar Tryggvason, Ruben Scardovelli and Stéphane Zaleski
Gas-liquid multiphase flows play an essential role in the workings of Nature and the enterprises of mankind. Our everyday encounter with liquids is nearly always at a free surface, such as when drinking, washing, rinsing and cooking. Similarly, such flows are in abundance in industrial applications: heat transfer by boiling is the preferred mode in both conventional and nuclear power plants and bubble driven circulation systems are used in metal processing operations such as steel making, ladle metallurgy and the secondary refining of aluminum and copper. A significant fraction of the energy needs of mankind is met by burning liquid fuel and a liquid must evaporate before it burns. In almost all cases the liquid is therefore atomized to generate a large number of small droplets and hence a large surface area. Indeed, except for drag (including pressure drops in pipes) and mixing of gaseous fuels we would not be far off to assert that nearly all industrial applications of fluids involve a multiphase flow of one sort or another. Sometimes, one of the phases is a solid, such as in slurries and fluidized beds, but in a large number of applications one phase is a liquid and the other is a gas. Of natural gas-liquid multiphase flows, rain is perhaps the experience that first comes to mind, but bubbles and droplets play a major role in the exchange of heat and mass between oceans and the atmosphere and in volcanic explosions. Living organisms are essentially large and complex multiphase systems.

Understanding the dynamics of gas-liquid multiphase flows is of critical engineering and scientific importance and the literature is extensive. From a mathematical point of view multiphase flow problems are notoriously difficult and much of what we know has been obtained by experimentation and scaling analysis. Not only are the equations, governing the fluid flow in both phases, highly nonlinear, but the position of the phase boundary must generally be found as a part of the solution. Exact analytical solutions therefore
exist only for the simplest problems such as the steady-state motion of bubbles and droplets in Stokes flow, linear inviscid waves and small oscillations of bubbles and droplets. Experimental studies of multiphase flows are not easy either. For many flows of practical interest the length scales are small, the time scales are short and optical access to much of the flow is limited. The need for numerical solutions of the governing equations has therefore been felt by the multiphase research community since the origin of computational fluid dynamics, in the late fifties and early sixties. Although much has been accomplished, simulations of multiphase flows have remained far behind homogeneous flows where direct numerical simulations (DNS)—where the governing equations are solved using sufficiently fine grids so that all continuum time- and length-scales are fully resolved—have become a standard tool in turbulence research. While this is not surprising, considering the added difficulty, the situation is certainly not due to lack of effort. However, in the last decade and a half or so, these efforts have started to pay off and rather significant progress has been accomplished on many fronts. It is now possible to do DNS for a large number of fairly complex systems and

Fig. 1.1. A picture of many buoyant bubbles rising in an otherwise quiescent liquid pool. The average bubble diameter is about 2.2 mm and the void fraction is approximately 0.75%. From Bröder and Sommerfeld (2007). Reproduced with permission.
Examples of multiphase flows

Since this is a book about numerical simulations, it seems appropriate to start by showing a few “real” systems. The following examples are picked somewhat randomly, but give some insight into the kind of systems that can be examined by direct numerical simulations.

Bubbles are found in a large number of industrial applications. For example they carry vapor away from hot surfaces in boiling heat transfer, disperse gasses and provide stirring in various chemical processing systems and also affect the propagation of sound in the ocean. To design systems that involve bubbly flows it is necessary to understand how the collective rise velocity of many bubbles depends on the void fraction and the bubble size distribution, how bubbles disperse and how they stir up the fluid. Figure 1.1 is a picture of air bubbles rising through water in a small bubble column. The average bubble diameter is about 2.2 mm and the void fraction is approximately 0.75%. At these parameters the bubbles rise with an average velocity of
Introduction

roughly 0.27 m/s, but since the bubbles are not all of the same size they will generally rise with different velocities.

To generate sprays for combustion, coating and painting, irrigation, humidification and a large number of other applications a liquid jet must be atomized. Predicting the rate of atomization and the resulting droplet size distribution, as well as their velocity, is critical to the successful design of such processes. In Fig. 1.2, a liquid jet is ejected from a nozzle of diameter 8 mm with a velocity of 0.6 m/s. To accelerate its breakup, the jet is injected into a co-flowing air stream, with a velocity of 35 m/s. Initially, the shear between the air and the liquid leads to large axisymmetric waves but as the waves move downstream the air pulls long filaments from the crest of the wave. The filaments then break into droplets by a capillary instability. See Marmottant and Villermaux (2001) and Villermaux et al. (2004) for details.

Droplets impacting solid or liquid surfaces generally splash, often disrupting the surface significantly. Rain droplets falling on the ground often result in soil erosion, for example. But droplet impact can also help to increase the heat transfer, such as in quenching and spray cooling, and rain often greatly enhances the mixing at the ocean surface. Figure 1.3 shows the splash created when a droplet of a diameter of about three millimeters, released from
Examples of multiphase flows

Fig. 1.4. Massive cavitation near the leading edge of an airfoil. The flow is from the left to right. From Kermeeen (1956). Reproduced with permission.

nearly half a meter above the surface, impacts a liquid layer a little over a droplet-diameter deep. The impact of the droplet creates a liquid crater and a rim that often breaks into droplets. As the crater collapses, air bubbles are sometimes trapped in the liquid.

While bubbles are often generated by air injection into a pool of liquid or are formed by entrainment at a free surface, such as when waves break, they also frequently form when a liquid changes phase into vapor. Such a phase change is often nucleated at a solid surface and can take place either by heating the liquid above the saturation temperature, as in boiling, or by lowering the pressure below the vapor pressure, as in cavitation. Figure 1.4 shows massive cavitation at the leading edge of an airfoil submerged in water. The chord of the airfoil is 7.6 cm, the flow speed is 13.7 m/s from left to right, and the increase in the liquid velocity as it passes over the leading edge of the airfoil leads to a drop in pressure that is sufficiently large so that the liquid “boils”. As the vapor bubbles move into regions of higher pressure at the back of the airfoil, they collapse. However, residual gases, dissolved in the liquid, diffuse into the bubbles during their existence, leaving traces that are visible after the vapor has condensed.

In many multiphase systems one phase is a solid. Suspensions of solid particles in liquids or gases are common and the definition of multiphase flows is sometimes extended to cover flows through or over complex stationary solids, such as packed beds, porous media, forests and cities. The main
difference between gas-liquid multiphase flows and solid-gas and solid-liquid multiphase flows is usually that the interface maintains its shape in the latter cases, even though the location of the solid may change. In some instances, however, that is not the case. Flexible solids can change their shape in response to fluid flow and during solidification or erosion the boundary can evolve, sometimes into shapes that are just as convoluted as encountered for gas-liquid systems. When a metal alloy solidifies, the solute is initially rejected by the solid phase. This leads to constitutional undercooling and an instability of the solidification front. The solute-rich phase eventually solidifies, but with a very different composition than the material that first became solid. The size, shape and composition of the resulting microstructures determine the properties of the material and those are usually sensitively dependent on the various process parameters. A representative micrograph of an Al-Si alloy prepared by metallographic techniques and etching to reveal phase boundaries and interfaces is shown in Fig. 1.5. The light gray phase is almost pure aluminum and solidifies first, but constitutional undercooling leads to dendritic structures of a size measured in few tens of micrometers.

Living systems provide an abundance of multiphase flow examples. Suspended blood cells and aerosol in pulmonary flow are obvious examples at the “body” scale, as are the motion of organs and even complete individu-
als. But even more complex systems, such as the motion of a flock of birds through air and a school of fish through water, are also multiphase flows. Figure 1.6 show a large number of yellow-tailed goatfish swimming together and coordinating their movement. An understanding of the motion of both a single fish as well as the collective motion of a large school may have implication for population control and harvesting, as well as the construction of mechanical swimming and flying devices.

1.2 Computational modeling

Computations of multi-fluid (two different fluids) and multiphase (same fluid, different phases) flows are nearly as old as computations of constant-density flows. As for such flows, a number of different approaches have been tried and a number of simplifications used. In this section we will attempt to give a brief but comprehensive overview of the major efforts to simulate multi-fluid flows. We make no attempt to cite every paper, but hope to mention all major developments.
1.2.1 Simple flows ($Re=0$ and $Re=\infty$)

In the limit of either very large or very small viscosity (as measured by the Reynolds number, see Chapter 2.2.6), it is sometimes possible to simplify considerably the flow description by either ignoring inertia completely (Stokes flow) or by ignoring viscous effects completely (inviscid flow). For inviscid flows it is usually further necessary to assume that the flow is irrotational, except at fluid interfaces. Most success has been achieved for disperse flows of undeformable spheres where, in both these limits, it is possible to reduce the governing equations to a system of coupled ordinary differential equations (ODEs) for the particle positions. For Stokes flow the main developer was Brady and his collaborators (see Brady and Bossis, 1988, for a review of early work) who have investigated extensively the properties of suspensions of particles in shear flows, among other problems. For inviscid flows, Sangani and Didwania (1993) and Smereka (1993) simulated the motions of spherical bubbles in a periodic box and observed that the bubbles tended to form horizontal clusters, particularly when the variance of the bubble velocity was small.

For both Stokes flows and inviscid potential flows, problems with deformable boundaries can be simulated with boundary integral techniques. One of the earliest attempts was due to Birkhoff (1954) where the evolution of the interface between a heavy fluid initially on top of a lighter one (the Rayleigh-Taylor instability) was followed by a method tracking the interface between two inviscid and irrotational fluids. Both the method and the problem later became a staple of multiphase flow simulations. A boundary integral method for water waves was presented by Longuet-Higgins and Cokelet (1976) and used to examine breaking waves. This paper had enormous influence and was followed by a large number of very successful extensions and applications, particularly for water waves (Baker et al., 1982; Vinje and Brevig, 1981; Schultz et al., 1994, and others). Other applications include the evolution of the Rayleigh-Taylor instability (Baker et al., 1980), the growth and collapse of cavitation bubbles (Blake and Gibson, 1981; Robinson et al., 2001), the generation of bubbles and droplets due to the coalescence of bubbles with a free surface (Oguz and Prosperetti, 1990; Boulton-Stone and Blake, 1993), the formation of bubbles and droplets from an orifice (Oguz and Prosperetti, 1993) and the interactions of vortical flows with a free surface (Yu and Tryggvason, 1990), just to name a few. All boundary integral (or boundary element, when the integration is element based) methods for inviscid flows are based on following the evolution of the strength of surface singularities in time by integrating a Bernoulli-type equation. The surface
1.2 Computational modeling

![Image of droplet breakup simulation](image)

Fig. 1.7. A Stokes flow simulation of the breakup of a droplet in a linear shear flow. The barely visible line behind the numerical results is the outline of a droplet traced from an experimental photograph. Reprinted with permission from Cristini et al. (1998). Copyright 2005, American Institute of Physics.

singularities give one velocity component and Green’s second theorem yields the other, thus allowing the position of the surface to be advanced in time. Different surface singularities allow for a large number of different methods (some that can only deal with a free surface and others that are suited for two-fluid problems) and different implementations multiply the possibilities even further. For an extensive discussion and recent progress see Hou et al. (2001). Although continuous improvements are being made and new applications continue to appear, two-dimensional boundary integral techniques for inviscid flows are by now—more than thirty years after the publication of the paper by Longuett-Higgins and Cokelet—a fairly mature technology. Fully three-dimensional computations are, however, still rare. Chahine and Duraiswami (1992) computed the interactions of a few inviscid cavitation bubbles and Xue et al. (2001) have simulated a three-dimensional breaking wave. While the potential flow assumption has lead to many spectacular successes, particularly for short-time transient flows, its inherent limitations are many. The lack of a small-scale dissipative mechanism makes those models susceptible to singularity formation and the absence of dissipation usually makes them unsuitable for the predictions of the long-time evolution of any system.
Introduction

The key to the reformulation of inviscid interface problems with irrotational flow in terms of a boundary integral is the linearity of the potential equation. In the opposite limit, where inertia effects can be ignored and the flow is dominated by viscous dissipation, the Navier-Stokes equations become linear (it is the so-called Stokes flow limit) and it is also possible to recast the governing equations into an integral equation on a moving surface. Boundary integral simulations of unsteady two-fluid Stokes problems originated with Youngren and Acrivos (1976) and Rallison and Acrivos (1978), who simulated the deformation of a bubble and a droplet, respectively, in an extensional flow. Subsequently, several authors have investigated a number of problems. Pozrikidis and collaborators have examined several aspects of suspensions of droplets, starting with a study by Zhou and Pozrikidis (1993) of the suspension of a few two-dimensional droplets in a channel. Simulations of fully three-dimensional suspensions have been done by Loewenber and Hinch (1996) and Zinchenko and Davis (2000). The method has been described in detail in the book by Pozrikidis (1992), and Pozrikidis (2001) gives a very complete summary of the various applications. An example of a computation of the breakup of a very viscous droplet in a linear shear flow, using a method that adaptively refines the surface grid as the droplet deforms is shown in Fig. 1.7.

In addition to inviscid flows and Stokes flows, boundary integral methods have been used by a number of authors to examine two-dimensional, two-fluid flows in Hele-Shaw cells. Although the flow is completely viscous, away from the interface it is a potential flow. The interface can be represented by the singularities used for inviscid flows (de Josselin de Jong, 1960) but the evolution equation for the singularity strength is different. This was used by Tryggvason and Aref (1983) and Tryggvason and Aref (1985) to examine the Saffman-Taylor instability, where an interface separating two fluids of different viscosity deforms if the less viscous fluid is displacing the more viscous one. They used a fixed grid to solve for the normal velocity component (instead of Green’s theorem), but Green’s theorem was subsequently used by several authors to develop boundary integral methods for interfaces in Hele-Shaw cells. See, for example, DeGregoria and Schwartz (1985), Meiburg and Homsy (1988) and the review by Hou et al. (2001).

Under the heading of simple flows we should also mention simulations of the motion of solid particles, in the limit where the fluid motion can be neglected and the dynamics is governed only by the inertia of the particles. Several authors have followed the motion of a large number of particles that interact only when they collide with each other. Here, it is also sufficient to solve a system of ODEs for the particle motion. Simulations of this
kind are usually called “granular dynamics.” For an early discussion see Louge (1994) and a more recent one can be found in Pöschel and Schwage (2005), for example. While these methods have been enormously successful in simulating certain types of solid-gas multiphase flows, they are limited to a very small class of problems. One could, however, argue that simulations of the motion of particles interacting through a potential, such as simulations of the gravitational interactions of planets or galaxies and molecular dynamics, also fall into this class. Discussing such methods and their applications would enlarge the scope of the present work enormously and we will confine our coverage by simply suggesting that the interested reader consults the appropriate references, such as Schlick (2002) for molecular simulations and Hockney and Eastwood (1981) for astrophysical and other systems.

\section*{1.2 Computational modeling}

\subsection*{1.2.2 Finite Reynolds number flows}

For intermediate Reynolds numbers it is necessary to solve the full Navier-Stokes equations. Nearly ten years after Birkhoff’s effort to simulate the inviscid Rayleigh-Taylor problem by a boundary integral technique, the Marker-and-Cell (MAC) method was developed at Los Alamos by Harlow and collaborators. In the MAC method the fluid is identified by marker particles distributed throughout the fluid region and the governing equations solved on a regular grid that covers both the fluid-filled and the empty part of the domain. The method was introduced in Harlow and Welch (1965)
Introduction

and two sample computations of the so-called dam breaking problem were shown in that first paper. Several papers quickly followed: Harlow and Welch (1966) examined the Rayleigh-Taylor problem (Fig. 1.8) and Harlow and Shannon (1967) studied the splash when a droplet hits a liquid surface. As originally implemented, the MAC method assumed a free surface so there was only one fluid involved. This required boundary conditions to be applied at the surface and the fluid in the rest of the domain to be completely passive. The Los Alamos group realized, however, that the same methodology could be applied to two-fluid problems. Daly (1969b) computed the evolution of the Rayleigh-Taylor instability for finite density ratios and Daly and Pracht (1968) examined the initial motion of density currents. Surface tension was then added by Daly (1969a) and the method again used to examine the Rayleigh-Taylor instability. The MAC method quickly attracted a small group of followers that used it to study several problems: Chan and Street (1970) applied it to free surface waves, Foote (1973) and Foote (1975) simulated the oscillations of an axisymmetric droplet and the collision of a droplet with a rigid wall, respectively, and Chapman and Plesset (1972) and Mitchell and Hammitt (1973) followed the collapse of a cavitation bubble. While the Los Alamos group did a number of computations of various problems in the sixties and early seventies and Harlow described the basic idea in a Scientific American article (Harlow and Fromm, 1965), the enormous potential of this newfound tool did not, for the most part, capture the fancy of the fluid mechanics research community. Although the MAC method was designed specifically for multifluid problems (hence the M for Markers!) it was also the first method to successfully solve the Navier-Stokes equation using the primitive variables (velocity and pressure). The staggered grid used was a novelty and today it is a common practice to refer to any method using a projection based time integration on a staggered grid as a MAC method (see Chapter 3).

The next generation of methods for multifluid flows evolved gradually from the MAC method. It was already clear in the Harlow and Welch (1965) paper that the marker particles could cause inaccuracies and of the many algorithmic ideas explored by the Los Alamos group, the replacement of the particles by a marker function soon became the most popular alternative. Thus the Volume-of-Fluid (VOF) method was born. VOF was first discussed in a refereed journal article by Hirt and Nichols (1981) but the method originated earlier (DeBar, 1974; Noh and Woodward, 1976). The basic problem with advecting a marker function is the numerical diffusion resulting from working with a cell-averaged marker function (see Chapter 4). To prevent the marker function from continuing to diffuse, the inter-
face is “reconstructed” in the VOF method in such a way that the marker does not start to flow into a new cell until the current cell is full. The one-dimensional implementation of this idea is essentially trivial and in the early implementation of VOF, the interface in each cell was simply assumed to be a vertical plane for advection in the horizontal direction and a horizontal plane for advection in the vertical direction. This relatively crude reconstruction often lead to large amount of “floatsam and jetsam” (small unphysical droplets that break away from the interface) that degraded the accuracy of the computation. To improve the representation, Youngs (1982) and Ashgriz and Poo (1991) and others introduced more complex reconstructions of the interface, representing it with a line (two dimensions) or a plane (three dimensions) that could be oriented arbitrarily in such a way as to best fit the interface. This increased the complexity of the method considerably but resulted in greatly improved advection of the marker function. Even with higher-order representation of the fluid interface in each cell, the accurate computation of surface tension remained a major problem. In his simulations of surface tension effects on the Rayleigh-Taylor instability, using the MAC method, Daly (1969b) introduced explicit surface markers for this purpose. However, the premise behind the development of the VOF method was to get away from using any kind of surface marker so that the surface tension had to be obtained from the marker function instead. This was achieved by Brackbill et al. (1992) who showed that the curvature (and hence surface tension) could be computed by taking the discrete divergence of the marker function. A “conservative” version of this “continuum surface force” method was developed by Lafaurie et al. (1994). The VOF method has been extended in various ways by a number of authors. In addition to better ways to reconstruct the interface (Rider and Kothe, 1998; Scardovelli and Zaleski, 2000; Aulisa et al., 2007) and compute the surface tension (Renardy and Renardy, 2002; Popinet, 2009), more advanced advection schemes for the momentum equation and better solvers for the pressure equation have been introduced (see Rudman, 1997, for example). Other refinement include the use of sub-cells to keep the interface as sharp as possible (Chen et al., 1997a). VOF methods are in widespread use today and many commercial codes include VOF to track interfaces and free surfaces. Figure 1.9 shows one example of a computation of the splash made when a liquid droplet hits a free surface, done by a modern VOF method. We will discuss the use of VOF extensively in later Chapters.

The basic ideas behind the MAC and the VOF methods gave rise to several new approaches in the early nineties. Unverdi and Tryggvason (1992) introduced a Front-Tracking method for multifluid flows where the interface
was marked by connected marker points. The markers are used to advect the material properties (such as density and viscosity) and to compute surface tension, but the rest of computations is done on a fixed grid as in the VOF method. Although using connected markers to update the material function was new, marker particles had already been used by Daly (1969a) who used them to evaluate surface tension in simulations with the MAC method, in the Immersed-Boundary Method of Peskin (1977) for one-dimensional elastic fibers in homogeneous viscous fluids and in the Vortex-in-Cell method of Tryggvason and Aref (1983) for two-fluid interfaces in a Hele-Shaw cell, for example. The Front-Tracking method of Unverdi and Tryggvason (1992) has been very successful for simulations of finite Reynolds number flows of immiscible fluids and Tryggvason and collaborators have used it to explore a large number of problems.

The early nineties also saw the introduction of the Level-Set, the CIP, and the Phase-Field methods to track fluid interfaces on stationary grids. The Level-Set method was introduced by Osher and Sethian (1988), but its first use to track fluid interfaces appears to be in the work of Sussman et al. (1994) and Chang et al. (1996) who used it to simulate the rise of bubbles and the fall of droplets in two-dimensions. An axisymmetric version was used subsequently by Sussman and Smereka (1997) to examine the behavior of bubbles and droplets. Unlike the VOF method, where a discontinuous marker function is advected with the flow, in the Level-Set method a continuous level-set function is used. The interface is then identified with the zero
1.2 Computational modeling

contour of the level-set function. To reconstruct the material properties of the flow (density and viscosity, for example) a marker function is constructed from the level-set function. The marker function is given a smooth transition zone from one fluid to the next, thus increasing the regularity of the interface over the VOF method where the interface is confined to only one grid space. However, this mapping from the level-set function to the marker function requires the level-set function to maintain the same shape near the interface and to deal with this problem, Sussman et al. (1994) introduced a reinitialization procedure where the level-set function is adjusted in such a way that its value is equal to the shortest distance to the interface at all times. This step was critical in making level-sets work for fluid-dynamics simulations. Surface tension is found in the same way as in the continuous surface force technique introduced for VOF methods by Brackbill et al. (1992). The early implementation of the Level-Set method did not conserve mass very well and a number of improvements and extension followed its original introduction. Sussman et al. (1998) and Sussman and Fatemi (1999) introduced ways to improved mass conservation, Sussman et al. (1999) coupled level-set tracking with adaptive grid refinement and a hybrid VOF/Level-Set method was developed by Sussman and Puckett (2000), for example.

The Constrained Interpolated Propagation (CIP) method introduced by Takewaki et al. (1985) has been particularly popular with Japanese authors who have applied it to a wide variety of multiphase problems. In the CIP method the transition from one fluid to another is described by a cubic polynomial. Both the marker function and its derivative are then updated to advect the interface. In addition to simulating two-fluid problems, the method has been used for a number of more complex applications, such as those involving floating solids, see Yabe et al. (2001).

In the Phase-Field method the governing equations are modified in such a way that the dynamics of the smoothed region between the different fluids is described in a thermodynamically consistent way. In actual implementations the thickness of the transition is, however, much larger than it is in real systems and the net effect of the modification is to provide an “antidiffusive” term that keeps the interface reasonably sharp. While superficially there are considerable similarities between Phase-Field and Level-Set methods, the fundamental ideas behind the methods are very different. In the Level-Set method the smoothness of the phase boundary is completely artificial and introduced for numerical reasons only. In Phase-Field methods, on the other hand, the transition zone is real, although it is made much thicker than it should be for numerical reasons. It is not clear, at the time of this writing, whether keeping the correct thermodynamic conditions in an
artificially thick interface has any advantages over methods that start with a completely sharp interface. The key drawback seems to be that since the propagation and properties of the interface depend sensitively on the dynamics in the transition zone, it must be well resolved. For the motion of two immiscible fluids, that are well described by assuming a sharp interface, this adds a resolution requirement that is more stringent than for other “one-fluid” methods. The phase-field approach was originally introduced to model solidification (see Kobayashi, 1992, 1993) and has found widespread use in such simulations. With the exception of the modeling of solidification in the presence of flows (Beckermann et al., 1999; Tonhardt and Amberg, 1998), its use for fluid dynamic simulations is relatively limited (Jacqmin, 1999; Jamet et al., 2001). The main appeal of the Phase-Field methods appears to be for problems where small-scale physics must be accounted for and it is difficult to do so in the sharp interface limit.

In the “one-fluid” methods described above, where a single set of governing equations is used to describe the fluid motion in both fluids, the fluid motion is computed on regular structured grids and the main difference between the various methods is how a marker function is advected (and how surface tension is found). The thickness of the interface varies from one cell in VOF methods to a few cells in Level-Set and Front-Tracking methods, but once the marker function has been found, the specific scheme for the interface advection is essentially irrelevant for the rest of the computations. While these methods have been enormously successful, their accuracy is generally somewhat limited. There have therefore recently been several attempts to generate methods that retain most of the advantages of these methods but treat the interface as “fully sharp.” The origin of these attempts can be traced to the work of Glimm and collaborators (Glimm et al., 1981; Glimm and McBryan, 1985; Chern et al., 1986), who used grids that were modified locally near an interface in such a way that the interface coincided with a cell boundary, and more recent “cut-cell” methods for the inclusion of complex bodies in simulations of inviscid flows (Quirk, 1994; Powell, 1998). In their modern incarnation, sharp interface methods include the Ghost Fluid method, the Immersed-Interface method and the method of Udaykumar et al. (2001). In the “ghost fluid” method introduced by Fedkiw et al. (1999) the interface is marked by advecting a level-set function, but to find numerical approximations to derivatives near the interface, where the finite difference stencil involves values from the other side of the interface, fictitious values are assigned to those grid points. The values are obtained by extrapolation and a few different possibilities for doing so are discussed by Glimm et al. (2001), for example. The “Immersed-Interface” method
of Lee and LeVeque (2003) is, on the other hand, based on modifying the numerical approximations near the interface by explicitly incorporating the jump across the interface into the finite difference equations. While this is easily done for relatively simple jump conditions, it becomes more involved for complex situations. Lee and LeVeque (2003) thus found it necessary to limit their development to fluids with the same viscosity. In the method of Udaykumar et al. (2001) complex solid boundaries are represented on a regular grid by merging cells near the interface and using polynomial fitting to find field values at the interface. This method, which is related to the “cut-cell” methods used for inviscid compressible flows (Powell, 1998) has so far only been implemented for solids and fluids, including solidification (Yang and Udaykumar, 2005), but there seems to be no reason why the method can not be used for multifluid problems. For an extension to three dimensions, see Marella et al. (2005).

While the original “one-fluid” methods require essentially no modification of the flow solver near the interface (except allowing for variable density and viscosity), the sharp interface methods all require localized modifications of the basic scheme. This results in considerably more complex numerical schemes, but is also likely to improve the accuracy. That may be important for extreme values of the governing parameters, such as large differences between the material properties of the different fluids and low viscosities. The sharp interface approach may also be required for flows with very complex interface physics. However, methods based on a straightforward implementation of the “one-fluid” formulation of the governing equations, coupled with advanced schemes to advect the interface (or marker function) have already demonstrated their usefulness for a large range of problems and it is likely that their simplicity will ensure that they will continue to be widely used.

In addition to the development of more accurate implementations of the “one-fluid” approach, many investigators have pursued extension of the basic schemes to problems that are more complex than the flow of two immiscible liquids. More complex physics has been incorporated to simulate contaminated interfaces, mass transfer and chemical reactions, electrorheological effects, boiling, solidification, as well as the interaction of solid bodies with a free surface or a fluid interface. We will briefly review such advanced applications at the very end of the book, in Chapter 11.

While methods based on the “one-fluid” approach were being developed, other techniques were also explored. Hirt et al. (1970) describe one of the earliest use of structured, boundary-fitted Lagrangian grids. In this approach a logically rectangular structured grid is used, but the grid points
move with the fluid velocity, thus deforming the grid. This approach is particularly well-suited when the interface topology is relatively simple and no unexpected interface configurations develop. In a related approach, a grid line is aligned with the fluid interface, but the grid away from the interface is generated using standard grid generation techniques such as conformal mapping or other more advanced elliptic grid generation schemes. The method was used by Ryskin and Leal (1984) to compute the steady rise of buoyant, deformable, axisymmetric bubbles. They assumed that the fluid inside the bubble could be neglected, but Dandy and Leal (1989) and Kang and Leal (1987) extended the method to two-fluid problems and unsteady flows. Several authors have used this approach to examine relatively simple problems such as the steady state motion of single particles or moderate deformation of free surfaces. Fully three-dimensional simulations are relatively rare (see though Takagi et al., 1997) and it is probably fair to say that it is unlikely that this approach will be the method of choice for very complex problems such as the three-dimensional unsteady motion of several particles.

A much more general approach to continuously represent a fluid interface by a grid line is to use fully unstructured grids. This allows grid points to be inserted and deleted as needed and distorted grid cells to be reshaped. While the grid was moved with the fluid velocity in some of the early applications of this method, the more modern approach is to either move only the interface points or to move the interior nodes with a velocity different from the fluid velocity, in such a way that the grid distortion is reduced but adequate resolution is still maintained. A large number of methods have

Fig. 1.10. The interaction of two falling spheres. The spheres are shown at four different times, going from left to right. Reprinted from Hu et al. (2001), with permission from Elsevier.
been developed that fall into this general category, but we will only reference a few examples. Oran and Boris (1987) simulated the breakup of a two-dimensional drop; Shopov et al. (1990) examined the initial deformation of a buoyant bubble; Feng et al. (1994), Feng et al. (1995) and Hu (1996) computed the unsteady two-dimensional motion of several particles and Fukai et al. (1995) followed the collision of a single axisymmetric droplet with a wall. Although this appears to be a fairly complex approach, Johnson and Tezduyar (1997) and Hu et al. (2001) have produced very impressive results for the three-dimensional unsteady motion of many spherical particles. Figure 1.10 shows an example of a simulation done using the Arbitrary-Lagrangian-Eulerian method of Hu et al. (2001). Here, two solid spheres are initially falling in-line (left frame). Since the trailing sphere is sheltered from the flow by the leading one, it catches up and “kisses” the leading one. The in-line configuration is unstable and the spheres “tumble” (two middle frames). After tumbling the spheres drift apart (right frame).

The most recent addition to the collection of methods to simulate finite Reynolds number multiphase flows is the Lattice-Boltzmann Method (LBM). It is now clear that LBM can be used to obtain results of accuracy comparable to more conventional methods. It is still not clear, however, whether the LBM is significantly faster or simpler than other methods (as sometimes claimed), but most likely these methods are here to stay. For a discussion see, for example, Shan and Chen (1993) and Sankaranarayanan et al. (2002). A comparison of results obtained by the LBM method and the Front-Tracking method of Unverdi and Tryggvason (1992) can be found in Sankaranarayanan et al. (2003). We will not discuss LBM in this book, but refer the reader to Rothman and Zaleski (1997) and Chapter 6 in Prosperetti and Tryggvason (2007).

1.3 Looking Ahead

Direct numerical simulations of multiphase flows have come a long way in the last decade and a half or so. It is now possible to simulate accurately the evolution of disperse flows of several hundred bubbles, droplets and particles for sufficiently long times so that reliable values can be obtained for various statistical quantities. Similarly, major progress has been achieved in the development of methods for more complex flows, including those where a liquid solidifies or evaporates. Simulations of large systems undergoing boiling and solidification are therefore within reach.

Much remains to be done, however, and it is probably fair to say that the use of direct numerical simulations of multiphase flows for research and
design is still in the embryonic state. The possibility of computing the evolution of complex multiphase flows—such as churn-turbulent bubbly flow undergoing boiling, or the breakup of a jet into evaporating droplets—will transform our understanding of flows of enormous economic significance. Currently, control of most multiphase flow processes is fairly rudimentary and almost exclusively based on intuition and empirical observations. Industries that deal primarily with multiphase flows are, however, multibillion dollar operations and the savings realized if atomizers for spray generation, bubble injectors in bubble columns and inserts into pipes to break up droplets, just to name a few examples, could be improved by just a little bit would add up to a substantial amount of money. Reliable predictions would also reduce the design cost significantly for situations such as space vehicles and habitats where experimental investigations are expensive. And, as the possibilities of manipulating flows at the very smallest scales by either stationary or free flowing MEMS devices become more realistic, the need to predict the effect of such manipulations becomes critical.

While speculating about the long term impact of any new technology is a dangerous thing—and we will simply state that the impact of direct numerical simulations of multiphase flows will without doubt be significant—it is easier to predict the near future. Apart from the obvious prediction that computers will continue to become faster and more available, we expect that the development of numerical methods will focus mainly on flows with complex physics. Although some progress has already been achieved for flows with variable surface tension, flows coupled to temperature and electric fields and flows with phase change, simulations of such systems are still far from being commonplace. In addition to the need to solve a large number of equations, coupled systems generally possess much larger ranges of length and time scales than simple two-fluid systems. Thus, the incorporation of implicit time-integrators for stiff systems and adaptive gridding will become even more important. It is also likely, as more and more complex problems are dealt with, that the differences between direct numerical simulations—where everything is resolved fully—and simulations where the smallest scales are modeled will become blurred. Simulations of atomization where the evolution of thin films are computed by “subgrid” models and very small droplets are included as point particles are relatively obvious examples of such simulations (for a discussion of the point-particle approximation see Chapter 9 in Prosperetti and Tryggvason, 2007, for example). Other examples include possible couplings of continuum approaches as those described in this Book with microscopic simulations of moving contact lines, kinetics effects at a solidifying interface, and reactions in thin flames. Simulations of
non-Newtonian fluids, where the microstructure has to be modeled in such a way that the molecular structure is accounted for in some way, also fall under this category.

In addition to the development of more powerful numerical methods, it is increasingly critical to deal with the “human” aspect of large-scale numerical simulations. The physical problems that we must deal with and the computational tools that are available are rapidly becoming very complex. The difficulty of developing fully parallelized software to solve the continuum equations (fluid flow, mass and heat transfer, etc), where three-dimensional interfaces must be handled and the grids must be dynamically adapted, are putting such simulations beyond the reach of a single graduate student. In the future these simulations may even be beyond the capacity of small research groups. It is becoming very difficult for a graduate student to learn everything that he or she needs to know and make significant new progress in four to five years. Lowering the “knowledge barrier” and ensuring that new investigators can enter the field of direct numerical simulations of multiphase flow may well become as important as improving the efficiency and accuracy of the numerical methods. The present book is an attempt to ease the entry of new researchers into this field.
Fluid mechanics with interfaces

The equations governing multiphase flows, where a sharp interface separates immiscible fluids or phases, are presented in this Chapter. We first derive the equations for flows without interfaces, in a relatively standard manner. Then we discuss the mathematical representation of a moving interface and the appropriate jump conditions needed to couple the equations across the interfaces. Finally, we introduce the so-called “one-fluid” approach where the interface is introduced as a singular distribution in equations written for the whole flow field. The “one-fluid” form of the equations plays a fundamental rôle for the numerical methods discussed in the rest of the book.

2.1 General principles

The derivation of the governing equations is based on three general principles: the continuum hypothesis, the hypothesis of sharp interfaces and the neglect of intermolecular forces. The assumption that fluids can be treated as a continuum is usually an excellent approximation. Real fluids are, of course, made of atoms or molecules. To understand the continuum hypothesis, consider the density or amount of mass per unit volume. If this amount were measured in a box of sufficiently small dimensions \( \ell \), it would be a wildly fluctuating quantity (see Batchelor, 1970, for a detailed discussion). However as the box side \( \ell \) increases the density becomes ever smoother, until it is well approximated by a smooth function \( \rho \). For liquids in ambient conditions this happens for \( \ell \) above a few tens of nanometers (1 nm = 10^{-9} m). In some cases, such as in dilute gases the discrete nature of matter may be felt over much larger length scales. For dilute gases the average distance between molecular collisions, or the mean free path \( \ell_{\text{mfp}} \), is the important length scale. The gas obeys the Navier-Stokes equations for
scales $\ell \gg \ell_{\text{mfp}}$. Molecular simulations, where the motion of many individual molecules is followed for sufficiently long times so that meaningful averages can be computed, show that the fluid behaves as a continuum for a surprisingly small number of molecules. Koplik et al. (1988) found, for example, that under realistic pressure and temperature a few hundred molecules in a channel resulted in a Poiseuille flow that agreed with the predictions of continuum theory.

Beyond the continuum hypothesis, for multiphase flows we shall make the assumption of sharp interfaces. Interfaces separate different fluids, such as air and water, oil and vinegar, or any other pair of immiscible fluids and different thermodynamic phases such as solid and liquid or vapor and liquid. The properties of the fluids, including their equation of state, density, viscosity and heat conductivity, generally change across the interface. The transition from one phase to another occurs on very small scales, as described above. For continuum scales we may safely assume that interfaces have vanishing thickness.

We also impose certain restrictions on the type of forces that are taken into account. Long range forces between fluid particles, such as electromagnetic forces in charged fluids, shall not be considered. Intermolecular forces, such as Van der Waals forces that play an important rôle in interface physics, are modelled by retaining their most important effect: capillarity. This effect, also called surface tension, amounts to a stress concentrated at the sharp interfaces.

The three assumptions above also reflect the fact that it would be nearly impossible, with the current state-of-the-art, to describe complex droplet and bubble interactions while keeping the microscopic physics. For instance, simulating physical phenomena from the nanometer to the centimeter scale would require $10^7$ grid points in every direction, an extravagant requirement for any type of computation, even with the use of cleverly employed adaptive mesh refinement.

Beyond the three assumptions above, we mostly deal with incompressible flows in this Book, although in the present Chapter we derive the equations initially for general flow situations.

## 2.2 Basic equations

Expressing the basic principles of conservation of mass, momentum and energy mathematically leads to the governing equations for fluid flow. In addition to the general conservation principles, we also need constitutive
assumptions about the specific nature of each fluid. Here we will work only with Newtonian fluids.

2.2.1 Mass conservation

The principle of conservation of mass states that mass cannot be created nor destroyed. Therefore, if we consider a volume $V$, fixed in space, then the mass inside this volume can only change if mass flows in or out through its boundary $S$. The flow out of $V$, through a surface element $ds$, is $\rho \mathbf{u} \cdot \mathbf{n} ds$ where $\mathbf{n}$ is the outward normal, $\rho$ is the density and $\mathbf{u}$ is the velocity. The notation is shown in Fig. 2.1. Stated in integral form, the principle of mass conservation is

$$\frac{d}{dt} \int_V \rho \, dv = - \int_S \rho \mathbf{u} \cdot \mathbf{n} ds. \tag{2.1}$$

Here, the left hand side is the rate of change of mass in the volume $V$ and the right-hand side represents the net flow through its boundary $S$. Since the volume is fixed in space we can take the derivative inside the integral and, by applying the divergence theorem to the integral of the fluxes through the boundary, we have

$$\int_V \left[ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right] dv = 0. \tag{2.2}$$

This relation must hold for any arbitrary volume, no matter how small, and that can only be true if the quantity inside the square brackets is zero. The
partial differential equation expressing conservation of mass is therefore

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0.$$  \hfill (2.3)

By using the definition of the substantial derivative

$$\frac{D()}{Dt} = \frac{\partial ()}{\partial t} + \mathbf{u} \cdot \nabla (),$$  \hfill (2.4)

and expanding the divergence, $\nabla \cdot (\rho \mathbf{u}) = \mathbf{u} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{u}$, the continuity equation can be rewritten in *convective form* as

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \mathbf{u},$$  \hfill (2.5)

emphasizing that the density of a material particle can only change if the fluid is compressed or expanded ($\nabla \cdot \mathbf{u} \neq 0$).

### 2.2.2 Momentum conservation

The equation of motion is derived by using the momentum-conservation principle, stating that the rate of change of fluid momentum in the fixed volume $V$ is the difference in momentum flux across the boundary $S$ plus the net forces acting on the volume. Therefore,

$$\frac{d}{dt} \int_V \rho \mathbf{u} dv = -\int_S \rho \mathbf{u} (\mathbf{u} \cdot \mathbf{n}) ds + \int_V \mathbf{f} dv + \int_S \mathbf{n} \cdot \mathbf{T} ds.$$  \hfill (2.6)

The first term on the right-hand side is the momentum flux through the boundary of $V$ and the next term is the total body force on $V$. Frequently, the force per unit volume $\mathbf{f}$ is only the gravitational force, $\mathbf{f} = \rho \mathbf{g}$. The last term is the total surface force. Here, the tensor $\mathbf{T}$ is a symmetric stress tensor constructed in such a way that $\mathbf{n} \cdot \mathbf{T} ds$ is the force on a surface element $ds$ with a normal $\mathbf{n}$.

By the same argument as applied to the mass conservation equation, equation (2.6) must be valid at every point in the fluid, so that

$$\frac{\partial \rho \mathbf{u}}{\partial t} = -\nabla \cdot (\rho \mathbf{u} \mathbf{u}) + \mathbf{f} + \nabla \cdot \mathbf{T}.$$  \hfill (2.7)

Here we denote with $\mathbf{ab}$, whose $ij$-th component is $a_i b_j$, the dyadic product of the two vectors $\mathbf{a}$ and $\mathbf{b}$, hence $\mathbf{uu}$ has components $u_i u_j$. The nonlinear advection term can be written as

$$\nabla \cdot (\rho \mathbf{uu}) = \rho \mathbf{u} \cdot \nabla \mathbf{u} + \mathbf{u} \nabla \cdot (\rho \mathbf{u}),$$  \hfill (2.8)
and using the definition of the substantial derivative and the continuity equation we can rewrite equation (2.7) as

$$\rho \frac{D\mathbf{u}}{Dt} = \mathbf{f} + \nabla \cdot \mathbf{T}. \quad (2.9)$$

This is Cauchy’s equation of motion and is valid for any continuous medium. For fluids like water, oil and air (as well as many others that are generally referred to as Newtonian fluids) the stress may be assumed to be a linear function of the rate of strain

$$\mathbf{T} = (-p + \lambda \nabla \cdot \mathbf{u}) \mathbf{I} + 2\mu \mathbf{S}. \quad (2.10)$$

Here, \( \mathbf{I} \) is the unit tensor, \( p \) the pressure, \( \mu \) the viscosity and \( \mathbf{S} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \) is the rate-of-strain or deformation tensor whose components are

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (2.11)$$

\( \lambda \) is the second coefficient of viscosity and if Stokes’ hypothesis is assumed to hold, then \( \lambda = -\frac{2}{3}\mu \).† Substituting the expression for the stress tensor into Cauchy’s equation of motion results in

$$\rho \frac{D\mathbf{u}}{Dt} = \mathbf{f} - \nabla p + \nabla(\lambda \nabla \cdot \mathbf{u}) + \nabla \cdot (2\mu \mathbf{S}), \quad (2.12)$$

which is the Navier-Stokes equation for fluid flow.

### 2.2.3 Energy conservation

In integral form the conservation of energy principle, as applied to a control volume \( V \) fixed in space, is:

$$\frac{d}{dt} \int_V \rho (e + \frac{1}{2} u^2) dv =$$

$$- \int_S \rho (e + \frac{1}{2} u^2) \mathbf{u} \cdot \mathbf{n} ds + \int_V \mathbf{u} \cdot \mathbf{f} dv + \int_S \mathbf{n} \cdot (\mathbf{u} \cdot \mathbf{T}) ds - \int_S \mathbf{q} ds. \quad (2.13)$$

Here, \( u^2 = \mathbf{u} \cdot \mathbf{u} \) and \( e \) is the total internal energy per unit mass. The left-hand side is the internal and kinetic energy, the first term on the right-hand side is the flow of internal and kinetic energy across the boundary, the second term represents the work done by body forces, the third term is the work done by the stresses at the boundary (pressure and viscous shear) and \( \mathbf{q} \) in the fourth term is the heat-flux vector. This equation can be simplified

† In several texts the discussion is based on the bulk viscosity \( \kappa = 2/3\mu + \lambda \). Stokes’ hypothesis is that the bulk viscosity vanishes.
by using the momentum equation. Taking the dot product of the velocity with equation (2.9) gives
\[ \rho \frac{\partial u^2}{\partial t} = -\rho u \cdot \nabla u^2 / 2 + u \cdot f + u \cdot (\nabla \cdot T) \] (2.14)
for the mechanical energy. After using this equation to cancel terms in (2.13) and applying the same arguments as before, we obtain the convective form of the energy equation:
\[ \rho \frac{De}{Dt} - T : \nabla u + \nabla \cdot q = 0. \] (2.15)
Here we denote with \( A : B = \sum_i \sum_j A_{ij} B_{ji} \) the scalar product of the two tensors \( A \) and \( B \). This equation needs to be supplemented by constitutive equations for the specific fluids we are considering.

We will assume that the flux of heat is proportional to the gradient of the temperature \( T \), so that
\[ q = -k \nabla T. \] (2.16)
This is called Fourier’s law and \( k \) is the thermal conductivity. Using Fourier’s law, assuming a Newtonian fluid, so that the stress tensor is given by equation (2.10), and that radiative heat transfer is negligible, then the energy equation takes the form:
\[ \rho \frac{De}{Dt} + p \nabla \cdot u = \Phi + \nabla \cdot k \nabla T, \] (2.17)
where \( \Phi = \lambda (\nabla \cdot u)^2 + 2\mu S : S \) is called the dissipation function. It can be shown that \( \Phi \), which represents the rate at which work is converted into heat, is always greater or equal to zero.

In general we also need an equation of state giving, say, pressure as a function of density and internal energy
\[ p = p(\varepsilon, \rho), \] (2.18)
as well as equations for the transport coefficients \( \mu, \lambda \) and \( k \) as functions of the state of the fluid.

The governing equations are summarized in Figs. 2.2 to 2.4. The integral form obtained by applying the conservation principles directly to a small control volume is shown in Fig. 2.2. While usually not very convenient for analytical work, the integral form is the starting point for Finite-Volume numerical methods. In Fig. 2.3 we show the differential form obtained directly by assuming that the integral laws hold at a point. This form can be used for finite difference numerical methods and usually leads to
\[
\frac{d}{dt} \int_V \rho dv + \oint_S \rho \mathbf{u} \cdot \mathbf{n} ds = 0
\]
\[
\frac{d}{dt} \int_V \rho \mathbf{u} dv = \int_V \mathbf{f} dv + \oint_S \left( \mathbf{n} \cdot \mathbf{T} - \rho \mathbf{u} (\mathbf{u} \cdot \mathbf{n}) \right) ds
\]
\[
\frac{d}{dt} \int_V \rho (e + \frac{1}{2} u^2) dv = \int_V \mathbf{u} \cdot \mathbf{f} dv + \oint_S \mathbf{n} \cdot \left( \mathbf{u} \cdot \mathbf{T} - \rho (e + \frac{1}{2} u^2) \mathbf{u} - \mathbf{q} \right) ds
\]

Fig. 2.2. The equations of fluid motion in integral form. The flux terms have been moved under the same surface integral as the stress terms.

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0
\]
\[
\frac{\partial \rho \mathbf{u}}{\partial t} = \mathbf{f} + \nabla \cdot (\mathbf{T} - \rho \mathbf{u} \mathbf{u})
\]
\[
\frac{\partial}{\partial t} \rho (e + \frac{1}{2} u^2) = -\nabla \cdot \left( \rho (e + \frac{1}{2} u^2) \mathbf{u} - \mathbf{T} \cdot \mathbf{u} + \mathbf{q} \right) + \mathbf{u} \cdot \mathbf{f}
\]

Fig. 2.3. The equations of fluid motion in conservative form.

\[
\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{u} = 0
\]
\[
\rho \frac{D\mathbf{u}}{Dt} = \mathbf{f} + \nabla \cdot \mathbf{T}
\]
\[
\rho \frac{De}{Dt} = \mathbf{T} : \nabla \mathbf{u} - \nabla \cdot \mathbf{q}
\]

Fig. 2.4. The equations of fluid motion in convective or non-conservative form.

discretizations that are essentially identical to those obtained by the Finite-Volume method. In Fig. 2.4 we have regrouped the terms to obtain the \textit{convective} or the \textit{non-conservative} form of the equations. This is the form usually shown in textbooks and is often used as a starting point for finite difference methods.
2.2 Basic equations

2.2.4 Incompressible Flow

For an important class of flows the density of each fluid particle does not change as they move. This is generally the case when the maximum or characteristic flow velocity $U$ is much smaller than the velocity of sound $c_s$, or equivalently when the Mach number $Ma = U/c_s$ is much smaller than unity. The equation for the evolution of the density is then

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0,$$

and the mass conservation equation becomes

$$\nabla \cdot \mathbf{u} = 0.$$  \hspace{1cm} (2.20)

Equation (2.20) states that the volume of any fluid element cannot be changed and these flows are therefore referred to as incompressible flows. If we integrate this equation over a finite volume $V$ with boundary $S$ and use the divergence theorem (or expand the divergence in (2.2) and use equation (2.3)), we find that the integral form of the mass conservation equation for incompressible flows is

$$\oint_S \mathbf{u} \cdot n ds = 0,$$

stating that inflow balances outflow.

Notice that there is no requirement that the density is the same everywhere in incompressible flows. The density of a material particle can vary from one particle to the next one, but the density of each particle must stay constant. When the density is not the same everywhere its value at any given point in space can change with time as material particles of different density are advected with the flow. In this case the density field must be updated using equation (2.19). If the density is constant everywhere this is, of course, not necessary.

The pressure plays a special rôle for incompressible flows. Instead of being a thermodynamic function of (say) density and temperature, it is determined solely by the velocity field and will take on whatever value is necessary to make the flow divergence free. It is sometimes convenient—and we will use this extensively—to think of the pressure as projecting the velocity field into the space of incompressible functions. That is, we imagine the velocity first being predicted by (2.12) without the pressure, then we find the pressure necessary to enforce incompressibility and correct the velocity field. Notice that for incompressible flow it is not necessary to solve the energy equation to find the velocity and the pressure, unless the material properties
\begin{equation}
\n\nabla \cdot \mathbf{u} = 0
\end{equation}
\begin{equation}
\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mathbf{f} + \nabla \cdot \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)
\end{equation}

Fig. 2.5. The equations of fluid motion for incompressible Newtonian flow in convective form.

are functions of the temperature. Thus, the flow field is found by solving the momentum equation, equation (2.12), along with the incompressibility condition, equation (2.20) or (2.21). The governing equations for incompressible, Newtonian flows in convective form are summarized in Fig. 2.5.

The total derivative in the momentum equation in Fig. 2.5 can be written in several different ways, all of which are equivalent analytically but that generally lead to slightly different numerical approximations. The most common ones are

\begin{equation}
\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{u}) = \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{\partial \mathbf{u}}{\partial t} + \frac{1}{2} \nabla (\mathbf{u} \cdot \mathbf{u}) - \mathbf{u} \times (\nabla \times \mathbf{u}).
\end{equation}

We will usually work with the second form, but we note that the third one is very commonly used as well. For compressible flows the fully conservative form of the momentum equations is usually used, but as discussed at the end of Chapter 3, doing so for incompressible flow with a density that changes abruptly can lead to certain numerical difficulties.

The special case of incompressible fluids with constant density and viscosity is of considerable importance. By writing the deformation tensor in component form is easily shown that \( \nabla \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) = \nabla^2 \mathbf{u} \) and the momentum equation becomes

\begin{equation}
\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \frac{\nabla p}{\rho} + \frac{\mathbf{f}}{\rho} + \nu \nabla^2 \mathbf{u},
\end{equation}

where \( \nu = \mu/\rho \) is the \textit{kinematic viscosity}.

\textbf{2.2.5 Boundary conditions}

One of the major difficulties in numerical simulations of fluid flows is the correct implementation of the boundary conditions. In principle the conditions at boundaries are well defined. For viscous, incompressible fluids we require the fluid to stick to the wall so the fluid velocity there is equal to the wall velocity

\begin{equation}
\mathbf{u} = \mathbf{U} \text{wall}.
\end{equation}
2.2 Basic equations

Fig. 2.6. In some models the fluid is allowed to slip on the solid surface. The velocity vanishes at some distance $\beta$ inside the solid, called the slip length. This model is useful when dealing with triple lines (contact lines) involving two fluids and a solid.

This equation includes both the normal and the tangential components of the velocity. For inviscid flows, where viscous stresses are absent and the fluid can slip freely at the wall, only the normal velocity is equal to that of the wall. In some cases, for instance for flow in very small channels, it is useful to introduce a slip boundary condition for the tangential velocity component $u_t$

$$u_t - U_{wall} = \beta \frac{\partial u_t}{\partial n}.$$  \hspace{1cm} (2.25)

Here $\beta$ is a slip coefficient, with the dimensions of a (small) length and $\partial / \partial n$ is the derivative in the direction normal to the wall. The slip length is the distance at which the velocity would vanish if extrapolated inside the wall (Fig. 2.6).

Frequently we are interested in simulating a fluid domain with in- and outflow boundaries, or we are interested in simulating only a part of a larger fluid-filled domain. In those cases it is necessary to specify in- and outflow boundary conditions. For inflow boundaries the velocity is usually given. Realistic outflow boundaries, on the other hand, pose a challenge that unfortunately does not have a simple solution. Similar difficulties are faced by the experimentalist who must carefully design his or her wind tunnel to provide uniform inlet velocity and outlet conditions that have minimal influence on the upstream flow. While it is probably easier to implement uniform inlet conditions computationally than in experiments, the outlet flow is more problematic since the solution often available to the experimentalist, simply by making the wind tunnel long enough, usually requires extensive number of grid points. The goal is generally to truncate the domain using
A simple way to parameterize an interface is through the equation $y = h(x)$. However, overhangs in the curve may make the height function $h$ multivalued. Other definitions then become necessary.

boundary conditions that make it “appear” longer. Conditions on the rate of change of the velocity, such as imposing a zero gradient on the velocity field, usually can be used if the outflow boundary is sufficiently far from the region of interest. Once the boundary conditions for the velocity have been specified, conditions for the pressure can be derived at both in and outflow boundaries.

In many cases computations are done in domains that are periodic in one or more coordinate direction. If we assume that the $x$ direction is periodic, then the boundary conditions for the velocity are found from $u(x, y, z) = u(x + L, y, z)$, where $L$ is the length of the period. Boundary conditions for pressure and other variables are found in the same way and similar formulae apply when the domain is periodic in $y$ and/or $z$. For theoretical work, periodic boundaries are often very attractive, since it is not necessary to deal with the often difficult task of specifying appropriate in- and outflow boundary conditions. The only new consideration is that if the domain is periodic in the direction of gravity, then it is necessary to add a body force, equal to $\rho_{av}g$, where $\rho_{av}$ is the average density of the whole domain, to prevent a uniform acceleration of the fluid. This force gives rise to an hydrostatic pressure gradient, but since it results in the conservation of momentum in the domain—not the average velocity—it does not correspond exactly to a container with a rigid bottom.
Following the motion of a deformable interface separating different fluids or phases is at the center of accurate predictions of multiphase flows. Describing the interface location and how it moves can be accomplished in several ways. The simplest case is when the location of the interface can be described by a single-valued function of one (in two dimensions) or two (in three dimensions) coordinates. For a thin liquid layer, we can, for example, write

\[ y = h(x) \quad \text{(two dimensions);} \quad z = h(x, y) \quad \text{(three dimensions).} \quad (2.26) \]

This situation is sketched in Fig. 2.7, which also shows the limitation of this approach: if part of the interface “overhangs,” \( h \) becomes a multi-valued function. While there are many situations, particularly for thin films, where equation (2.26) is the most convenient description, generally we need a more sophisticated way to describe the interface.

To handle interfaces of arbitrary shape we can parameterize the interface by introducing a coordinate \( u \) in two dimensions, such that the location of the interface is given by

\[ \mathbf{x}(u) = (x(u), y(u)). \quad (2.27) \]

Fig. 2.8 shows a closed contour separating fluid “1” from fluid “2”, described in this way. For a curve in two dimensions, described by equation (2.27), \( d\mathbf{x}/du \) yields a vector tangent to the curve. Although \( u \) can be any
parameterization of the curve, the simplest case is when \( u \) is taken to be the arc length \( s \), where \( ds^2 = dx^2 + dy^2 \). The tangent vector \( t = \frac{dx}{ds} \) is then the unit tangent and \( |dx/ds| = 1 \). The unit normal \( n \) is perpendicular to the curve, such that \( t \cdot n = 0 \). The orientation of the normal is arbitrary, and one should take the most convenient one depending on the problem. For a closed curve it is customary to let the normal point outwards. However later on we will use the convention that the normal points from one of the fluids, to the other. For instance for an air-water mixture, water may be taken as the reference fluid, air being the other fluid. The normal then points from water to air. We usually label the reference fluid as fluid 1 and the other as fluid 2 and the normal points from 1 to 2.

Consider the case when the normal points outwards and the parameterization follows the curve in the trigonometric (anti-clockwise) direction as in Fig. 2.8. Then if the tangent vector is given by \( t = (t_1, t_2) \) the normal is given by \( n = (t_2, -t_1) \). The length of the unit tangent is by definition constant so as we move along the curve, the change of the unit tangent vector must be in the normal direction. The magnitude of the change of the tangent vector, with respect to the arc length \( s \), is the curvature of the interface. We therefore have

\[
\frac{dt}{ds} = \kappa n. \tag{2.28}
\]

Similarly, it can be shown that

\[
\frac{dn}{ds} = -\kappa t. \tag{2.29}
\]

The two previous equations are known as an instance of the Frenet-Serret formulae. The sign of the curvature may be a vexing question. It depends on one’s choice of normal orientation as shown on Fig. 2.9. With our previous conventions, for a closed circle of radius \( R \), the curvature is negative and equal to \(-1/R\).

By taking the dot product of these equations with the normal and the tangent, respectively, the curvature is found to be

\[
\kappa = n \cdot \frac{dt}{ds} = -t \cdot \frac{dn}{ds}. \tag{2.30}
\]

In Appendix 1 we examine the properties of a two-dimensional curve in more detail and work out the above expressions in component form.

For an interface in three-dimensional space we need two independent coordinates, \( u \) and \( v \) and the interface is described by

\[
x(u, v) = \left( x(u, v), y(u, v), z(u, v) \right), \tag{2.31}
\]
The sign of the curvature depends on one’s choice for the orientation of the normal. If the curve folds towards the normal the curvature is positive.

A small surface element. The surface is parameterized by $u$ and $v$. $\mathbf{p}$ is a vector perpendicular to the edge of the element but tangent to the surface.

as shown in Fig. 2.10. The geometry of three-dimensional surfaces is an elaborate subject and we will only need a few specific results. To avoid cluttering the discussion unnecessarily we have gathered some of the more pertinent elements of the theory in Appendix 1 and here we simply quote the results that we need.

In addition to knowing the location of the interface, we frequently need to compute the normal to the interface, tangent vectors and the mean curvature. Differentiating $\mathbf{x}(u,v)$ with respect to the surface coordinates yields
two vectors tangent to the surface. We will use the shorthand
\[ \mathbf{x}_u = \frac{\partial \mathbf{x}}{\partial u}, \quad \text{and} \quad \mathbf{x}_v = \frac{\partial \mathbf{x}}{\partial v}. \] (2.32)

With \( \mathbf{x}_u \) and \( \mathbf{x}_v \) both tangent to the interface, we can find the unit normal by taking the cross product of the tangent vectors and dividing by the length of the product
\[ \mathbf{n} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{|\mathbf{x}_u \times \mathbf{x}_v|}. \] (2.33)

In Appendix 1 it is shown that the mean curvature can be found by taking the surface divergence of the normal vector
\[ \kappa = -\nabla_S \cdot \mathbf{n}. \] (2.34)

Here \( \nabla_S \) denotes the surface divergence, defined in Appendix 1. However, as also shown in the Appendix, if we define a normal field that extends off the interface, then the curvature can be found by taking the usual divergence of this extended normal field
\[ \kappa = -\nabla \cdot \mathbf{n}. \] (2.35)

We will also use the fact that the curvature can be found as the integral, along the edges of an infinitesimal surface element of area \( \delta A \), of the interface tangent vector \( \mathbf{p} \) perpendicular to the edge of the surface element (see Fig. 2.10 and a derivation in Appendix 1)
\[ \kappa \mathbf{n} = \lim_{\delta A \to 0} \frac{1}{\delta A} \oint \mathbf{p} dl. \] (2.36)

Here, the vector \( \mathbf{p} = \mathbf{t} \times \mathbf{n} \) is tangent to the surface element and also perpendicular to both the normal vector \( \mathbf{n} \) and the vector \( \mathbf{t} \) tangent to the edge of the surface element. Thus, \( \mathbf{p} \) “pulls” on the boundary of the surface element.

Instead of identifying the interface by explicitly specifying the location of every point on the interface, the interface can also be given by a marker function defined in the whole domain. Such marker functions can take many forms. We can, for example, use the characteristic function, defined as a discontinuous function by
\[ H(\mathbf{x}) = \begin{cases} 
1 & \text{if inside a closed interface;} \\
0 & \text{if outside a closed interface.} 
\end{cases} \] (2.37)

Alternatively, we may decide that \( H = 1 \) for fluid “1” and \( H = 0 \) for fluid “2”. With \( H \) given, the interface is identified with the sharp change from one value to the other (Fig. 2.11). \( H \) can be constructed in different ways,
but for our purpose it is convenient to express it in terms of an integral over the product of one-dimensional $\delta$-functions. For a two-dimensional domain:

$$H(x, y) = \int_V \delta(x - x')\delta(y - y')\,dv'. \quad (2.38)$$

Here, the integration is over the region bounded by the contour $S$ and $dv' = dx'dy'$ (Fig. 2.11). Obviously, $H = 1$ if the point $(x, y)$ is inside the contour, where it will coincide with $(x', y')$ during the integration, and zero if it is outside and will never be equal to $(x', y')$.

The properties of the step function are discussed in more detail in Appendix 1 where we show that the gradient of $H$ is given by

$$\nabla H(x, y) = -\int_S \delta(x - x')\delta(y - y')n'ds' = -\delta(n)n, \quad (2.39)$$

where $n$ is the coordinate normal to the interface in a local coordinate system aligned with the interface. We can also define a surface distribution $\delta_S(x) = \delta(n)$, and write

$$\nabla H = -\delta_S n. \quad (2.40)$$

In addition, the interface can be described by a smooth function $F$, if the interface is identified with a particular value of the function, say $F = 0$ (Fig. 2.12). Obviously, $F < 0$ on one side of the interface and $F > 0$ on the other. One usually takes the reference fluid (fluid “1”) to be in the $F > 0$ region. Taking $F > 0$ inside the closed surface, the normal points outwards and is
found by

\[ \mathbf{n} = -\frac{\nabla F}{|\nabla F|} \]  

(2.41)

and the curvature is given by

\[ \kappa = -\nabla \cdot \mathbf{n} = \nabla \cdot \left( \frac{\nabla F}{|\nabla F|} \right). \]  

(2.42)

The representation of the interface as a contour with a specific value is used in Level-Set methods.

Fig. 2.12. The identification of an interface by a level-set function.

The motion of the interface \( S \) is determined by the normal velocity \( V(u, v, t) \), for each point on \( S \). This velocity may be that of the fluid itself, or may be different due to, for example, evaporation or condensation. In parametric form

\[ V(u, v, t) = \mathbf{n} \cdot \frac{\partial}{\partial t} \mathbf{x}(u, v, t). \]  

(2.43)

To update the location of \( S \), we note that there is no need to specify a tangential velocity, although from a physical point of view the interface is
made of fluid particles that travel at some tangential as well as normal velocity. The fact that the motion of the interface is determined only by the normal velocity at the interface can be seen by considering the interface as a level-set function $F$. Since the function moves with the fluid velocity, its motion is described by

$$\frac{D F}{D t} = \frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0. \quad (2.44)$$

The normal to the interface is given by (2.41) so we can rewrite the above equation as

$$\frac{\partial F}{\partial t} - \mathbf{u} \cdot \mathbf{n} |\nabla F| = \frac{\partial F}{\partial t} - V |\nabla F| = 0. \quad (2.45)$$

which involves only the normal velocity $V = \mathbf{u} \cdot \mathbf{n}$.

### 2.4 Fluid mechanics with interfaces

Consider the incompressible flow of two immiscible fluids filling a given domain. The domain may be decomposed into any number of subdomains filled with the individual phases. In any subdomain the usual Navier-Stokes equations hold and here we derive the interface conditions that allow us to couple the fluid motion in the different fluids, across the interface. These conditions are, for the most part, derived using mass and momentum conservation. In one case (for surface tension) it is necessary to introduce additional physics.
2.4.1 Mass conservation and velocity conditions

To derive the jump conditions for the normal velocity, we apply the conservation principle used in Section 2.2.1 to the control volume shown in Fig. 2.13. Since its thickness is taken to be zero, there can be no accumulation of mass inside it, so the mass flux into the control volume must be equal to the flow out. If the velocity on one side of the interface is \( u_1 \) and on the other it is \( u_2 \) and the normal velocity of the interface is \( V \), then the inflow is \( u_1 \cdot n - V \) and the outflow is \( u_2 \cdot n - V \). Denoting the mass flow across the interface by \( \dot{m} \), we have

\[
\rho_1 (u_1 \cdot n - V) = \rho_2 (u_2 \cdot n - V) = \dot{m}.
\]

This is simply the Rankine-Hugoniot condition. If there is no change of phase, \( \dot{m} = 0 \). For incompressible flows this condition must hold for arbitrary density ratios and we therefore have

\[
V = u_1 \cdot n = u_2 \cdot n.
\]

Mass conservation places no restriction on the tangential velocity components. Indeed, inviscid fluids are usually assumed to slip at the interface. For very low viscosity and rarefied gases it is possible that finite slip is the appropriate interface condition, but for viscous fluid under normal operating conditions it is an experimentally observed fact—like the no-slip boundary conditions at solid walls—that no slip takes place. It is also possible to argue that if the sharp interface is assumed to come about by taking the limit of a finite interface zone as its thickness goes to zero, then anything but no slip would result in infinitely high stresses if the interface zone consists of a fluid with a finite viscosity. Thus, if there is no phase change and the fluids are incompressible, the interfacial condition for viscous fluid is simply \( u_1 = u_2 \), or

\[
[u]_S = 0,
\]

where we have used the jump notation, i.e. the notation \( [x]_S = x_2 - x_1 \).

2.4.2 Surface tension

From a molecular point of view, surface tension arises because the interface is not an optimal region thermodynamically: the molecules “prefer” to be at the gas or the liquid density, which minimizes the free energy. The non-optimal conditions near the interface result in an excess energy

\[
dE^\sigma = \sigma ds,
\]
where $ds$ is an infinitesimal interface or surface area and $\sigma$ is a material property usually referred to as the surface tension coefficient or simply surface tension. From a mechanical point of view, however, surface tension is simply a force per unit length acting perpendicularly on any line segment in the surface. If $p$ is a vector perpendicular to the line segment, the “pull” is simply $\sigma p$ per unit length. These points of view are equivalent: to stretch the surface one has to pull on it. Suppose that we pull in the $x$ direction and that the surface has extension $L$ in the other direction. Stretching it increases the area by $ds = Ldx$, at the expense of a work $\sigma Ldx = \sigma ds$ increasing as much the interfacial free energy.

By arguments that are exactly analogous to the introduction of the stress tensor for a three-dimensional continuum (see Aris, 1962, for example), it can be shown that the force on the edge of a small surface element can be written as $T^\sigma_S \cdot p$, where $T^\sigma_S$ is the surface tension tensor and $p$ is a vector normal to the edge of the element, in the tangent plane to the surface. Since

$$T^\sigma_S \cdot p = \sigma p = \sigma I_S \cdot p$$

we find that $T^\sigma_S = \sigma I_S$, where $I_S$ is the surface identity tensor.

The surface tension tensor was defined above in terms of the surface identity tensor. Frequently it is more convenient to work with the full three-dimensional operator and use the fact that $I_S$ is the tangential projection of the three-dimensional identity tensor $I$. Thus $I_S = (I - nn)$ and

$$T^\sigma_S = \sigma(I - nn).$$

Using that $n = t_1 \times t_2$ where $t_1$ and $t_2$ are orthonormal tangent vectors to the surface, the surface tension tensor can also be written as

$$T^\sigma_S = \sigma(t_1 t_1 + t_2 t_2).$$

The force on a surface element of area $S$, bounded by a contour $C$, is the integral of the “pull” on its edges

$$\delta F_\sigma = \int_C T^\sigma_S \cdot p dl = \int_S \nabla_S \cdot T^\sigma_S ds.$$  

Taking the limit of equation (2.53) as the surface area shrinks to a point, we define the surface force per unit area as

$$f_\sigma = \nabla \cdot T^\sigma_S = \nabla \cdot \sigma I_S = \sigma \nabla_S \cdot I_S + I_S \cdot \nabla_S \sigma.$$  

The first term can be shown to be $\sigma \kappa n$ and the second term is simply $\nabla_S \sigma$. Thus,

$$f_\sigma = \sigma \kappa n + \nabla_S \sigma,$$
where the last term, the surface gradient of $\sigma$, is obviously zero for constant surface tension.

### 2.4.3 Momentum conservation with interfaces

Applying the conservation of momentum principle to the control volume in Fig. 2.13, moving with the interface $S$, we obtain

\[
0 = - \oint_{\delta S} \rho u(u \cdot n - V) ds + \oint_{\delta S} n \cdot T ds + \oint_{S} f_r ds, \tag{2.56}
\]

Notice that the integration is around the edges of the control volume $\delta V$ for the first two terms and along the interface for the last term. The first term on the right-hand side is zero for incompressible flows in the absence of phase change, since the control volume is moving with the fluid velocity. As the thickness of the control volume approaches zero, the boundary coincides with the interface and by integrating the stresses first on one side and then on the other, the second term yields the jump in $T$ across the interface. Since the results hold for any control volume that includes the interface, they must be valid at any interface point, resulting in

\[
- [T]_S \cdot n = \sigma \kappa n + \nabla_S \sigma. \tag{2.57}
\]

This condition may be split into a normal and tangential stress condition, using equation (2.55) for $f_r$, yielding the boxed jump conditions in Fig. 2.14.

### 2.4.4 Free-surface flow

Free-surface flow is a limiting case of flows with interfaces, in which the treatment of one of the fluids is simplified. For air-water flow, for instance, we can sometimes assume that the pressure $p_{\text{free}}$ in the air is a constant, or depends only on time, and that the viscous stresses in the air are negligible. The jump conditions then become boundary conditions for the liquid domain as shown in Fig. 2.15. In the case of constant $\sigma$, the tangential stresses vanish and

\[
t^{(k)} \cdot S \cdot n \bigg|_S = 0. \tag{2.58}
\]

This is a purely kinematic condition: it does not involve material properties at all. For this reason it has interesting consequences. The vorticity vector is defined in general as $\nabla \times u$ and its study reveals interesting aspects of the
2.4 Fluid mechanics with interfaces

\[ [\mathbf{u}]_S = 0, \]
\[ -[-p + 2\mu \mathbf{n} \cdot \mathbf{S} \cdot \mathbf{n}]_S = \sigma \kappa, \]
\[ -\left[2\mu t^{(k)} \cdot \mathbf{S} \cdot \mathbf{n}\right]_S = t^{(k)} \cdot \nabla_S \sigma, \]
\[ \nabla \cdot \mathbf{u} = 0, \]
\[ \rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mathbf{f} + \mu \nabla^2 \mathbf{u}. \]

Fig. 2.14. The jump conditions and the partial differential equations for a Newtonian, incompressible fluid, without evaporation or condensation. The last two equations apply in the bulk of each phase. The first three are jump conditions on the interface \( S \). There, the \( t^{(k)} \) are two unit tangent vectors. In the two-dimensional flow case there is a single tangent vector. Together with boundary conditions on walls or entry and exit conditions, these equations form a complete set for flows with interfaces.

\[ \left. (-p + 2\mu \mathbf{n} \cdot \mathbf{S} \cdot \mathbf{n}) \right|_S = -p_{\text{free}} + \sigma \kappa, \]
\[ 2\mu t^{(k)} \cdot \mathbf{S} \cdot \mathbf{n} \bigg|_S = t^{(k)} \cdot \nabla_S \sigma, \]
\[ \nabla \cdot \mathbf{u} = 0, \]
\[ \rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mathbf{f} + \mu \nabla^2 \mathbf{u}. \]

Fig. 2.15. The boundary conditions and the partial differential equations for a Newtonian, incompressible fluid with a free surface. In this example the pressure on the free surface is set to \( p_{\text{free}} \).

dynamics of the flow. In two dimensions, we note \( \omega = (\nabla \times \mathbf{u}) \cdot \mathbf{e}_z \) its only non-zero component. Then on the free surface

\[ \omega = 2\kappa q \]  \hspace{1cm} (2.59)

where \( q = (\mathbf{u} \cdot \mathbf{u})^{1/2} \) is the norm of the velocity as shown in Batchelor (1970). This implies that a large vorticity will be seen on the free surface in regions of high velocity and high curvature. This is where the departure from the hypothesis of irrotational flow will be the most pronounced. Moreover, the vorticity is also seen in these regions for two-phase flows with a strong density and viscosity contrast, since these flows resemble free-surface flows.
Numerical simulations show characteristic patches of vorticity in the regions of high curvature.

2.5 Fluid mechanics with interfaces: the one-fluid formulation

In contrast to the approach described in Section 2.4, where we wrote the governing equations separately for each phase and used jump conditions to couple the solutions at the fluid interface, it is possible to write one set of governing equations for the whole flow domain occupied by the various phases, without resorting to jump conditions. The various phases are treated as one fluid with variable material properties that change abruptly at the phase boundary. To account for the “extra” forces at the phase boundary it is, however, necessary to add singular terms (δ-functions) to the equations. These singular terms are the counterpart of the jump conditions of the preceding section and it can be shown that both formulations are equivalent.

This form of the equations is often referred to as the “one-fluid” approach. Since the solution can change discontinuously across the interface, we must either interpret the governing equations in a weak sense where they are satisfied only in an integral sense, or admit solutions that include generalized functions such as delta functions and step functions. The latter approach is taken here. The “one-fluid” formulation is the starting point for several numerical methods based on the use of fixed grids, as we will see in later Chapters.

The derivation of the “one-fluid” equations is exactly the same as before (Section 2.2) except that we need to add the surface tension as a body force to the momentum equation. The mass conservation equation for the total mass has no source terms. For reacting flows the conservation equation for each species may have localized source terms and for flows with phase change (Chapter 11) we shall see that a localized volume source emerges naturally at the phase boundary.

For a control volume including an interface, the surface tension force is given by \( f_s \), integrated over \( S \), the part of the surface enclosed in the control volume. Just as we use the divergence theorem to convert surface integrals to volume integrals, we use equation (A2.9) to transform the surface integral

\[
\int_S f_s ds = \int_V f_s \delta_S dv, \tag{2.60}
\]

where \( \delta_S = \delta_S(\mathbf{x} - \mathbf{x}_s) \). Adding this force to the integral form of the momentum equation, equation (2.6), and applying the same arguments as before—that the integral can only be zero for all possible control volumes if the
2.6 Nondimensional numbers

The dynamics of multifluid and multiphase flows is governed by a variety of nondimensional numbers, depending on the exact situation driving the flow. Those include the *Reynolds number* $Re$, usually defined based on the properties of one of the fluids, using an external length scale $L$ and velocity $U$

$$Re = \frac{\rho U L}{\mu} = \frac{LU}{\nu}. \quad (2.64)$$

$Re$ represents the ratio of inertial to viscous forces. If we construct a velocity scale by $\sqrt{(\Delta \rho/\rho)gL}$, then the square of the Reynolds number becomes the *Galileo number*

$$N = \frac{g \rho L}{\mu^2}. \quad (2.65)$$
The ratio of inertia to surface tension gives the *Weber number* defined by

\[
We = \frac{\rho U^2}{\sigma},
\]

and the ratio of viscous to capillary stresses yields the *capillary number* given by

\[
Ca = \frac{\mu U}{\sigma}.
\]

Eliminating the velocity \( U \) between the Reynolds and Weber numbers gives the *Ohnesorge number*

\[
Oh = \frac{\mu}{\sqrt{\rho \sigma L}}.
\]

We sometimes also work with the *Laplace number*, defined by \( La = 1/Oh^2 \). The combination

\[
Eo = \frac{\Delta \rho g L^2}{\sigma},
\]

defines the *Eötvös number*, sometimes also called the *Bond number*. The *Morton number* is given by

\[
Mo = \frac{\Delta \rho g \mu^4}{\rho^2 \sigma^3}.
\]

The Morton number has the attractive property that for a given \( g \), it depends only on the fluid properties. When inertia and gravity are the important forces, such as for surface waves, we often work with the *Froude number*

\[
Fr = \frac{U^2}{gL}.
\]

The numbers listed here are the most common ones for two-fluid flows. For systems with more complex physics, such as heat or mass transfer, phase change and chemical reactions, additional numbers must be defined.

### 2.7 Thin films, intermolecular forces and contact lines

Topology changes in multiphase flows, such as when two droplets coalesce into one or one droplet breaks into two, take place when thin films rupture and thin threads snap. Thin threads appear to be—by far—the easiest to deal with. There are good reasons to believe that the Navier-Stokes equations describe how their diameter becomes zero in a finite time and simulations suggest that the overall results are relatively insensitive to how they are treated. For thin films, on the other hand, it is necessary to include
additional physics. While including molecular effects cannot, in principle, be done in the framework of continuum mechanics and sharp interfaces, it is arguable that mesoscopic scales, only moderately larger than the microscopic scale, may still be described by continuum mechanics. We give a very elementary introduction to this kind of modeling in this Section. More developments on breakup are given in Chapter 9.

2.7.1 Disjoining pressure and forces between interfaces

When two interfaces are less than a few hundred nanometers apart, intermolecular forces are significant. Those may be modeled as an additional singular force on the interface. The force $f_I$ per unit area of the interface, directed away from the other interface is often taken to be given by

$$f_I = -Ah^{-3},$$

where $A$ is the Hamaker’s constant and $h$ is the distance between the two interfaces, see Fig. 2.16. For two parallel interfaces the surface force in equation (2.60) then takes the form

$$f_S = -Ah^{-3}n_S + \sigma \kappa n_S + \nabla S \sigma n_S,$$

where $n$ is the normal oriented away from the reference phase. Such forces are expected both for a free film (Fig. 2.16) and for a solid-bounded film (Fig. 2.17).

The equivalent of the normal stress condition on a free surface (Laplace’s law in the static case) is now

$$(-p + 2\mu n \cdot S \cdot n)|_S = -p_{\text{free}} + \sigma \kappa - Ah^{-3}. $$

In the static case the viscous terms may be neglected and

$$p = p_{\text{free}} - \sigma \kappa + Ah^{-3},$$

Fig. 2.16. A thin film of thickness $h$ may be subject to intermolecular forces, creating an effective attraction or repulsion between the two interfaces.
so the pressure is increased or decreased depending on the sign of $A$. Thus $\Pi(h) = -Ah^{-3}$ is called the disjoining pressure. When $A > 0$ the disjoining pressure is negative and the interfaces attract each other. On the other hand when $A < 0$ the disjoining pressure is positive and the interfaces repel each other. The value of $A$ depends on the fluid but values in the range of $10^{-20}$ Joules are common. It is well known that when $A > 0$ the film breaks, as for pure water, while when $A < 0$ the film is stable, as for water contaminated with surfactants. Indeed positive $A$ tends to increase the pressure in narrow regions and forces the fluid to flow out of them. Thus a catastrophic instability occurs and the film ruptures.

In principle, equation (2.74) can be used when solving the Navier-Stokes equation for the macroscopic flow evolution. However, the range of scales that must be resolved is enormous and this has seldom been achieved in the literature.

### 2.7.2 Contact line statics and dynamics

The interface separating two fluids may meet a solid boundary at a contact line (Fig. 2.18). In the absence of motion, the contact line angle $\theta$ is fixed by the capillary properties of the substrate and the two fluids. Consider a liquid and a gas on a solid surface. The liquid-gas surface energy per unit area is the capillary tension $\sigma$. For the liquid-solid and the gas-solid interfaces we have respectively energies per unit area $\sigma_{ls}$ and $\sigma_{gs}$. Moving the contact point by a small distance $\delta x$ along the solid surface results in a gain of energy $\sigma_{gs}\delta x$ on one side and a loss $(\sigma_{ls} + \sigma \cos \theta_{eq})\delta x$ on the other side. Equating these energies yields the famous Young-Laplace relation

$$\sigma_{gs} = \sigma_{ls} + \sigma \cos \theta_{eq},$$  \hspace{1cm} (2.76)
where $\theta_{eq}$ is the equilibrium value of the contact angle. When $\theta > \theta_{eq}$, it is energetically favorable to displace the contact line to the right, and the reverse for $\theta < \theta_{eq}$.

However, in experiments it is found that the situation is more complicated: the contact angle exhibits a hysteresis. The interface does not move in some range of values $\theta_a < \theta < \theta_r$ where $\theta_a$ is the advancing contact angle and $\theta_r$ is the receding contact angle. For instance, a droplet skidding on a window may at times stop: the contact line is then pinned by surface heterogeneity in an equilibrium position between the advancing and receding angles.

When the contact line moves, the energy gained is both surface energy and any potential energy due to the external forces moving the fluid: gravity or inertia. The energy lost is dissipated by viscosity. The balance of dissipation and energy gained does not lead simply to a law for the contact line motion. The difficulty comes from a famous property of the viscous dissipation: the dissipation per unit volume is singular near the contact line. Consider a perfect wedge of fluid near the contact line, so that the height function is

$$h(x, t) = \theta(x_c - x)$$

for small $\theta$, where $x_c$ is the contact line position. Consider a steadily advancing contact line at velocity $U$. The velocity on the wall should be $u = 0$, so the horizontal component should be $u = 0$. On the other hand near the interface the velocity should be of order $U$ to advance the interface so that velocity gradients are of order

$$\frac{\partial u}{\partial y} \sim \frac{U}{h}.$$  

Moreover, the energy dissipated up to a distance $a$ from the contact line is

$$\epsilon = \int_{x_c}^{x_c+a} \int_0^{h(x,t)} \mu \left( \frac{\partial u}{\partial y} \right)^2 \, dy \, dx.$$
Using (2.77) and the estimate (2.78) for the velocity gradient this becomes

$$
\epsilon = \int_0^a \mu \frac{U^2}{\partial x} dx
$$

(2.80)

which diverges logarithmically near \( x = 0 \). This divergence leads to a paradox: in the framework of continuum mechanics any motion of the contact line would dissipate infinite energy and thus the contact line cannot move. One way out of the paradox is to use partial-slip boundary conditions such as equation (2.25). This cuts off the divergence and the dissipation becomes

$$
\epsilon = \frac{\mu U^2}{\theta} \ln \left( \frac{a}{\beta} \right).
$$

(2.81)

2.8 Notes

2.8.1 Fluid and interface mechanics

A derivation or even a textbook presentation of the equations governing fluid interfaces is not the primary purpose of this monograph. The interested reader may consult basic fluid mechanics treatises such as Batchelor (1970) who devotes ample space to issues involving interfaces and for a more systematic description of interface geometry the reader is referred to Weatherburn (1927) as well as other references, such as Aris (1962).

While the constitutive assumption given by equation (2.49) allows for variable surface tension, due to contaminants or temperature variations, for example, we have excluded any stresses due to surface shear or dilation. More complex constitutive equations can be introduced to account for such effects, see Scriven (1960), Aris (1962) and Edwards et al. (1961) for discussions.

2.8.2 Thin films and contact lines

Intermolecular forces, our equation (2.72), can be modeled in a variety of ways, depending on the polar nature of the interface, on electric charges and the presence of large molecules on the interface. Other forms of the disjoining pressure may be found in Oron et al. (1997), which also provides a general introduction to the dynamics of thin films. The thin film equations are in general of the form

$$
\frac{\partial h}{\partial t} = \frac{\partial}{\partial x} \left[ f(h) \frac{\partial h}{\partial x} + g(h) \frac{\partial^3 h}{\partial x^3} + k(h) \right],
$$

(2.82)

where \( f(h) \), \( g(h) \) and \( k(h) \) are some functions dependent on the exact problem being solved. For Hamaker’s law and surface tension on a solid substrate
(equation 2.74) the film breaks forming a self-similar solution for the thickness $h$ that goes to zero as $t^{1/5}$ (Lister and Zhang, 1999).

The slip length condition (2.25) first appeared in Navier (1823) in the very paper that gave the viscous flow equations. It acquired renewed interest as Huh and Scriven (1971) proposed it to remove the contact line dissipation paradox. The contact line paradoxes are discussed in detail in Dussan (1979) and issues related to static and dynamic wetting in general are discussed in de Gennes (1985), de Gennes et al. (2003), Oron et al. (1997), Pomeau (2002) and Bonn et al. (2009).

Experimental observations yield mobility relation of the form

$$\theta(a) = f(Ca, a/\beta, ...),$$

relating the velocity of the contact line to the apparent contact angle $\theta(a)$ at a distance $a$ from the contact line. The length $\beta$ is some microscopic scale, for instance the slip length in equation (2.25). The theoretical form of the function $f$ is still debated. For a perfectly wetting fluid ($\theta_{eq} = 0$) and small angles several approximations and theories (Bonn et al., 2009) yield

$$\theta(a) \simeq [9Ca \ln(a/\beta)]^{1/3},$$

The logarithmic dependence may be seen as arising from the logarithmic singularity in equation (2.81). This equation was justified by the Navier slip condition (2.25) but other microscopic theories have been proposed. A generalized Navier slip model has been proposed by Shikmurzaev (1997), who also noticed that the Navier slip, while removing the dissipation singularity did not remove a singularity in the pressure. Intermolecular forces have also been invoked (Hervet and de Gennes, 1984; de Gennes, 1985) as well as Phase-Field (or Cahn-Hilliard, or second gradient, or diffuse interface) models (Pomeau, 2002) or interface relaxation theories (Blake and Shikmurzaev, 2002). For a recent discussion see Bonn et al. (2009).

In computational studies it is natural to fix the contact angle in the first cell near the wall. The grid size (assuming a square grid $\Delta x = \Delta y$) is the minimum distance from the tip that is “seen” by the numerical method. Thus an expression useful as a sort of “effective boundary condition” for numerics should be of the form

$$\theta_{num} = f(Ca, \Delta x/\beta, ...),$$

as for example discussed in Afkhami et al. (2009). However this approach has not often been used and many authors have instead fixed the dynamic contact angle in the numerical method without dependence on the grid size,
or have tried to use a grid scale smaller than the slip length to avoid the grid dependence.
The one-field formulation of the Navier-Stokes equations described in the last Chapter, where a single set of equations is used to describe the motion of all the fluids present, allows us to use numerical methods developed for single phase flows. There are, however, two complications: the material properties (usually density and viscosity) generally vary from one fluid to the other and to set these properties we must construct an indicator function that identifies each fluid. We must usually also find the surface tension at the interface. The advection of the indicator function is the topic of Chapters 4 to 6 and finding the surface tension will be dealt with in Chapter 7. In the present Chapter we discuss numerical methods to solve the Navier-Stokes equations, allowing for variable density and viscosity. We will use the Finite-Volume method and limit the presentation to regular Cartesian grids. Since the multiphase flows considered in this book all involve relatively low velocities, we will assume incompressible flows.

For any numerical solution of the time-dependent Navier-Stokes equations it is necessary to decide:

(i) how the grid points, where the various discrete approximations are stored, are arranged,
(ii) how the velocity field is integrated in time,
(iii) how the advection and the viscous terms are discretized,
(iv) how the pressure equation, resulting from the incompressibility condition, is solved,
(v) how boundary conditions are implemented.

These tasks can be accomplished in a variety of ways, but the approach outlined here has been widely used for multiphase flow simulations and results in a reasonably accurate and robust numerical method.
The governing equations are the Navier-Stokes equations for incompressible, Newtonian, flow (Fig. 2.5) including the incompressibility condition, (equation 2.20 or 2.21), derived in the last Chapter. As discussed there, the equations can be written in a number of equivalent forms and, to keep the discussion as general as possible, we will write the momentum equation symbolically as

\[
\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{A} = -\nabla p + \mathbf{D} + \mathbf{f},
\]

where \( \mathbf{A} = \nabla \cdot (\mathbf{uu}) \) is the advection term, \( \mathbf{D} = \nabla \cdot \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \) the diffusion term, and \( \mathbf{f} \) denotes all other forces, such as gravity and surface tension. Although we have selected a specific form for the advection term here, obviously any of the forms shown in equation (2.22) could be used instead.

### 3.1 Time integration

For incompressible flows, where the pressure must be adjusted to give a divergence-free velocity field at the end of each time-step, the governing equations are usually solved by the so-called projection method. In this method, a temporary velocity field is first found by ignoring the pressure gradient. This velocity field is generally not divergence free and in the second step the velocity is corrected by adding the appropriate pressure gradient. Formally, the second step is a projection onto a space of divergence-free velocity fields. Hence the name. Although it is likely that the early MAC method for incompressible flows was programmed in exactly this way, the concept was introduced by Chorin (1968) and Yanenko (1971).

The integration in time is usually done using a second (or higher) order method, but for simplicity we start by describing a first-order method. A few different ways of achieving second-order accuracy are then outlined. Using a first-order, explicit, forward in time discretization of the time derivative, the momentum equation (3.1) becomes

\[
\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{A}_h^n = \frac{1}{\rho^n} \left( -\nabla_h p + \mathbf{D}_h^n + \mathbf{f}^n \right).
\]

Here, \( \Delta t \) is the size of the time step. The superscript \( n \) denotes a quantity evaluated at the beginning of the step and \( n + 1 \) identifies the end of the step. We have introduced the notation \( \mathbf{A}_h \) and \( \mathbf{D}_h \) for numerical approximations of the advection and diffusion terms, respectively and \( \nabla_h \) stands for a numerical approximation of the gradient or the divergence operators. The velocity at
the end of the time step must be divergence free:

$$\nabla_h \cdot \mathbf{u}^{n+1} = 0. \quad (3.3)$$

Since the momentum equation includes a time derivative for the velocity field it can be used to find the velocity at the next time step, once the pressure is known. The continuity equation, however, is a condition that must be satisfied by the velocity at the end of the step, and although the pressure must be adjusted in such a way that it is satisfied, there is no explicit equation for the pressure in the formulation described by equations (3.2) and (3.3). It is therefore necessary to devise a strategy to solve the equations in the right order, where all the pieces of information are available when they are needed. This is where the projection method comes in. The momentum equation, (3.2), is split into two parts: the first is a predictor step where a temporary velocity field ($\mathbf{u}^*$) is found by ignoring the effect of the pressure:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\mathbf{A}_h^n + \frac{1}{\rho^n} \left( \mathbf{D}_h^n + \mathbf{f}^n \right). \quad (3.4)$$

In the second step, the projection step, the pressure gradient is added to yield the final velocity at the new time step:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho^n} \nabla_h p. \quad (3.5)$$

Adding those two equations yields exactly equation (3.2).

To find the pressure, such that the velocity at the new time step is divergence free, we take the divergence of (3.5) and use (3.3) to eliminate $\mathbf{u}^{n+1}$, resulting in a Poisson equation for the pressure:

$$\nabla_h \cdot \left( \frac{1}{\rho^n} \nabla_h p \right) = \frac{1}{\Delta t} \nabla_h \cdot \mathbf{u}^*. \quad (3.6)$$

Once the pressure has been found, equation (3.5) can be used to find the projected velocity at time $n + 1$.

The scheme described above is completely explicit and the time step must therefore be small enough to ensure stability. The advection term $\mathbf{A}$ and the diffusion term $\mathbf{D}$ both impose restrictions on $\Delta t$. The diffusion term is almost always discretized using second-order, centered, finite-difference approximations, which require

$$\frac{\mu \Delta t}{\rho h^2} \leq \frac{1}{4} \quad (3.7)$$

for two-dimensional flow. In three dimensions, the right-hand side is replaced
Numerical solutions of the Navier-Stokes equations

by $1/6$. Here, $h$ is the smallest grid spacing ($\Delta x$, $\Delta y$, or $\Delta z$). If we use a second-order centered approximation for the advection terms, the explicit approximation in equation (3.4) results in a scheme that is unstable in the absence of diffusion. The scheme is stabilized by the viscous terms, if $\Delta t$ is limited by

$$\frac{(u \cdot u) \rho \Delta t}{\mu} \leq 2. \quad (3.8)$$

Notice that this condition does not involve the grid spacing $h$.

Temporal and spatial discretizations that result in explicit but stable schemes for the advection terms (see below) are usually subject to the Courant, Friedrichs, and Lewy (CFL) condition

$$\frac{u_{\text{max}} \Delta t}{h} \leq 1, \quad (3.9)$$

where $u_{\text{max}}$ is the maximum velocity. Equation (3.9) states that $\Delta t$ must be sufficiently small so that a material point travels less than one grid spacing during a time step. This condition is derived using one-dimensional flow. For multidimensional schemes the limitation can be more restrictive, but advanced methods for the advection terms are frequently applied using splitting—where advection in each spatial direction is done sequentially—and then the one-dimensional limitations holds.

The time-integration scheme outlined by equations (3.4), (3.5) and (3.6) is only first order, as already observed. The simplest way to generate a higher-order method for the integration in time, is by a second-order predictor-corrector method where the first-order explicit method described above is used to find a temporary velocity $u^{\text{tmp}}$. The temporary velocity is then used to compute an approximate right-hand side (RHS) of equation (3.2) at time $n+1$ so that a second-order approximation for the new velocity can be found by the trapezoidal rule

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{1}{2} \left( \text{RHS}^n + \text{RHS}^{\text{tmp}} \right). \quad (3.10)$$

Here $\text{RHS}^n$ is computed using $u^n$ and $\text{RHS}^{\text{tmp}}$ is found using $u^{\text{tmp}}$. For actual programming, it is easier to take two first-order steps and then average $u^n$ and $u^{n+2}$. Doing this is easily shown to be equivalent to the above scheme and makes it particularly simple to extend a first-order scheme to second order. This method is generally subject to the same stability limits as the first-order explicit method and is, in particular, subject to the diffusion limitation imposed by equation (3.7). For low Reynolds numbers, explicit
3.1 Time integration

Treatment of the viscous terms results in impractically small time steps and it is better to treat those terms implicitly.

A large number of methods have been developed to solve the Navier-Stokes equations with second-order accuracy in time. Many of the most popular versions treat the viscous terms implicitly and the advection terms explicitly. We outline below the method of Kim and Moin (1985), extended to flows with variable material properties. The Kim and Moin method is a two-step projection method where in the first step the advection terms are computed explicitly using an Adams-Bashforth scheme and half of the viscous terms are implicit in the temporary velocity:

\[
\frac{u^* - u^n}{\Delta t} = -\left(\frac{3}{2} A_h^{n-1} - \frac{1}{2} A_h^n\right) + \frac{1}{2} \rho^n \left(D_h^n + D_h^*\right).
\] (3.11)

In the second step the velocity is corrected by adding the gradient of a pressure-like variable \(\Phi\):

\[
\frac{u^{n+1} - u^*}{\Delta t} = -\nabla \Phi.
\] (3.12)

An equation for \(\Phi\) is derived by combining (3.12) and (3.3), resulting in an equation that looks like (3.6) but with \(p\) replaced by \(\Phi\). To see exactly what \(\Phi\) is we add equations (3.11) and (3.12), yielding:

\[
\frac{u^{n+1} - u^n}{\Delta t} = -\left(\frac{3}{2} A_h^{n-1} - \frac{1}{2} A_h^n\right) + \frac{1}{\rho^n} \left[\frac{1}{2} (D_h^n + D_h^{n+1}) + \frac{1}{2} (D_h^* - D_h^{n+1}) - \nabla \Phi\right].
\]

Here we have added and subtracted \(D_h^{n+1}\) to allow a comparison with an Adams-Bashforth/Crank-Nicholson method where the viscous terms are evaluated using the velocities at the new time, \(u^{n+1}\), instead of the temporary velocities \(u^*\). From this we see that \(\Phi\) is related to \(p\) by

\[
\frac{1}{2} (D^* - D^{n+1}) - \nabla \Phi = -\nabla p.
\] (3.13)

Kim and Moin used a staggered grid, second order spatial differences and solved (3.11), which is implicit, by splitting.

The projection method is not the only way of integrating the Navier-Stokes equations in time. Essentially any standard method to solve the Navier-Stokes equations for homogeneous flows, such as the artificial compressibility method of Chorin (1967), the SIMPLE method of Patankar (1980), the PISO method of Issa (1985), and many others, can be adapted to multiphase flows, when the one-fluid formulation is used.
For flows where there are strong capillary effects, additional stability restrictions may apply. Usually, those are taken to be analogous to the CFL conditions and a capillary wave is allowed to move only one grid spacing during time $\Delta t$. The phase velocity of a capillary wave is:

$$c = \left( \frac{\sigma k}{\rho_1 + \rho_2} \right)^{1/2},$$

where $k$ is the wavenumber. The smallest resolved wave has $k = \pi / h$ and using this wave number and replacing $u_{\text{max}}$ in equation (3.9) by $c$, we find that the time step is limited by

$$\Delta t \leq \left( \frac{(\rho_1 + \rho_2)h^3}{\pi \sigma} \right)^{1/2}. \quad (3.15)$$

### 3.2 Spatial discretization

To discretize the momentum equations, we use the Finite-Volume approach where the conservation principles of mass and momentum are applied to a small control volume. While the governing equations in integral form, as shown in Fig. 2.2, are often taken as the starting point for the derivation of the discrete approximations, we can also simply integrate the differential form, as in Fig. 2.5, over the control volume. In either case, the velocity at the center of the control volume, $\mathbf{u}_c$, is approximated by the average value

$$\mathbf{u}_c = \frac{1}{V} \int_V \mathbf{u}(\mathbf{x}) dV. \quad (3.16)$$

Here $V$ is the volume of the control volume and for a two-dimensional rectangular control volume, $V = \Delta x \Delta y$. To find numerical approximations for the advection and the diffusion terms, we first define the average value over a control volume

$$A_c = \frac{1}{V} \int_V \nabla \cdot (\mathbf{u} \mathbf{u}) dV = \frac{1}{V} \int_S \mathbf{u} (\mathbf{u} \cdot \mathbf{n}) ds \quad (3.17)$$

and

$$D_c = \frac{1}{V} \int_V \nabla \cdot \mathbf{T}^v dV = \frac{1}{V} \int_S \mathbf{n} \cdot \mathbf{T}^v ds, \quad (3.18)$$

where $\mathbf{T}^v$ is the viscous stress tensor for incompressible flow

$$\mathbf{T}^v = \mu [\nabla_h \mathbf{u} + (\nabla_h \mathbf{u})^T]. \quad (3.19)$$

Here, we have used the divergence theorem to rewrite the volume integrals as surface integrals over the boundary $S$ of a control volume. Numerical
3.2 Spatial discretization

Fig. 3.1. The notation used for a standard staggered MAC mesh. The pressure is assumed to be known at the center of the control volume outlined by a thick solid line. The horizontal velocity components \( u \) are stored at the middle of the left and right edges of this control volume and the vertical velocity components \( v \) are stored at middle of the top and bottom edges.

approximations for these terms, \( A_h \) and \( D_h \), are obtained by evaluating the integrals numerically. The average value of the pressure gradient is:

\[
(\nabla p)_c = \frac{1}{V} \int_V \nabla p \, dv = \frac{1}{V} \oint_S p \, ds, \tag{3.20}
\]

and the average value of the body force is:

\[
f_c = \frac{1}{V} \int_V f(x) \, dv. \tag{3.21}
\]

The continuity equation is approximated in the integral form (equation 2.21) at time \( n + 1 \) by evaluating

\[
\oint_S \mathbf{u}^{n+1} \cdot \mathbf{n} \, ds = 0. \tag{3.22}
\]

Before we derive discrete approximations to the Navier-Stokes equations by approximating the integrals above, we have to decide what the control volumes look like and where the various variables, or rather the discrete approximations to these variables, are located with respect to each other. Generally, the discrete variables are put at the nodes of a regular grid and while it may seem most natural to store the pressure, the velocity, and the material properties of the fluid at the same grid node (usually referred as collocated grids), experience shows that for incompressible flows it is simpler to use a staggered grid where each of these variables is located on a separate
Numerical solutions of the Navier-Stokes equations

This is especially true if the grid lines are straight. For simplicity, we shall limit the discussion in the rest of this Chapter to two-dimensional flows. The extension to three dimensions is straightforward but makes the exposition more cumbersome.

The staggered grid was introduced by Harlow and Welch (1965) for the Marker and Cell (MAC) method and has been used extensively in computational fluid mechanics ever since. Fig. 3.1 shows the structure of the grid and the location of each variable for two-dimensional flow. The starting point is a control volume around the pressure points (outlined by a thick line). The role of pressure for incompressible flows is to force the divergence of the velocity field to be zero: the pressure must be raised if there is a net inflow into this control volume and lowered if there is net outflow. For the computation of the net flow in or out of the control volume around the pressure node we need the horizontal velocity components \( u \) at the vertical boundaries and the vertical velocity components \( v \) on the horizontal boundaries. It is natural to locate the velocity components at the middle of the boundaries, where we need them, and the grid for the horizontal velocity is therefore displaced half a mesh to the right from the pressure node and the vertical velocity grid is displaced half a mesh upward (see Fig. 3.2). It is customary to identify the pressure nodes by the indices \((i, j)\) and to refer to the location of the \( u \)-velocity component by \((i + 1/2, j)\) and the location of the \( v \)-velocity component by \((i, j + 1/2)\). In an actual computer code the grids are, of course, simply shifted and each component referenced by an integer. The density and other material properties are usually stored at the pressure nodes.

On the staggered grid shown in Fig. 3.1, the discrete form of the continuity equation is obtained by writing the integral in equation (3.22) for each side of the control volume

\[
0 = \frac{1}{\Delta x \Delta y} \int_S \mathbf{u}^{n+1} \cdot \mathbf{n} \, ds = \frac{1}{\Delta x \Delta y} \left\{ \int_{i+1/2,j} u^{n+1} dy \right. \\
- \left. \int_{i-1/2,j} u^{n+1} dy + \int_{i,j+1/2} v^{n+1} dx - \int_{i,j-1/2} v^{n+1} dx \right\},
\]

and then by approximating the integrals by the midpoint rule:

\[
\frac{1}{\Delta x \Delta y} \left[ \left( u_{i+1/2,j}^{n+1} - u_{i-1/2,j}^{n+1} \right) \Delta y + \left( v_{i,j+1/2}^{n+1} - v_{i,j-1/2}^{n+1} \right) \Delta x \right] = 0. \tag{3.24}
\]
3.2 Spatial discretization

Rewriting this equation slightly, the final result is:

\[
\frac{u_{i+1/2,j}^{n+1} - u_{i-1/2,j}^{n+1}}{\Delta x} + \frac{v_{i,j+1/2}^{n+1} - v_{i,j-1/2}^{n+1}}{\Delta y} = 0,
\]

(3.25)

which could also have been obtained by discretizing equation (3.3) with finite differences.

Using the grid in Fig. 3.2 and the notation introduced above, the discrete approximations for the \(x\) and \(y\) component of the predicted velocities (equation 3.4) are:

\[
u_{i+1/2,j}^{n*} = u_{i+1/2,j}^{n} + \Delta t \left\{ \left(-A_g\right)_{i+1/2,j}^{n} + \left(f_{bx}\right)_{i+1/2,j}^{n} \right\}
\]

(3.26)

\[\frac{1}{\left(\rho_{i+1,j}^{n} + \rho_{i,j}^{n}\right)} \left(\left(D_x\right)_{i+1/2,j}^{n} + \left(f_{ax}\right)_{i+1/2,j}^{n}\right)\]

and

\[
u_{i,j+1/2}^{n*} = v_{i,j+1/2}^{n} + \Delta t \left\{ \left(-A_g\right)_{i,j+1/2}^{n} + \left(f_{by}\right)_{i,j+1/2}^{n} \right\}
\]

(3.27)

\[\frac{1}{\left(\rho_{i,j+1}^{n} + \rho_{i,j}^{n}\right)} \left(\left(D_y\right)_{i,j+1/2}^{n} + \left(f_{ay}\right)_{i,j+1/2}^{n}\right)\].
The equations for the projected velocities (equation 3.5) are:

\[
u^{n+1}_{i+1/2,j} = u^*_i + \frac{\Delta t}{\frac{1}{2} \left( \rho^n_{i+1,j} + \rho^n_{i,j} \right)} \frac{p_{i+1,j} - p_{i,j}}{\Delta x} \tag{3.28}\]

and

\[
v^{n+1}_{i,j+1/2} = v^*_j + \frac{\Delta t}{\frac{1}{2} \left( \rho^n_{i,j+1} + \rho^n_{i,j} \right)} \frac{p_{i,j+1} - p_{i,j}}{\Delta y}. \tag{3.29}\]

We note that some of the variables in the equations above, such as the density, are at locations where they are not defined. For those values, we use linear interpolation (by taking the average).

There are three main reasons why the staggered grid is so widely used instead of collocated grids where all the variable are located at the same point. One is accuracy. Since the pressure gradient is computed as the difference between adjacent points, versus points two \(\Delta x\) or \(\Delta y\) apart when a collocated grid is used, the mesh is in effect finer. However, since other derivatives are evaluated using grid points further apart the result are not as accurate as if a twice as fine collocated grid were used. The second reason is that it is relatively simple to produce conservative methods when working on a staggered grid. The third, and perhaps the most important advantage of a staggered grid, is that it results in a tighter coupling between the variables than if they were all located at the same grid node. The continuity equation, when written on a collocated grid, links every other grid point and it is possible to satisfy continuity exactly, yet have highly fluctuating velocities. Although there are ways to overcome this problem for collocated grids (see Rhie and Chow, 1983, for example), the staggered grid provides a simpler approach.

### 3.3 Discretization of the advection terms

In the original MAC method, centered differencing was used for all spatial variables and the time integration was done by the simple explicit first-order projection method described above. Later implementations used first-order upwind or the so-called donor-cell method for the advection terms. Here we will first present the second-order centered scheme and then discuss other alternatives.

The discrete approximation of the \(x\)-component of the advection term is found by approximating the integral in equation (3.17) by the midpoint rule:

\[
(A_x)_{i+1/2,j} = \tag{3.30}
\]
3.3 Discretization of the advection terms

\[
\frac{1}{\Delta x \Delta y} \left\{ [(uu)_{i+1,j} - (uu)_{i,j}] \Delta y + [(uv)_{i+1/2,j+1/2} - (uv)_{i+1/2,j-1/2}] \Delta x \right\}
\]

and the \( y \)-component is approximated by:

\[
(A_y)_{i,j+1/2} = \frac{1}{\Delta x \Delta y} \left\{ [(uv)_{i+1/2,j+1/2} - (uv)_{i-1/2,j+1/2}] \Delta y + [(vv)_{i,j-1} - (vv)_{i,j}] \Delta x \right\}. \tag{3.31}
\]

The velocities are given at the center of the respective control volumes (Fig. 3.2), but the momentum fluxes in (3.30) and (3.31) are computed at the boundaries of the control volumes. The different way of finding the velocities for the momentum-flux calculations differentiates between the different numerical schemes.

In the centered, second-order scheme, the velocities at the boundaries of the velocity control volumes are found by linear interpolation and the momentum fluxes computed using these interpolated values. The \( x \)-component at \((i + 1/2, j)\) is then

\[
(A_x)_{i+1/2,j} = \frac{1}{\Delta x} \left[ \left( \frac{u^n_{i+1/2,j} + u^n_{i+1/2,j+1}}{2} \right)^2 - \left( \frac{u^n_{i+1/2,j} + u^n_{i-1/2,j}}{2} \right)^2 \right] + \frac{1}{\Delta y} \left[ \left( \frac{u^n_{i+1/2,j+1} + u^n_{i+1/2,j}}{2} \right) \left( \frac{v^n_{i+1/2,j+1} + v^n_{i,j+1/2}}{2} \right) - \left( \frac{u^n_{i+1/2,j} + u^n_{i+1/2,j-1}}{2} \right) \left( \frac{v^n_{i+1/2,j-1} + v^n_{i,j-1/2}}{2} \right) \right].
\]

The \( y \)-component, at the \((i, j + 1/2)\) point, is:

\[
(A_y)_{i,j+1/2} = \frac{1}{\Delta x} \left[ \left( \frac{u^n_{i+1/2,j} + u^n_{i+1/2,j+1}}{2} \right) \left( \frac{v^n_{i,j+1/2} + v^n_{i+1/2,j}}{2} \right) - \left( \frac{u^n_{i-1/2,j+1} + u^n_{i-1/2,j}}{2} \right) \left( \frac{v^n_{i,j+1/2} + v^n_{i-1,j+1/2}}{2} \right) \right] + \frac{1}{\Delta y} \left[ \left( \frac{v^n_{i,j+1/2} + v^n_{i,j+1/2}}{2} \right)^2 - \left( \frac{u^n_{i,j+1/2} + u^n_{i,j-1/2}}{2} \right)^2 \right]. \tag{3.33}
\]

The centered second-order scheme is more accurate than any non-centered
Numerical solutions of the Navier-Stokes equations

second-order scheme and usually gives the best results for fully resolved flows. It has, however, two serious shortcomings. The first problem is that for flows that are not fully resolved it can produce unphysical oscillations that can degrade the quality of the results. The second problem is that the centered second-order scheme is unconditionally unstable for inviscid flows when used in combination with the explicit forward-in-time integration given by equation (3.2). It is only the addition of the diffusion terms that makes the scheme stable and if diffusion is small, the time-step must be small. At high Reynolds numbers this usually results in excessively small timesteps.

The velocity at the boundaries of the velocity control volumes, \( u_{i,j} \), for example, can be found in many other ways. Each approach leads to a new scheme whose properties are different from other schemes. The simplest way to generate a robust scheme that is stable in the limit of zero viscosity is to use upwinding to find the edge velocity:

\[
u_{i,j} = \begin{cases} 
  u_{i-1/2,j}, & \text{if } \frac{1}{2}(u_{i-1/2,j} + u_{i+1/2,j}) > 0; \\
  u_{i+1/2,j}, & \text{if } \frac{1}{2}(u_{i-1/2,j} + u_{i+1/2,j}) < 0.
\end{cases}
\]  

(3.34)

While very robust, upwinding is only first order accurate and leads to excessive numerical diffusion, as we will see in the next Chapter. The accuracy is improved significantly by using a third-order upwind biased polynomial:

\[
u_{i,j} = \begin{cases} 
  \left(\frac{1}{8}\right)(3u_{i+1/2,j} + 6u_{i-1/2,j} - u_{i-3/2,j}), & \text{if } \frac{1}{2}(u_{i-1/2,j} + u_{i+1/2,j}) > 0; \\
  \left(\frac{1}{8}\right)(3u_{i-1/2,j} + 6u_{i+1/2,j} - u_{i+3/2,j}), & \text{if } \frac{1}{2}(u_{i-1/2,j} + u_{i+1/2,j}) < 0.
\end{cases}
\]  

(3.35)

This is the so-called QUICK (Quadratic Upstream Interpolation for Convective Kinematics) scheme of Leonard (1979). QUICK and its variants are not completely free of “wiggles” for steep-enough gradients, but they are much more robust than the centered difference scheme and much more accurate than the first-order upwind method. Even more robust, the ENO (Essentially Non-Oscillating) scheme of Shu and Osher (1989) has been used by a number of authors. In this scheme the edge value is extrapolated from the known values at the center of the control volumes by:

\[
u_{i,j} = \begin{cases} 
  u_{i-1/2,j} + s_i \left(\frac{\Delta x}{2}\right), & \text{if } \frac{1}{2}(u_{i-1/2,j} + u_{i+1/2,j}) > 0; \\
  u_{i+1/2,j} - s_{i+1} \left(\frac{\Delta x}{2}\right), & \text{if } \frac{1}{2}(u_{i-1/2,j} + u_{i+1/2,j}) < 0.
\end{cases}
\]  

(3.36)

The slope \( s_i \) is the smallest (in terms of absolute value) of

\[
s_i^+ = \frac{(u_{i+1/2,j} - u_{i-1/2,j})}{\Delta x}, \\
s_i^- = \frac{(u_{i-1/2,j} - u_{i-3/2,j})}{\Delta x}.
\]  

(3.37)
This selection of the slopes in the second-order ENO scheme is the same as that used in the MINMOD limiter (Sweby, 1984). A variant of this approach, the Bell-Colella-Glaz (BCG) (Bell et al., 1989) scheme (see also Martin, 1998) is used in the Gerris code (see Popinet, 2003, 2008, 2009) used for several of the simulations shown in this Book. While the edge velocities found using either QUICK or ENO can be used with the conservative discretization in equation (2.2), they are also often used with a discretization of the non-conservative version of the advection terms (2.5). Both QUICK and ENO greatly improve the robustness of the computation of the advection terms for low viscosities, while yielding accuracy that is comparable to the centered difference scheme for viscosities where both give reliable results.

### 3.4 The viscous terms

We approximate the integral of the viscous fluxes (equation 3.18) at the boundaries of the velocity control volumes shown on Fig. 3.3 by the midpoint rule. This results in

\[
(D_x)_i^{n+1/2} = \frac{T_{i+1,j}^{v,xx} - T_{i,j}^{v,xx}}{\Delta x} + \frac{T_{i+1/2,j+1/2}^{v,xy} - T_{i+1/2,j-1/2}^{v,xy}}{\Delta y}
\]  

(3.38)

and

\[
(D_y)_i^{n+1/2} = \frac{T_{i+1/2,j+1/2}^{v,xy} - T_{i-1/2,j+1/2}^{v,xy}}{\Delta x} + \frac{T_{i,j+1}^{v,yy} - T_{i,j}^{v,yy}}{\Delta y}
\]  

(3.39)

The velocity derivatives that appear in the viscous stress tensor $T_v$ are found using the standard second-order centered differences, resulting in

\[
T_{i,j}^{v,xx} = \mu_{i,j}^{n} \frac{u_{i+1/2,j}^{n} - u_{i-1/2,j}^{n}}{\Delta x},
\]

\[
T_{i+1/2,j+1/2}^{v,xy} = \mu_{i+1/2,j+1/2}^{n} \left( \frac{u_{i+1/2,j+1}^{n} - u_{i+1/2,j}^{n}}{\Delta y} + \frac{v_{i+1,j+1/2}^{n} - v_{i+1,j+1/2}^{n}}{\Delta x} \right),
\]

\[
T_{i,j+1}^{v,yy} = \mu_{i,j+1}^{n} \frac{v_{i+1/2,j+1}^{n} - v_{i+1/2,j-1/2}^{n}}{\Delta y}.
\]  

(3.40)

In equations (3.40) we account for the fact that the viscosity coefficient $\mu$ may be space-dependent. If the viscosity is constant in each fluid, but changes discontinuously across the interface, it is, in principle, given by

\[
\mu(x) = \mu_1 H(x) + \mu_2 [1 - H(x)],
\]  

(3.41)

where $H$ is the Heaviside function defined in Section 2.3. In many cases we set the viscosity using this equation, with $H$ replaced by a marker function.
approximating it. In a Volume-of-Fluid method, for instance, one would compute the viscosity at location \((i, j)\) as

\[
\mu_{i,j} = \mu_1 C_{i,j} + \mu_2 (1 - C_{i,j}),
\]

where the Volume-of-Fluid color function \(C_{i,j}\) approximates \(H(x)\). In other methods, typically Front-Tracking or Level-Set methods, one would first construct a smooth indicator function and then use it as an approximation for \(H\). For small viscosity differences this approach generally works well. However, for larger differences this approach can result in an error since a discontinuous viscosity implies, by virtue of the equations in Fig. 2.14, a discontinuous velocity derivative, so that the approximation of the derivatives in (3.40) is not consistent.

For higher accuracy we need to do something else. One option is to use one-sided approximations of the velocity derivatives that use only the data on a given side of the interface. A few authors have implemented this idea. Another option is to use harmonic means as advocated by Patankar (1980) in the relatively simpler case of heat diffusion, or by Coward et al. (1997) for viscous shear flows. To explain the idea behind the harmonic mean we consider the simple two-phase parallel shear flow of Fig. 3.4. For simplicity
3.4 The viscous terms

![Diagram of a simple two-phase parallel shear flow with a horizontal interface.](image)

Fig. 3.4. A simple two-phase parallel shear flow with a horizontal interface. Because of the viscosity jump the velocity gradient is discontinuous.

The interface is assumed to be horizontal. We need to compute the shear stress at points \((i + 1/2, j - 1/2)\) and \((i + 1/2, j + 1/2)\). The horizontal velocity is given at levels \(j - 1, j, \) and \(j + 1\). Because the region between the two levels \(j - 1\) and \(j\) is completely filled with a single phase, say fluid 1, the stress located at the intermediate level \(j - 1/2\) can easily be computed to be

\[
\tau_{j-1/2} = \mu_1 \frac{u_j - u_{j-1}}{\Delta y}, \tag{3.43}
\]

where we use the compact notation \(\tau_{j-1/2} = T^{u,xy}_{i+1/2,j-1/2}\) and \(u_j = u_{i+1/2,j}\).

On the other hand the stress at level \(j + 1/2\) must be computed using an effective viscosity \(\mu_{i+1/2,j+1/2}\) so that

\[
\tau_{j+1/2} = \mu_{i+1/2,j+1/2} \frac{u_{j+1} - u_j}{\Delta y}. \tag{3.44}
\]

What is the correct value of \(\mu_{i+1/2,j+1/2}\)? First we notice that in a shear flow the stress is not only continuous but constant: \(\tau_{j-1/2} = \tau_{j+1/2} = \tau\).

Second, it is useful to introduce the velocity \(u_I = u(y_I)\) on the interface at \(y_I = y_j + h\), where \(h\) is the height of the interface above the grid line \(j\). Then

\[
\tau = \mu_2 \frac{u_{j+1} - u_I}{\Delta y - h} = \mu_1 \frac{u_I - u_j}{h}. \tag{3.45}
\]

Hence

\[
u_{j+1} - u_j = u_{j+1} - u_I + u_I - u_j = \frac{\Delta y - h}{\mu_2} + \frac{h}{\mu_1} \tag{3.46}
\]
and using (3.44)

$$\mu_{i+1/2,j+1/2} = \frac{\tau \Delta y}{u_{j+1} - u_j} = \Delta y \left( \frac{h}{\mu_1} + \frac{\Delta y - h}{\mu_2} \right)^{-1}. \quad (3.47)$$

Notice that $h/\Delta y$ is the fraction of phase 1 in the control volume centered around $(i + 1/2, j + 1/2)$ (see Fig. 3.4), so we can write it using a color function notation $C_{i,j+1/2} = C_{i+1,j+1/2} = C_{j+1/2}$, where we are consistent with the hypothesis of a horizontal interface and assume that there is no dependence on index $i$. Then the mixed-cell viscosity becomes

$$\mu_{i+1/2,j+1/2} = \left( \frac{\phi}{\mu_1} + \frac{1 - \phi}{\mu_2} \right)^{-1}, \quad (3.48)$$

a typical harmonic mean, where $\phi = C_{j+1/2}$. However the VOF color function $C_{i,j}$ is defined at pressure grid points $(i, j)$ (see Chapter 5). Thus we may use the second-order approximation $\phi = h/\Delta y = C_j + C_{j+1} - 1/2$. The interface is actually located between $j$ and $j + 1$, as we have assumed, whenever $1/2 \leq C_j + C_{j+1} \leq 3/2$, in other cases the single-phase formulas, for instance (3.43), must be used. An example of the use of the harmonic mean is given in Chapter 9.

In general the problem is more complex than the simple case above where the interface is aligned both with the grid axis and the flow. The harmonic mean is not a panacea, for instance if the horizontal interface in Fig. 3.4 is replaced by a vertical interface, the arithmetic mean (3.42) becomes exact. A consistent second-order method for this problem has not yet been published.

In practice, interfaces tend to be aligned with the flow direction in a shear flow so the harmonic mean is more accurate. It may however be less robust: in flows with large density differences, the arithmetic mean “favors” the largest viscosity while the harmonic mean “favors” the smallest viscosity. The arithmetic mean thus displaces the “effective” interface from its true position towards the small viscosity region. It thus creates a larger viscosity region around the interface, which may have the effect of “protecting” the interface against its destruction by short wavelength physical or numerical instabilities.

In this section we have assumed that the viscous terms are computed explicitly. For an implicit treatment the subscript $n$ must be replaced by $n + 1$. 


Fig. 3.5. The pressure equation next to a vertical boundary is derived using the continuity equation for a control volume where one side (the wall) has a given normal velocity.

### 3.5 The pressure equation

To derive a discrete Poisson equation for the pressure, we substitute equations (3.28) and (3.29), for the corrected velocity components at the new time, into equation (3.25) for the continuity of the new velocities. This yields

\[
\begin{align*}
\frac{1}{\Delta x^2} \left( \frac{p_{i+1,j} - p_{i,j}}{\rho_{i+1,j}^n + \rho_{i,j}^n} - \frac{p_{i,j} - p_{i-1,j}}{\rho_{i,j+1}^n + \rho_{i,j-1}^n} \right) + \frac{1}{\Delta y^2} \left( \frac{p_{i,j+1} - p_{i,j}}{\rho_{i,j+1}^n + \rho_{i,j}^n} - \frac{p_{i,j} - p_{i,j-1}}{\rho_{i,j}^n + \rho_{i,j-1}^n} \right) \\
= \frac{1}{2\Delta t} \left( \frac{u_{i+1/2,j}^n - u_{i-1/2,j}^n}{\Delta x} + \frac{v_{i,j+1/2}^n - v_{i,j-1/2}^n}{\Delta y} \right) \quad (3.49)\end{align*}
\]

The pressure equation must be solved using the appropriate boundary conditions and the staggered mesh makes the derivation of the appropriate boundary conditions particularly straightforward. Consider the vertical boundary in Fig. 3.5, on the left side of the domain and passing through node \((i-1/2, j)\), where the horizontal velocity is \(U_{b,j}\). The continuity equation for the cell next to the boundary, for the cell surrounding the pressure node \((i, j)\), is

\[
\frac{u_{i+1/2,j}^n - U_{b,j}}{\Delta x} + \frac{v_{i,j+1/2}^n - v_{i,j-1/2}^n}{\Delta y} = 0. \quad (3.50)
\]

Since one of the velocities is known, we only substitute the equations for the correction velocities (3.28 and 3.29) for the three unknown velocities,
through the top, bottom and right edge. Thus, we have

$$\frac{1}{\Delta x^2} \left( \frac{p_{i+1,j} - p_{i,j}}{\rho_{i+1,j}^n + \rho_{i,j}^n} \right) + \frac{1}{\Delta y^2} \left( \frac{p_{i,j+1} - p_{i,j}}{\rho_{i,j+1}^n + \rho_{i,j}^n} - \frac{p_{i,j} - p_{i,j-1}}{\rho_{i,j}^n + \rho_{i,j-1}^n} \right) = \frac{1}{2\Delta t} \left( \frac{u_{i+1/2,j}^* - U_{b,j}}{\Delta x} + \frac{v_{i,j+1/2}^* - v_{i,j-1/2}^*}{\Delta y} \right)$$

(3.51)

for the pressure node next to the boundary. Similar equations are derived for the pressure next to the other boundaries and for each corner point. Notice that it is not necessary to impose any new conditions on the pressure at the boundaries and that simply using incompressibility yields the correct boundary equations.

Since the value of the pressure is not specified at the boundaries, equations (3.49) and (3.51) do not determine the pressure uniquely and we can add an arbitrary constant to the solution. For many iterative techniques this constant is set by the average value of the initial guess and the iteration can be carried out without explicitly enforcing the constant. In other cases, the average pressure, or the pressure at one point must be specified.

We also note that when only inflow or no-through-flow boundary conditions are used it must be verified that no net flow enters the domain, so that \( \sum U_b \cdot n = 0 \), where the sum is over boundary nodes. Otherwise the pressure solver will never converge.

The boundary conditions given by equation (3.51) apply to solid walls and boundaries where the inflow is specified. For outflow boundaries, see Section 3.7.

Solving the pressure equation (3.49), with the appropriate boundary conditions, (3.51), can be done in many ways. During code development and for preliminary runs, it is usually more convenient to use a simple Successive Over-Relaxation (SOR) iteration method. To do so, equation (3.49) is rearranged to isolate \( p_{i,j} \) on the left hand side. The pressure is then updated iteratively by substituting on the right-hand side the approximate values \( p^\alpha \) of the pressure from the previous iteration and by taking a weighted average of the updated value and the pressure \( p_{i,j}^\alpha \) from the last iteration to compute
the new approximation $p_{i,j}^{n+1}$:

$$
p_{i,j}^{n+1} = \beta \left[ \frac{1}{\Delta x^2} \left( \frac{1}{\rho_{i+1,j}^n + \rho_{i,j}^n} + \frac{1}{\rho_{i,j}^n + \rho_{i-1,j}^n} \right) \\
+ \frac{1}{\Delta y^2} \left( \frac{1}{\rho_{i,j+1}^n + \rho_{i,j}^n} + \frac{1}{\rho_{i,j}^n + \rho_{i,j-1}^n} \right) \right]^{-1} \\
\left[ \frac{1}{\Delta x^2} \left( \frac{p_{i+1,j}^n}{\rho_{i+1,j}^n + \rho_{i,j}^n} + \frac{p_{i-1,j}^n}{\rho_{i,j}^n + \rho_{i-1,j}^n} \right) + \frac{1}{\Delta y^2} \left( \frac{p_{i,j+1}^n}{\rho_{i,j+1}^n + \rho_{i,j}^n} + \frac{p_{i,j-1}^n}{\rho_{i,j}^n + \rho_{i,j-1}^n} \right) \right] + (1 - \beta)p_{i,j}^n.
$$

Here, $\beta$ is a relaxation parameter and for over-relaxation we must have $\beta > 1$. For stability reasons we must also have $\beta < 2$. Taking $\beta = 1.2 - 1.5$ is usually a good compromise between stability and accelerated convergence. $p_{i,j}$ is also isolated in the same way in the pressure equations for boundary and corner points. Initially, the pressure can be set to zero, but once the time integration has started, the value of the pressure from the last time step usually provides a good first estimate. For vector computers the SOR process needs to be modified slightly, resulting in the so-called Black and Red SOR.

The chief merit of the SOR iteration is its simplicity. The solution of the pressure equation is generally the most time-consuming part of any simulation of incompressible flows and the SOR converges too slowly for resource-intensive computations. Thus, while we recommend the use of SOR during code development (or when a small problem has to be solved infrequently), for production runs it is necessary to use more advanced methods.

The development of efficient solution methods for elliptic equations is a highly developed field and a large number of methods have been proposed. Advanced techniques include fast Poisson solvers based on the discrete fast Fourier transform or cyclic reduction, multigrid methods, and advanced iterative Krylov-subspace methods. Fast Poisson solvers were immensely popular in the eighties, in part due to the freely available and popular routines in FISHPAK (Swarztrauber and Sweet, 1979) but these methods are limited to simple boundary conditions and separable elliptic equations. They cannot therefore be used when the density changes, as in equation (3.49).

Multigrid methods, developed in the late eighties, (Briggs, 1987; Wesseling, 1992) have been used by many authors to solve the pressure equation for multifluid flow simulations. The fundamental idea behind multigrid solvers
is perhaps best understood by looking at the iterative solution of the Poisson
equation as an explicit “pseudo” time integration of a diffusion equation to
a steady state solution. To obtain convergence (steady-state) as quickly as
possible, we would like to use the maximum allowable time step. Elementary
analysis shows that the maximum allowable time step is proportional
to $\Delta x^2$ (or $\Delta y^2$), so more time steps must be used for fine grids. If we
write the difference between the initial conditions and the final steady state
as a Fourier series, it is easily seen that the decay rate is proportional to the
square of the wave number. The high-wavenumber components of the solution
die out quickly and the time needed to reach steady state is controlled
by the decay rate of the low-wavenumber modes. Thus, if we use a fine grid,
we are forced to take small steps to accurately resolve wavenumbers that do
not exist after a few steps! This suggests that the convergence rate can be
improved considerably by using a coarser grid, where large time steps can be
used, for the slowly decaying low-wavenumber modes. In multigrid methods
we generally start on a fine grid and iterate until the high wavenumber com-
ponents of the error have decayed and the rate of convergence becomes slow.
The approximate solution is then transferred to a coarser grid and an equa-
tion for the correction iterated until the convergence rate slows down. The
process is repeated on coarser and coarser grids until the grid is so coarse
that the equations can be solved exactly. The corrections are then added
to the solution found on the finer grids and the equations iterated again
to eliminate errors introduced by the transfer between the grids. Once all
the corrections have been added to the results on the finer grids, we should
only need a few iterations to converge fully. In practice, it is usually more
efficient to adopt more complex strategies where the solution is transferred
more frequently between the coarse and the fine grids.

Although multigrid methods have been used for a large number of simula-
tions of multiphase flows, it is often found that multigrid solvers have diffi-
culty converging for very high density ratios. Another option is therefore to
use advanced Krylov methods to solve the pressure equation. Moreover, sev-
eral researchers have recently been using a combination of advanced Krylov
methods and multigrid, where several multigrid iterations are used as a pre-
conditioner for the Krylov method. Krylov methods seek to minimize the
residual iteratively using only the multiplication of a matrix with a vector.
The SOR method is an example of a stationary Krylov method, but in non-
stationary methods the search direction is determined based on the current
residual. Practical implementations of the Krylov methods almost always
involve the use of a preconditioner where the original operator is replaced
by a simpler operator that in some ways preserves the properties of the
original operator. Early Krylov methods include the Conjugate-Gradient method, but more recent methods are generally based on the Generalized-Minimal-RESidual method (GMRES). For each iteration, GMRES requires the results from every previous iteration to produce a new approximation to the solution and the required memory and work therefore increases with the number of iterations. Nevertheless, it has been used successfully in multiphase calculations (Francois et al., 2006). A large number of methods have been developed to reduce the memory and work requirement, but as of this writing it looks like there is no “best” method for all problems. While it is possible that there is a method that is optimum for the rather special situation that we are interested in—a Poisson equation with coefficients that change discontinuously—the optimal method has yet to be discovered. Methods that have been used successfully, however, include the Bi-Conjugate Gradient STABilized (BiCGSTAB) method of van der Vorst (1992, 2003) used for example by Gerlach et al. (2006b). Takahira et al. (2004) claim that density ratios as large as $10^4$ may be reached using BiCGSTAB.

The ease by which the pressure equation is solved depends generally on the density jump. While a SOR method with a low enough overrelaxation parameter always allows us to solve the pressure equation, it can be very slow and more advanced methods can fail to converge. Furthermore, small errors in the indicator or marker function can lead to negative densities that in turn usually cause convergence difficulties. These problems are, however, eliminated relatively easily by minor filtering. For many problems, such as for bubbles, the exact density difference plays a relatively minor role, once it is large enough, and it is possible to speed up the calculations by approximating the density of the bubbles by a value that is much higher than the real one. For droplets subject to aerodynamic forces, as in atomisation, droplet impact or wave breaking problems, the exact density ratios must, however, be used.

### 3.6 Velocity boundary conditions

Although the treatment of the pressure boundary conditions, as well as the normal velocities, simplifies considerably on a staggered mesh, the tangential velocity boundary conditions become a little more complicated. When $v_{i,j+1/2}$ in Fig. 3.6 (next to a left boundary) is updated, the velocity at the node to the left, $v_{i-1,j+1/2}$, is needed. This node is, however, outside the boundary. The easiest way to implement the boundary conditions for the tangent velocity is by the use of ghost points. The value of the tangential
Numerical solutions of the Navier-Stokes equations

velocity at the ghost point is obtained by treating it as regular point and using the fact that for no-slip boundary conditions the fluid velocity at the boundary is equal to the wall velocity. A linear interpolation gives

\[ v_{bdry,j+1/2} = \frac{1}{2}(v_{i,j+1/2} + v_{i-1,j+1/2}), \] (3.53)

where \( v_{bdry,j+1/2} \) is the tangential velocity of the wall. Solving for the velocity at the ghost point yields:

\[ v_{i-1,j+1/2} = 2v_{bdry,j+1/2} - v_{i,j+1/2}, \quad \text{no slip boundary.} \] (3.54)

Full-slip boundary conditions are usually implemented by simply putting the velocity at the ghost point equal to the velocity inside the domain. The derivative of the tangential velocity is then zero, assuming that we use a centered finite difference approximation. For the situation shown in Fig. 3.6

\[ v_{i-1,j+1/2} = v_{i,j+1/2}, \quad \text{full slip boundary.} \] (3.55)

3.7 Outflow boundary conditions

Often, we would like to simulate only the part of a flow field that is of interest to us, even though the domain is in reality much larger. In other cases, we are interested in a region with in and outflow. While it is often realistic to specify the inflow velocity exactly, the goal of the outflow boundary conditions is to let fluid leave the computational domain in realistic ways.

The simplest situation arises if it is physically justifiable to specify the outflow velocity at each grid point on the outflow boundary. Then the situation is exactly the same as for the inflow boundaries described above
Fig. 3.7. At an outflow boundary it is generally desirable to apply conditions that disturb the upstream flow as little as possible. If the outflow is perpendicular to the boundary, assuming that the normal velocity is constant across the boundary results in a reasonably “soft” outflow conditions.

and the implementation is straightforward. This condition is, however, too restrictive in most cases.

Although we stop following the flow computationally at an outflow boundary, the flow usually continues beyond it. To justify truncating the domain, we have to assume that whatever the flow does downstream, it has little impact upstream of the boundary. This usually implies that the downstream flow is smooth and uniform. Thus, the outflow boundary conditions should constrain the outflow as little as possible and let the fluid “flow freely” out of the domain.

One possibility is to assume that the streamlines at the outflow are perpendicular to the boundary. If the outflow takes place at the right boundary of a rectangular domain (Fig. 3.7), this means that $v = 0$ at the boundary.

From continuity it follows that $\partial u / \partial x = 0$ and for the grid shown in Fig. 3.7 the velocity of the boundary is therefore

$$u_{i+1/2,j} = u_{i-1/2,j}. \tag{3.56}$$

Thus we can sometimes simply use (3.56) along with $p_{i,j} = \text{constant}$ and $v_{i,j+1/2} = 0$. This approach results in a relatively “soft” outflow condition and does not, in particular, require the $u$ velocity to be uniform across the outflow boundary. In principle, $u$ can even change sign, allowing some inflow. However, the location of the outflow boundary will generally impact the computed solution if there is a significant variation in the outflow velocity.

Many other boundary conditions have been proposed in the literature to account for outflows in a physically plausible, yet computationally efficient way. A discussion of outflow boundary conditions can be found in section 6.5 of Wesseling (2001) and in Johansson (1993), for example.
In many cases it is convenient to make one or more coordinate directions periodic. If the domain is periodic in the $x$ direction and the length of the period is $L$, then $a(x, y, z) = a(x \pm mL, y, z)$ where $a$ is any variable and $m$ is any positive or negative integer. If the domain is discretized by $N$ cells in the $x$ direction and we have one ghost grid point on either side then the total number of grid points is $N + 2$. If the ghost point on the left is given by $i = 1$ and the one on the right by $i = N + 2$, then the value of $a$ at these points is given by $a(1, j, k) = a(N + 1, j, k)$ and $a(N + 2, j, k) = a(2, j, k)$, respectively.

3.8 Adaptive Mesh Refinement

For many multiphase flow problems—as for single phase flows—the resolution requirements vary greatly from one point to another. Allocating computational resources according to need can, at least in principle, result in significant savings compared with using a single grid with equal size cells. In spite of major efforts to develop fluid solvers based on unstructured, often arbitrarily shaped, control volumes, several decades of experience have shown that in terms of accuracy and computational efficiency it is hard to beat regular structured grids. The obvious solution is to use structured grids with curved grid lines that remain logically rectangular but whose grid lines can be pushed and pulled to regions of the computational domain where they are needed. This obvious and attractive approach has been used extensively for single phase flow in complex domains (but less so for multiphase flows, see Muradoglu and Kayaalp, 2006, for an exception) but it is now understood that since the number of grid points, in this approach, is fixed, the achievable adaptivity is limited and the method is only suitable for relatively simple problems.

Currently, the main choice for multiphase flows are techniques based on adding grid points by refining a regular structured grid locally. A single grid cell is usually split into four (in two dimensions) or eight (three dimensions) smaller cells, either by embedding a patch of finer grid into the coarser grid or by splitting each cell individually. In the latter case, which appears to be more popular, an octree data structure (quadtree in two dimensions) is used to organize the grid and to transfer information between cells. In most cases, the small cells can be split again, for increased resolution. This approach to grid adaptivity is usually referred to as Adaptive Mesh Refinement (AMR) although the name could be used for many other strategies for local refinement of the grid. Early examples of the use of octree/quadtree AMR include Front-Tracking simulations of biological cells by Agresar et al. (1998) and
the Level-Set simulations of the breakup of a jet by Sussman et al. (1999). More recent uses include simulations of underwater explosions by Kadio-
glu and Sussman (2008), done by a Level-Set/Volume-of-Fluid method and Hua and Lou (2007) who used a Front-Tracking method in conjunction with a
general purpose AMR code (PARAMESH, see MacNeice et al., 2000) to simulate the rise of bubbles. In Figure 3.8 we show one frame from a simulation of the atomization of a fuel jet using the open-source flow solver Gerris (see Popinet, 2003, 2008, 2009) where a Volume-of-Fluid interface-tracking scheme is coupled with an octree Adaptive Mesh Refinement. The use of AMR for multiphase flow simulations involves essentially no major issues beyond those encountered for single-phase flow simulations and we refer the reader to the already voluminous literature for more details.

3.9 Summary

In this Chapter, we have outlined a very standard numerical method to solve the Navier-Stokes equations numerically for flows with variable density and
viscosity. Combined with the techniques described in the next Chapters to
advect the phase boundary (and therefore the density and viscosity fields)
and to compute the surface tension, this results in a method capable of
simulating reasonably accurately a wide variety of multifluid flows. The
accuracy, robustness, and efficiency of the method can, however, be improved
in a number of ways and we have indicated briefly a few ways to do so.

While the equations presented in this Chapter are reasonably complex,
they generally do not result in a very lengthy code. More advanced treat-
ment, particularly of the advection terms, adds some complexity but fre-
quently it is not necessary to start completely “from scratch” when de-
veloping a new code. The equations of this Chapter have been implemented
by a large number of investigators and many codes and code “snippets” are
available as open source codes on the Internet. Using those as a starting
point can result in a considerably shorter development time.

Appendix: Conservative vs. non-conservative form
To gain some insight into the differences between the discretization of
the conservative and non-conservative form of the Navier-Stokes equations (Fig.
2.3 versus Fig. 2.4) we will consider the uniform one-dimensional motion of
a fluid with a sharp density interface. The pressure gradient and the viscous
terms can be taken to be zero so we only need to solve
\[
\frac{\partial p}{\partial t} + \frac{\partial p u^2}{\partial x} = 0
\]  
\[\text{(A.1)}\]
for the momentum and
\[
\frac{\partial p}{\partial t} + \frac{\partial p u}{\partial x} = 0
\]
\[\text{(A.2)}\]
for the density. We assume a collocated grid, so that \(u_i\) and \(\rho_i\) are given
at the same physical location. A simple forward in time, centered in space
scheme\(\dagger\) for the momentum equation lets us write
\[
u_i^{n+1} = \frac{1}{\rho_i^{n+1}} \left[ \rho_i^n u_i^n - \frac{\Delta t}{2h} (p u^2)^n_{i+1} - (p u^2)^n_{i-1} \right].
\]
\[\text{(A.3)}\]
The initial velocity field is constant, \(u_i = u_o\), so we can write this as
\[
u_i^{n+1} = \frac{u_o}{\rho_i^{n+1}} \left[ \rho_i^n - \frac{u_o \Delta t}{2h} (\rho_n^{i+1} - \rho_n^{i-1}) \right]
\]
\[\text{(A.4)}\]
and since the velocity should remain constant, the right-hand side should
be \(u_o\) which requires the parenthesis to be equal to \(\rho_i^{n+1}\). If we apply the
\(\dagger\) This is actually an unstable scheme, but that is immaterial for the argument being made here.
same scheme to the advection equation for the density, we find that

$$\rho_{i}^{n+1} = \rho_{i}^{n} - \frac{u_{o} \Delta t}{2h} (\rho_{i+1}^{n} - \rho_{i-1}^{n}), \quad (A.5)$$

using that the velocity is constant. The right hand side of this equation is exactly the same as the bracket on the right hand side of equation (A.4), showing that the velocity remains constant, as it should.

If the density is, however, advected in a different way, say by an upwind scheme, such that

$$\rho_{i}^{n+1} = \rho_{i}^{n} - \frac{u_{o} \Delta t}{h} (\rho_{i}^{n} - \rho_{i-1}^{n}), \quad (A.6)$$

then the same cancelation will obviously not take place and the velocity will not remain exactly equal to $u_{o}$. While the error is generally small, it is not exactly zero, particularly if the density jump is large. The same considerations hold for a staggered grid. Equation (A.1) can, however, be written in a non-conservative form as

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} = 0 \quad (A.7)$$

using equation (A.2). Any discretization of this form will preserve the constant velocity.

For methods where the density is generated from a marker function that is advected differently from how the momentum is updated, we are likely to encounter the situation described above and the non-conservative form of the equations is therefore generally preferred. Another way to avoid the problem is to first advect the density using a scheme consistent with the advection of the momentum and then re-advect the density using the more sophisticated scheme at the end of the time step.
When the governing equations are solved on a fixed grid, using one set of equations for the whole flow field, the different fluids must be identified in some way. This is generally done by using a marker function that takes different values in the different fluids. Sometimes a material property, such as the fluid density for incompressible fluids, can serve as a marker function, but here we shall assume that the role of the marker function is only to identify the different fluids. As the fluids move, and the boundary between the different fluids changes location, the marker function must be updated. Updating the marker function accurately is both critical for the success of simulations of multiphase flows and also surprisingly difficult. In this Chapter we discuss the difficulties with advecting the marker function directly and the various methods that have been developed to overcome these difficulties.

The Volume-of-Fluid (VOF) method is the oldest method to advect a marker function and—after many improvements and innovations—continues to be widely used. Other marker function methods include the Level-Set method, the Phase-Field method, and the CIP method. Instead of advecting the marker function directly, the boundary between the different fluids can also be tracked using marker points, and the marker function then reconstructed from the location of the interface. Methods using marker points are generally referred to as “front-tracking” methods to distinguish them from “front-capturing” methods where the marker function is advected directly. In addition to the difference in the way the marker function is updated, surface tension can be included in several different ways, as explained in Chapter 7.
4.1 Notations

To identify whether a given fluid \( i \) is present at a particular location \( x \), we use a Heaviside (step) function \( H \), defined by

\[
H_i(x) = \begin{cases} 
1, & \text{if } x \text{ is in fluid } i; \\
0, & \text{if } x \text{ is not in fluid } i.
\end{cases}
\]  

(4.1)

For a two fluid system, \( i = 1, 2 \), and \( H_2 = 1 - H_1 \) so it is sufficient to work with \( H = H_1 \). As the interface moves, the shape of the region occupied by each fluid changes, but each fluid particle retains its identity. Thus, the material derivative (following the motion of a fluid particle) of \( H \) is zero, or

\[
\frac{DH}{Dt} = \frac{\partial H}{\partial t} + \mathbf{u} \cdot \nabla H = 0.
\]  

(4.2)

Once \( H \) is known, the material properties of each fluid can be found, and the velocity updated as described in the last Chapter. The accurate advection of \( H \), or numerical approximations to \( H \), is the central topic of this and the next two Chapters.

When following the flow computationally, we have to work with an approximation of \( H \). Several approximations are possible. The color function \( C \) is defined as the average value of \( H \) in each computational cell. For a rectangular two-dimensional cell

\[
C_{i,j} = \frac{1}{\Delta x \Delta y} \int_{V} H(x,y) \, dx \, dy.
\]  

(4.3)

Cells away from the interface are either full, \( C = 1 \), or empty, \( C = 0 \), but if an interface is located somewhere in a given cell, \( C \) for that cell has a fractional value. Often, we shall find it useful to work with smoother versions of the color function, denoted here by the “indicator function” \( I \). It is convenient to think of \( I \) as a smoothed version of \( H \), obtained by

\[
I(x,y) = \int G(x-x',y-y') H(x',y') \, dx' \, dy',
\]  

(4.4)

where \( G \) is a smoothing kernel. Here we have assumed a two-dimensional domain. In practice, \( I \) is obtained by smoothing the color function or by constructing a smooth function directly from the location of the interface. Yet another possibility is to give up any attempts to link the shape of the marker function to the step function and simply use a smooth function \( F \), where \( F > 0 \) in one fluid and \( F < 0 \) in the other. The interface is therefore identified by the level-set \( F = 0 \). Various other approximations to \( H \) are in common use and while these are generally similar to \( C \) and \( I \) described above, there are often minor differences in their properties, their names, or
their physical interpretation. With this in mind, we will use $c$ for the phase-field variable in Section 4.6 and $f$ for the CIP marker function in Section 4.7.

Figure 4.1 shows a few different ways to identify the interface. The Heaviside function $H(x)$ is the discontinuous function shown by a solid line that changes abruptly from $H = 1$ to $H = 0$ in cell $j$, exactly where the interface is located. The color function $C_j$, the average value of the marker in each cell, is shown by the shaded area, and the smooth approximation $I(x)$ is shown by the dashed line. Notice that $H$ and $C$ are identical everywhere except in the interface cell $j$, but that $I$ approaches a uniform value more slowly. The level-set function $F(x)$, defined such that $|F| = d$ where $d$ is the distance from the interface, is shown by a dashed-dot line. While $H$, $I$, and $F$ obviously vary with $x$, $C_j$ is inherently identified with cell $j$. In the discussion below we will, however, sometimes talk about $C$ as a continuous function approximated by the $C_j$s in each cell.

### 4.2 Adverting the color function

It is perhaps somewhat counterintuitive that the seemingly simple problem of pushing a piecewise-constant function around by a prescribed velocity field, using the linear first order advection equation, should be one of the hard problems in computational science. The difficulty is even more surprising given the deceptive simplicity of the one-dimensional problem. Yet, the one-dimensional problem can serve as both an introduction to the difficulties as well as a motivation for the various techniques that have been designed to overcome the difficulties. We shall therefore examine the one-dimensional problem in some detail here. To simplify our task, we assume that the flow...
is incompressible so that the velocity is constant, \( u = u_o > 0 \). The advection of the color function is therefore governed by
\[
\frac{\partial C}{\partial t} + \frac{\partial F}{\partial x} = 0,
\]
where \( F = u_oC \) is the flux function. As the interface moves to the right, \( C \) flows into cell \( j \) through the left boundary and out through the right boundary. For cell \( j \) we denote the left and the right boundaries by \( j - 1/2 \) and \( j + 1/2 \), respectively, and the fluxes by \( F_{j-1/2} \) and \( F_{j+1/2} \). If the value of \( C \) in cell \( j \) at time level \( n \) is denoted by \( C^n_j \) and the value at the end of a time step \( \Delta t \) by \( C^{n+1}_j \), then we can write:
\[
C^{n+1}_j = C^n_j - \frac{1}{\Delta x} \int_{t}^{t+\Delta t} (F_{j+1/2} - F_{j-1/2}) \, dt. \tag{4.6}
\]
Or, in words, the change in \( C_j \) is the difference between what flows in and what flows out of cell \( j \). Here, \( F_{j-1/2} = u_oC_{j-1/2} \) and \( F_{j+1/2} = u_oC_{j+1/2} \) and the key challenge is to accurately estimate \( C_{j-1/2} \) and \( C_{j+1/2} \).

If we take \( C \) in each cell to be a constant, then the integration of the fluxes over time is simple. The value of \( C \) that crosses the \( j - 1/2 \) boundary is simply the value in the cell to the left, \( C_{j-1} \), and the value of \( C \) crossing the \( j + 1/2 \) boundary is \( C_j \). Thus, as long as we limit the size of each time step to \( \Delta t u_o \leq \Delta x \), we can update \( C_j \) by
\[
C^{n+1}_j = C^n_j - \frac{u_o \Delta t}{\Delta x} (C_j - C_{j-1}). \tag{4.7}
\]
In Fig. 4.2 we show the advection of \( C \) using this scheme. The initial condition is a blob of constant \( C \) that should move to the right by a constant velocity without changing its shape. The exact solution is shown at time \( t = 0 \), \( t = 140\Delta t \), and \( t = 280\Delta t \), along with the solution advected using the scheme above (equation 4.7 using 200 equal-sized control volumes for a domain of length 1, blob width of \( 20\Delta x \), and \( \Delta t = 0.5\Delta x \)). Obviously, scheme (4.7) does a terrible job. Even though the blob has only moved twice its length it has diffused excessively and as it moves further it continues to diffuse. What went wrong? We used the exact fluxes, so the problem cannot be there. The culprit is the assumption that the correct way to compute the flux of \( C \) is to represent the distribution of \( C \) in each cell by the cell average \( C_j \). Since \( C \) is supposed to approximate \( H \) we do not want any \( C \) to flow out of cell \( j \) until it is full. Thus, if cell \( j \) is initially half full, so the interface is in middle, and we take a time step \( \Delta t = u_o \Delta x / 4 \) so that the interface moves to the left boundary, then cell \( j \) should be full and cell \( j + 1 \) should still be empty at the end of the step. The problem is therefore
4.2 Advecting the color function

Fig. 4.2. Evolution of a slug of material using the first order upwind method. The
domain is resolved by 200 equal sized control volumes for a domain of length 1,
blob width of $20\Delta x$, and $\Delta t = 0.5\Delta x$. The solution is plotted at $t = 0$, $t = 140\Delta t$, and $t = 280\Delta t$.

that we computed the fluxes by assuming that $C$ is equal to its average,
everywhere in each cell.

The obvious solution to this smearing is to represent the distribution of
the marker in each cell by a higher-order function that recognizes that it
is distributed over the cell in such a way that the left hand side is full
and the right hand side is empty. If the distribution in each cell is given
by a linear function with slope $s_j$, a little bit of geometry, using Fig. 4.3,
shows that during time interval $\Delta t$, the amount that flows into cell $j$ is

$$
(C^m_{j-1} + (\Delta x - u_o \Delta t) s_{j-1}/2) u_o \Delta t,
$$

and the amount that flows out is $(C^m_j + (\Delta x - u_o \Delta t) s_j/2) u_o \Delta t$. Thus, the amount of $C$ in cell $j$ at time $n + 1$ is given by

$$
C_j^{n+1} = (1 - \lambda) C_j^n + \lambda C_{j-1}^n - \frac{\Delta x}{2} (1 - \lambda) \lambda (s_j - s_{j-1})
$$

(4.8)

where $\lambda = u_o \Delta t / \Delta x$. The slope in each cell can be estimated in several
ways. Define the left and the right slopes for cell $j$ by

$$
s^+_j = \frac{C_{j+1}^n - C_j^n}{\Delta x}; \quad \text{and} \quad s^-_j = \frac{C_{j}^n - C_{j-1}^n}{\Delta x}.
$$

(4.9)
Advection a fluid interface

Fig. 4.3. The advection of a color function using a higher-order scheme. After the slopes in each cell have been constructed, using the values in adjacent cells, the fluxes can be computed exactly by geometric considerations.

The weighted average is:

\[ s_j = \frac{1 - \kappa}{2} s_j^- + \frac{1 + \kappa}{2} s_j^+, \]

(4.10)

where \( \kappa \) is an arbitrary constant. Different selections of \( \kappa \) give different schemes. \( \kappa = 1 \), for example, gives the Lax-Wendroff scheme.

The advection of a blob using equation (4.8) is plotted in Fig. 4.4 for \( \kappa = -1, \ 0, \ +1 \), at the same times and using the same resolution and time step as in Fig. 4.2. The good news are that the results are not as diffusive as for the first order upwind method (equation 4.7 and Fig. 4.2), but the bad news are that the solutions oscillates around the interface for all values of \( \kappa \). Scheme (4.8) therefore does not really capture the sharp interface any better than the upwind scheme in Fig. 4.2. Unfortunately using a higher-order function, such as a parabola, only produces more oscillations.

This problem—the fact that using a first-order scheme to advect a discontinuity results in excessive diffusion and that higher-order methods yield oscillations—has been the subject of extensive research by a large number of investigators. The remedies are all similar in nature. The solution is advected by a high-order scheme nearly everywhere, but around the discontinuity the solution is carefully modified to eliminate oscillations. This smoothing amounts to adding artificial viscosity and results in a formal reduction in the order of the scheme near the discontinuity. Artificial viscosity was originally introduced by von Neuman (see Richtmyer and Morton, 1967) to damp out oscillations around shocks for gas dynamics simulations, but the modern way of looking at artificial viscosity originated with the high-order Godunov method of van Leer (van Leer, 1979) and the Flux Corrected
Transport method (Boris and Book, 1973). While modern shock capturing methods have reached a high degree of sophistication and do, in particular, capture shocks in compressible flows very well, they are usually not entirely satisfactory for the advection of a passive marker function, particularly for a long time. Shocks in compressible flows generally “want” to remain sharp\(^\dagger\) and slight smoothing of the shock is immediately repaired. For a passive marker there is no inherent mechanism that sharpens a smoothed interface and the flow does not “know” the difference between an absolutely sharp interface and a smoother one. Thus, shocks are generally captured better than contact discontinuities in simulations of compressible flows. For the advection of a marker function, advanced monotonicity-preserving methods for shock capturing work well for short times, but for longer times different approaches are needed.

\(^\dagger\) since characteristics flow into the shock
Adverting a fluid interface

Fig. 4.5. One-dimensional advection by the Volume of Fluid method. Given the value of the color function in the interface cell \( j \), and the side where the full cell is, the location of the interface can be found and the fluxes computed exactly.

4.3 The Volume-of-Fluid (VOF) method

While high-order monotonicity-preserving methods have been a great success for computations of flows with shocks, their success in solving equation (4.5) is mixed at best—although we should acknowledge that for relatively short times, the best of these methods do just fine. For the long time evolution of a marker function we have to step back, take a fresh look at the problem of advecting the color function \( C \), and seek a different way of doing so.

In one-dimension, as depicted in Fig. 4.5, the answer is embarrassingly simple. \( C_j \) is either 0 or 1, except in an interface cell, so the value of \( C_j \) in the interface cell yields immediately the exact location of the interface. If \( C_{j-1} = 1 \), \( C_j = 1/3 \), and \( C_{j+1} = 0 \) the interface is \( \Delta x/3 \) from the left boundary. If \( C_j = 1/2 \), the interface is in the middle of the cell, and so forth. If \( u_0 \geq 0 \) the flux through the left boundary is always \( u_0 \) and we can compute exactly the flux through the right boundary, since we know the exact location of the interface. If \( u_0 \Delta t < (1 - C_j) \Delta x \), the flux is zero and if \( u_0 \Delta t > (1 - C_j) \Delta x \) the flux is first zero and then \( u_0 \) after the interface reaches the left boundary. Thus,

\[
\int_{t}^{t+\Delta t} F_{j+1/2} \, dt = \begin{cases} 
0, & \Delta t \leq (1 - C_j) \Delta x / u_0; \\
(C_j - 1) \Delta x + u_0 \Delta t, & \Delta t > (1 - C_j) \Delta x / u_0.
\end{cases}
\]

This expression was used to compute the exact motion of the blob in Figs. 4.2 and 4.4.

Thus, for one-dimensional problems the advection of the marker function is essentially trivial. Not so in higher dimensions! The first attempt to extend the advection described by equation (4.11) to higher dimension was the
4.3 The Volume-of-Fluid (VOF) method

Simple Line Interface Calculation, or SLIC, method of Noh and Woodward (1976). In this approach the marker function is advected by time splitting where we advect first in one coordinate direction and then in the other (assuming two-dimensional flow). For advection in the horizontal direction, an interface cell is divided by a vertical line into a full part and an empty part, with the decision of which side is empty and which is full depending on the volume fraction in the cells to the left and the right. Once the location of the interface has been determined, the time integration of the fluxes is done using equation (4.11). The interface is then advected in the vertical direction by dividing the cell by a horizontal line into a full and an empty part. In three dimensions this process obviously must also be repeated for the third coordinate direction. Hirt and Nichols (1981) proposed a slightly different method where the interface was still approximated by straight lines, parallel to the coordinate axis, but the same orientation was used for the advection in the different coordinate directions. To determine whether the interface should be horizontal or vertical, Hirt and Nichols found the normal to the interface, using values of $C$ in the neighboring cells, and selected the orientation of the interface depending on whether the normal was more closely aligned with the horizontal or the vertical axis. Although perhaps more appealing than the original SLIC method, tests by Rudman (1997) suggest that the Hirt and Nichols method is not significantly more accurate. In addition to distorting the interfaces, both methods generally generate considerable amount of “floatsam” and “jetsam” where pieces of the interface break away in an unphysical way.

Although the method of Hirt and Nichols perhaps did not improve significantly on the SLIC approach, it nevertheless suggested that the key to improving the behavior of the advection scheme was the reconstruction of the interface in each cell, using the values of the marker function in each cell, along with the value in the neighboring cells. In the Piecewise Linear Interface Calculation (PLIC) method introduced by DeBar (1974) and Youngs (1982), the interface is approximated by a straight line segment in each cell, but the line can be oriented arbitrarily with respect to the coordinate axis. The orientation of the line is determined by the normal to the interface, which is found by considering the average value of $C$ in both the cell under consideration as well as the adjacent cells. Once the interface in each cell has been constructed, the fluxes from one cell to another are computed by geometric considerations. The result of the advection generally depends on the accuracy of the interface reconstruction and finding the normal accurately therefore becomes critical for PLIC methods. Several methods have been proposed to do so. Given $C$ in each cell and the normals, the exact loca-
Advection a fluid interface

The original interface. (b) The original SLIC reconstruction. (c) The Hirt and Nichols reconstruction. (d) PLIC reconstruction. Figure adapted from Rudman (1997).

The position of the interface can be determined. In two dimensions the line segment can cross any of two adjacent or opposite cell faces, so there are four basic interface configurations. In three dimensions there are many more possible configurations, adding considerably to the complexity of the method.

Figure 4.6, adapted from Rudman (1997), shows the main difference between the interface reconstruction using SLIC, the Hirt and Nichols VOF method, and PLIC. While the linear reconstruction captures slowly varying sections of the interface very well, it usually does less well for interfaces that are changing rapidly. Notice that the line segments are not continuous across cell boundaries.

In the next Chapter we discuss modern VOF methods in detail, and later in the book we will show several examples of computations done using VOF methods.
When using a Front-Tracking method the fluid interface is represented by connected marker particles that are advected by the fluid velocity, interpolated from the fixed grid.

4.4 Front Tracking

Instead of advecting a marker function by reconstructing the location of the interface in a partially full cell, for one-dimensional problems we can, of course, simply use a marker point that is moved with the imposed velocity. If the interface is located at $x_f$, then the amount of $C_j$ in the interface cell is given exactly by $C_j = (x_f - x_{j-1/2})/\Delta x$. We can also construct a smoother approximations to the step function by setting the cell values as a function of the distance to the interface. In one dimension both approaches are equally straightforward. For two- and three-dimensional problems, we need to use many marker points that are connected to represent a curve (in two-dimensions) or a surface (in three-dimensions). Figure 4.7 shows schematically the representation of an interface using connected marker points, in two dimensions.

The use of connected marker points in simulations of the motion of a fluid interface in viscous fluids goes back to Daly (1969b), who used the points to compute surface tension in MAC simulations of the Rayleigh-Taylor instability. The use of marker points to track shocks was discussed by Richtmyer and Morton (1967) who appear to have introduced the term “Front Tracking,” but they neither showed nor referenced any implementations. Glimm and
collaborators (Glimm et al., 1981; Glimm and McBryan, 1985; Chern et al., 1986) developed the preliminary ideas of Richtmyer and Morton (1967) into a general methodology for flows with shocks and interfaces.

Connected marker points can be used to track the boundary between different fluids or phases in several different ways. In most case it is, however, necessary to decide how the front is represented and managed as it stretches and deforms; how the interface is advanced in time; how the interface interacts with the underlying grid used to solve the equations governing the fluid flow; and how the topology of the interface is changed when fluid blobs merge and break apart. With the exception of two-dimensional flows, where any data structure can be made to work relatively easily, essentially all Front-Tracking methods use triangulated unstructured grids to represent the interface. This allows grid points to be added and deleted as the grid stretches and compresses. Unstructured triangulated grids have been used extensively for finite element methods and for Finite-Volume computations of inviscid flows in aerospace applications. Extensive literature is therefore available for both the generation and management of such grids (see Thompson et al., 1998, for example). For multiphase flow simulations, where the conservation equations governing the fluid flow are solved on a fixed grid, the effort required to manage the front is usually small compared with the overall effort required for the simulation. Furthermore, the information carried by the grid are usually fairly simple and the demand on grid quality is not as high as when the triangulated grid is used to solve the full fluid equations. The ease of writing the front part of a simulation code does, however, depend sensitively on using the most appropriate data structure for the front.

The way in which the front interacts with the underlying fixed grid is what distinguishes between the various Front-Tracking methods. Several issues must be decided, including how information is transmitted between the front and the grid and how the update of variables next to the front is accomplished. The simplest approach is to take the front to represent a smooth transition between the different fluids. Since the the interface is accounted for by singular source terms in the governing equations, this corresponds to approximating the singular functions by smooth distributions on the fixed grid. The origin of this approach can be found in several particle-in-cell methods for fluid flow and plasma simulations (Hockney and Eastwood, 1981; Birdsall and Langdon, 1985), the vortex-in-cell algorithm of Christiansen (1973), and the Immersed-Boundary Method of Peskin (1977). Since the role of the front is exclusively the advection of the marker function, this approach is very much like a front-capturing method with a perfect
(or nearly perfect) advection scheme. While information must be passed between the front and the grid, the smoothing of the interface alleviates the need for any modification of the solution method for the fluid equations near the interface (except that the solver must be written for material properties that vary spatially). For finite-Reynolds-number multiphase flows, Unverdi and Tryggvason (1992) developed a method where the front is used both to update the marker function and to include surface tension. Their method has been used to study a wide variety of multiphase flows and we describe it in more detail in Chapter 6.

To avoid smoothing the interface, but still use a fixed grid for the solution of the conservation equations, it is necessary to modify the numerical approximations near the front. In Glimm’s Front-Tracking method, the field variables at the front are extrapolated to grid points on the other side of the front, allowing the use of regular finite-difference discretizations for grid points next to an interface. A similar approach is used in the Ghost-Fluid method of Fedkiw et al. (1999), although the interface tracking is done using a Level-Set method instead of connected marker particles. As discussed by Glimm et al. (2001), the extrapolation can be done in several different ways, but in all cases the contribution from the ghost points results in the equivalence of source terms in the discrete governing equations. This is also the case in the Immersed-Interface Method of Lee and LeVeque (2003) where the jump conditions at the front are incorporated directly into the numerical approximations for gradients evaluated near the front, on the fixed grid.

Another approach to keep the interface sharp is the modification of the fixed grid near the front in such a way that the grid lines coincide with the interface. This generally leads to irregularly-shaped control volumes near the interface and often involves merging some of the control volumes to eliminate very small ones. The origin of this approach is the “cut cell” methods initially developed for aerospace applications (see Powell, 1998, for example), but its application to multiphase flow simulations was originated by Udaykumar et al. (1997). See also Udaykumar et al. (1999), Udaykumar et al. (2001), Liu et al. (2005) and Marella et al. (2005). Since the jump conditions can be imposed directly at the cell boundary, this method is, in many ways, similar to using two separate grids for each phase. We do, however, mention it here as most of the grid is left unchanged.

One of the main objections to the use of Front-Tracking methods is that topology changes, where fluid regions merge or breakup, are not handled automatically, as in methods where the marker function is advected directly. Changing the connection of the front points can obviously be accomplished, but at the cost of increased code complexity. When two interfaces come close
together, the film between them can sometimes become very thin and rupture. However, it may or may not be appropriate to fuse interfaces together when the film is of the order of one grid spacing. In Front Tracking the interfaces never fuse together unless something special is done and sometimes the added level of control provided by the tracking is desirable. The question of how interfaces merge is still not a completely resolved issue as discussed in Chapter 2, and we will return to how to handle it computationally at the end of Chapter 6.

4.5 The Level-Set method

In the Level-Set method the different fluid regions are identified by a smooth marker function, \( F(x, t) \) which is positive in one fluid and negative in the other. The boundary between the fluids is identified by the \( F(x, t) = 0 \) level curve. The level-set function moves with the fluid and therefore evolves according to

\[
\frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0. \tag{4.12}
\]

The motion of the zero level-set curve depends only on the normal velocity component. The normal can be found by

\[
\mathbf{n} = -\frac{\nabla F}{|\nabla F|}. \tag{4.13}
\]

Substituting for \( \nabla F \) in equation (4.12) and using that \( \mathbf{u} \cdot \mathbf{n} = V_n \) results in

\[
\frac{\partial F}{\partial t} - V_n|\nabla F| = 0 \tag{4.14}
\]

which is an alternate expression for the advection of \( F \).

Unlike the \( C \) function in the VOF method, the level-set marker function is smooth and it can therefore—at least in principle—be advected using any standard method for hyperbolic equations. Most authors have, however, used high order schemes such as the ENO method of Osher and Shu (1991).

To reconstruct the material properties of the flow (density and viscosity, for example), a marker function is constructed from \( F \)

\[
I(F) = \begin{cases} 
0, & \text{if } F < -\alpha \Delta x; \\
\frac{1}{2}(1 + (F/\alpha \Delta x) + \frac{1}{2} \sin(\pi F/\alpha \Delta x)), & \text{if } |F| \leq \alpha \Delta x; \\
1, & \text{if } F > \alpha \Delta x.
\end{cases} \tag{4.15}
\]

Here \( \Delta x \) is the size of a grid cell and \( \alpha \) is an empirical coefficient, often taken to be equal to 3, giving an interface thickness of about 6 cells. Thus, \( I \) changes from 0 to 1 over only a few cells, describing a smooth transition
4.5 The Level-Set method

zone from one fluid to the next. Once $I$ has been constructed, the various material properties can be assigned. Given $I(x)$, we can generate a smoothed grid delta function at the interface by taking its gradient

$$
\delta = \nabla I = \frac{dI}{dF} \nabla F.
$$

The mapping in equation (4.15) requires the use of the values of $F$ off the $F = 0$ contour (or level set). For the thickness of the transition zone to remain approximately constant, $F$ must have the same shape near the interface for all times. If $F$ is simply advected by the fluid, this is not usually the case. Where the interface is stretched the gradient of $F$ becomes steeper and where the interface is compressed the gradient becomes smaller. To deal with this problem, Sussman et al. (1994) introduced a reinitialization procedure where they modified $F$ by solving

$$
\frac{\partial F}{\partial \tau} + \text{sgn}(F_0)(|\nabla F| - 1) = 0
$$

at each time step. Here, $\text{sgn}$ is the sign function and $F_0$ is the un-reinitialized level-set function. $\tau$ is a pseudo-time and equation (4.17) is integrated to steady state, thus enforcing $|\nabla F| = 1$. This makes $F$ a distance function and ensures that the slope near the interface is always the same. For slowly-moving interfaces the level-set function only needs to be re-initialized every once in a while, but for more rapidly moving interfaces it may have to be reinitialized at every time step. The re-initialization can result in an artificial motion of the interface which can contribute to poor mass conservation. Several improvements have therefore been proposed, including approximating the $\text{sgn}$ function using a smoother function. Peng et al. (1999) proposed replacing $\text{sgn}(F_0)$ in equation (4.17) by

$$
S(F) = \frac{F}{\sqrt{F^2 + |\nabla F|^2 \Delta x^2}}
$$

and applying (4.17) at every time step. As long as the level-set function takes a large positive or negative values far away from the interface, the reconstruction only needs to be done near the $F = 0$ contour. Since the solution to equation (4.17) propagates outward from the interface, where $F$ is given, the solution is first corrected there and it is usually sufficient take only a few steps in pseudo time. The reconstruction of the level-set function as a distance function was critical in making level sets work for fluid dynamics simulations. As Osher and Fedkiw (2001) point out, re-initialization is still an active area of research.

For fluid-dynamics problems the $F$ function is generally advected by the
fluid velocity. In other applications, such as the motion of a solidification front, the velocity is only defined at the interface and must be extended off the interface to advect the level-set function. The extension of the velocities off the interface has been the subject of several papers, including Chen et al. (1997b) who showed that the velocity field known at the interface could be extended into the region around the interface by using an upwind method to solve

\[
\frac{\partial S}{\partial \tau} + S(F) \mathbf{n} \cdot \nabla S = 0
\]  

for each velocity component (taking \( S = u, v, \) or \( w \)). This process results in a velocity field that is constant in a direction normal to the interface. As for the reinitialization of the level-set function as a distance function (equation 4.17), the velocity field only needs to be constructed in the neighborhood of the \( F = 0 \) contour.

The Level-Set method, at least in its original embodiment, has the appeal of simplicity. For a fluid mechanics problem one only has to solve one additional partial differential equation and there are essentially no additional complex steps, such as the reconstruction of an interface in the VOF method, or the addition of new computational objects such as when Front Tracking is used. This simplicity has, however, come at some cost and early implementations had considerable problems with mass conservation. While many of the early difficulties have been overcome, efforts to generate more accurate methods generally result in added complexity. This has eroded the main appeal of Level-Set methods and made them more comparable to other approaches, both in terms of complexity and performance.

For further discussions of the Level-Set method, the reader is referred to Sussman et al. (1998) who proposed a way to improve the mass conservation; Sussman and Puckett (2000), who introduced a hybrid VOF/Level-Set method, sometimes known by its acronym CLSVOF; and Fedkiw et al. (1999) who developed a “ghost fluid” method based on assigning fictitious values to grid points on the other side of a fluid discontinuity. For general reviews of the Level-Set method, see Osher and Fedkiw (2002) and Sethian (2001).

### 4.6 Phase-Field methods

In the Phase-Field method, the interface is kept relatively sharp by a modification of the governing equations. The interface is assumed to be of a finite thickness and described by thermodynamically consistent conservation laws. In actual implementations, however, the thickness of the transition is much
larger than it is in real systems and it is not clear whether keeping the thermodynamic conditions correct in an artificially thick interface has any advantages over methods that model the behavior in the transition zone in other ways. The Phase-Field approach has found widespread use in simulation of solidification, but its use for fluid dynamic simulations is more limited.

To update the phase function $c$, which identifies the different fluids, a nonlinear diffusion term is added to the advection equation, resulting in the so-called Cahn-Hilliard equation with advection. The diffusion terms smear out an interface that is becoming thinner due to straining but an antidiffusive part prevents the interface from becoming too thick, such as if the interface is compressed. The Navier-Stokes equations are also modified by adding a term that results in surface tension in the interface zone. The key to the modification is the introduction of a free-energy function and by selecting the function in the proper way—including the appropriate values for the various adjustable coefficients—it is possible to ensure that the thickness of the interface remains of the same order as the grid spacing.

For a two-fluid system where the two fluids have similar densities (so the Boussinesq approximation can be used) and same viscosity and mass diffusion coefficient, Jacqmin (1999) solved

$$
\rho_0 \frac{Du}{Dt} = -\nabla S + \mu \nabla^2 u - c \nabla \phi + g \rho(c)
$$

(4.20)

for the fluid velocity and updated the phase-field variable by

$$
\frac{Dc}{Dt} = \kappa \nabla^2 \phi.
$$

(4.21)

Here, $\kappa$ is a diffusion coefficient and the potential $\phi$ is derived from the free energy of the fluid by

$$
\phi = \beta \frac{d\psi(c)}{dc} - \alpha \nabla^2 c,
$$

(4.22)

where $\psi(c) = (c + 1/2)^2(c - 1/2)^2$. The coefficients $\alpha$ and $\beta$ describe the relative importance of the “gradient” energy and bulk energy density. It can be shown that the thickness of the phase boundary is $O(\alpha/\beta)$ and surface tension is proportional to $\sqrt{\alpha/\beta}$. $S$ is a pressure-like variable used to enforce incompressibility. For any given grid spacing $\Delta x$, we must select the appropriate diffusion coefficient $\kappa$ as well as $\alpha$ and $\beta$. For the $\psi$ used above, Jacqmin (1999) showed that the surface tension is $\sigma = \sqrt{\alpha/\beta}/18$ and that the interface thickness $\epsilon$, defined as the 90% variation of $c$, is given by $\epsilon \approx 4.164 \sqrt{\alpha/\beta}$. Thus, if surface tension and interface thickness are given
(and we take $\epsilon$ to be equal to two or three grid spacings $\Delta x$) then the coefficients are determined. Jacqmin also showed that the diffusion coefficient for the phase-field variable $c$ must be selected such that $\kappa = O(\epsilon^\delta)$ where $\delta < 2$. To advect $c$, Jacqmin introduced a fourth-order method that allowed the interface thickness to be two or three grid spacings.

Jacqmin (1999) analyzed the method in some detail and showed examples of computations of a two-dimensional Rayleigh-Taylor instability in the Boussinesq limit. Other applications of the Phase-Field method to two fluid simulations include Jamet et al. (2001), who focused on phase change problems, Verschueren et al. (2001) who examined thermocapillary flow instabilities in a Hele-Shaw cell, and Jacqmin (2000) who studied the contact-line dynamics of a diffuse fluid interface. More recent progress can be found in Yang et al. (2006a) and Ding et al. (2007), for example.

Results for two-fluid problems produced by the phase-field method suggest that it is comparable to other methods that use a fixed grid to solve the one-field formulation of the governing equations. The smoothing of the transition zone and the use of front capturing (rather than explicit tracking) is likely to make the phase-field method similar to the level-set approach. It is, however, important to note that the methods are derived based on fundamentally different assumptions and that the Phase-Field method can, at least in principle, be used to study small-scale phenomena, such as contact line motion, for which tracking methods that start from the sharp interface hypothesis are unsuitable.

4.7 The CIP method

The CIP method is designed to reduce dispersive error in the advection of a function $f$ by fitting a cubic polynomial to the nodal values of $f$ and its derivatives. While the initials CIP have stayed constant since the introduction of the method (Takewaki et al., 1985; Takewaki and Yabe, 1987) the name of the method has evolved. CIP initially stood for Cubic Interpolated pseudo-Particle then for Cubic Interpolated Propagation (Nakamura and Yabe, 1999) and most recently for the Constrained Interpolation Profile method (Yabe et al., 2001).

The fundamental idea of the CIP method is that in addition to solving an advection equation for a conserved marker function $f$:

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} = 0,$$

we also solve an advection equation for the derivative of $f$, $g = \frac{\partial f}{\partial x}$. This
advection equation is easily constructed by differentiating equation (4.23)

\[
\frac{\partial g}{\partial t} + u \frac{\partial g}{\partial x} = 0.
\] (4.24)

Here we have assumed that \( u \) is a constant, so the right hand side of both equations is zero. If the velocity field is not constant, the advection equation for \( g \) can be split into two parts that are treated sequentially. In the first part we solve the advection part, assuming that the right-hand side is zero and in the second part we add the effect of the right-hand side. The discussion below applies to the first part. Equations (4.23) and (4.24) state that \( f \) and its derivative \( g \) are advected without change of shape. Thus, the solution at time \( t \) is simply the solution at \( t - \Delta t \).

To advect \( f \) and \( g \), we introduce a cubic polynomial, \( P(x) = ax^3 + bx^2 + cx + d \) and determine the coefficients in such a way that \( P \) matches \( f \) and \( g \) at grid points \( j \) and \( j - 1 \) (for \( u > 0 \), when \( u < 0 \) we use points \( j \) and \( j + 1 \)).
Adverting a fluid interface

Taking \( x_{j-1} = 0 \) and \( x_j = \Delta x \), we easily find that

\[
a_j = \frac{2}{\Delta x^2} (f_{j-1} - f_j) + \frac{1}{\Delta x^2} (g_{j-1} + g_j),
\]

\[
b_j = \frac{3}{\Delta x^2} (f_j - f_{j-1}) - \frac{1}{\Delta x^2} (g_j + 2g_{j-1}),
\]

\[
c_j = g_{j-1},
\]

\[
d_j = f_{j-1}.
\]

The exact solution of equations (4.23) and (4.24) can be found by simply translating the profiles, so that \( f(t, x) = f(t - \Delta t, x - u\Delta x) \), and \( g(t, x) = \partial f(t - \Delta t, x - u\Delta x)/\partial x \). Therefore, if \( \xi = u\Delta t \), the values of \( f_j^{n+1} \) and \( g_j^{n+1} \) are

\[
f_j^{n+1} = a_j \xi^3 + b_j \xi^2 + g_{j-1}^n \xi + f_j^n,
\]

\[
g_j^{n+1} = 3a_j \xi^2 + 2b_j \xi + g_{j-1}^n.
\]

In Fig. 4.8 we show the advection of a square hump by the CIP method described above. The grid used, the initial conditions, and the time step are the same as in Figs. 4.2 and 4.4. To generate the initial conditions for \( g \), we took the derivative of the initial \( f \) using centered finite differences. Although the discontinuity is preserved very well, small oscillations still appear. We note that the original CIP method was not invented specifically for discontinuous solutions and that its remarkably good performance in Fig. 4.8 is due to the very low dispersive error of the method.

For multidimensional flows it is, of course, possible to use time splitting and do the advection separately in each direction. More advanced implementations, however, introduce a multidimensional polynomial that is then shifted with the local velocity. For a discussion of the CIP method see Yabe et al. (2001).

4.8 Summary

The use of a single set of governing equations to describe the flow of two or more fluids separated by a common interface requires the accurate tracking of the interface. This is usually done by adverting a marker function which takes one value in one fluid and another value in the other fluid. Adverting the marker function in such a way that it remains sharp at the interface is one of the most challenging problems of modern computational fluid dynamics. Several methods have been developed to do this and here we have given a brief overview of some of the main alternatives. In the remainder of the book we will focus on two of those, the Volume-of-Fluid
(VOF) method and the Front-Tracking method of Unverdi and Tryggvason (1992). The VOF method is the most widely used method to directly advect the marker function. While early implementations may not have been able to capture interfaces accurately, major improvements have been made and these methods are now capable of producing solutions comparable to the best of the more recent alternatives. Although some of the alternative, such as level sets, have gained popularity because of their apparent simplicity, effort to increase their accuracy generally result in added complexity, thus reducing their appeal when compared to the VOF method. The most advanced methods to directly advect a marker function on a fixed grid are currently capable of producing solutions of a quality comparable to early implementations of front-tracking methods. However, the accuracy of front-tracking methods can also be improved and it is probably fair to say that for comparable computational effort, the use of interface markers will always yield superior results. Such methods are, however, more complex and it is likely that methods where the marker function is advected on a fixed grid will remain more popular for the foreseeable future.