# A PARALLEL LEVEL-SET APPROACH FOR TWO-PHASE FLOW PROBLEMS WITH SURFACE TENSION IN THREE SPACE DIMENSIONS

ROBERTO CROCE*, MICHAEL GRIEBEL* , AND MARC ALEXANDER SCHWEITZER*

**Abstract.** In this paper we present a parallel three-dimensional Navier–Stokes solver for incompressible two-phase flow problems with surface tension. It employs a standard finite difference discretization on uniform Cartesian staggered grids and uses Chorin's projection approach. The free surface between the two fluid phases is tracked with a level-set approach. Here, the interface conditions are implicitly incorporated into the momentum equations by the continuum surface force method. Surface tension is evaluated using a smoothed delta function and third order interpolation. The problem of mass conservation for the two phases is treated by a reinitialization of the level-set function employing a mollified signum function. All convective terms are discretized by a WENO scheme of fifth order. Altogether, our approach exhibits a second order convergence away from the free surface. For the discretization of surface tension we need to apply a smoothing scheme near the free surface which leads to a first order convergence in the smoothing region. The parallelization of the code is based on conventional domain decomposition using MPI.

We discuss the details of the method and present the results of several numerical experiments concerning mass conservation, convergence of curvature and convergence of the complete two-phase flow scheme. Furthermore, we give a comparison of our simulated results to physical data for the collapsing water column problem. Finally, we present the results of a direct numerical simulation of the dynamic contact angle behavior for a drop which moves down an inclined solid wall.

**Key words.** incompressible two-phase flow, surface tension, capillarity, continuum surface force method, level-set method

**AMS subject classifications.** 76B45, 76T10, 76M20

**1. Introduction.** Many problems in fluid flow applications require the consideration of two incompressible immiscible fluids and, consequently, the treatment of the free surface between these two phases. Examples are the formation of droplets, different coating processes (roll, dip or curtain coating) or various microfluidic applications. The main challenge for numerical methods for time-dependent free surface flow problems in three dimensions is to provide an accurate representation of the interface which separates the two different fluid phases. This involves the tracking of a discontinuity in the material properties like density and viscosity. Furthermore, surface tension on the moving interface must be taken into account to be able to deal with capillary effects and other curvature dependent phenomena. This is especially important if the length scales and the velocities of the fluids are small so that sufficiently large surface tension and curvature dependent forces result. A successful approach to deal with free surfaces, especially in the presence of topological changes, is the level-set method (LSM) due to Osher and Sethian [40]. But so far, most multi-phase flow solvers which employ the LSM only deal with two-dimensional problems. Furthermore, the original LSM does not ensure mass conservation for the two fluid phases due to the necessary reinitialization in every time step. To treat surface tension effects, the continuum surface force (CSF) method [5] is widely used.

In this paper we present a fully three-dimensional incompressible Navier–Stokes solver for two-phase flow problems including surface tension. It employs a standard

*Supported in part by the Sonderforschungsbereich 611 *Singuläre Phänomene und Skalierung in Mathematischen Modellen* sponsored by the *Deutsche Forschungsgemeinschaft*. Institut für Numerische Simulation, Universität Bonn, Wegelerstr. 6, D-53115 Bonn, Germany (croce@ins.uni-bonn.de, griebel@ins.uni-bonn.de, schweitzer@ins.uni-bonn.de).

discretization on uniform Cartesian staggered grids and uses Chorin's projection approach. The free surface between the two fluid phases is tracked with the level-set approach. Here, the interface conditions are implicitly incorporated into the momentum equations by the CSF method. Surface tension is evaluated using a smoothed delta function and third order interpolation. Furthermore, we use a sophisticated reinitialization scheme which improves mass conservation significantly. All convective terms are discretized by a weighted essentially non-oscillatory (WENO) scheme of fifth order. Altogether, our approach exhibits a second order convergence away from the free surface and a first order convergence in the smoothing region, i.e., in the support of the smoothed delta functional, near the free surface. The parallelization of the code is based on conventional domain decomposition techniques using MPI. This allows to deal with reasonably fine mesh resolutions in three dimensions.

The remainder of this paper is organized as follows: In section 2 we give a short review on numerical schemes for free surface representation and discuss their properties. The mathematical model is presented in section 3. Here, we implement the coupling of the Navier–Stokes equations for two flow phases by classical interface conditions which incorporate surface tension. To this end, we discuss the CSF method in the context of the level-set approach. The discretization of the coupled model is presented in detail in section 4. The Navier–Stokes equations as well as the transport equation for the level-set function are discretized in time using an Adams–Bashford method of second order, whereas the reinitalization of the level-set function is discretized in time using an explicit third order Runge–Kutta scheme. The discretization in space employs a standard finite difference approach on a staggered uniform grid. Furthermore, we use a signed distance function in the level-set representation of the free surface, a fifth order WENO scheme for all convective terms, a higher order interpolation scheme for the level-set function and an improved reinitialization scheme. Then we shortly review the parallelization of our solver by domain decomposition techniques in section 4.8. In section 5 we present the results of several numerical experiments concerning mass conservation, convergence of curvature and convergence of the complete two-phase flow scheme. Furthermore, we give a comparison of our simulated results to physical data for the collapsing water column problem. In our last example we show the results of a direct numerical simulation of the dynamic contact angle behavior for a drop moving down an inclined solid wall. Finally, we conclude with some remarks in section 6.

**2. Numerical Techniques for Free Surface Representation.** The earliest attempts to compute solutions of fluid-dynamics problems with free boundaries were made in the middle of the 1960s. Harlow and Welch [25] developed the popular Marker and Cell method (MAC) which uses massless marker particles to explicitly represent the flow domain and thus its free surface, see Figure 2.1. Daly [11] extended this method to two-phase flows. Note, however, that an explicit reconstruction of the interface requires some additional work in the MAC scheme. Various modifications of the original MAC scheme were developed over the years and have further improved this approach, see Daly [12], Amsdon and Harlow [2], Griebel et al. [21, 22], Neunhoeffer [37] and Tryggvason et al. [50, 51]. Experiments with only a small layer of particles near the free surface were carried out in [7].

In the 1970s the Volume of Fluid method (VOF) was developed. The VOF employs a piecewise constant scalar field $F$ instead of marker particles to locate the position of both fluid phases. Here, $F$ describes the volume fraction of one fluid for each discretization cell. One fluid phase corresponds to the function value $F = 1$, and

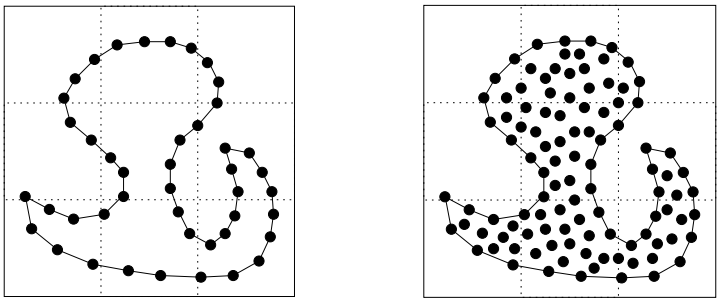

FIGURE 2.1. *MAC particle surface tracking (left) and volume tracking (right).*

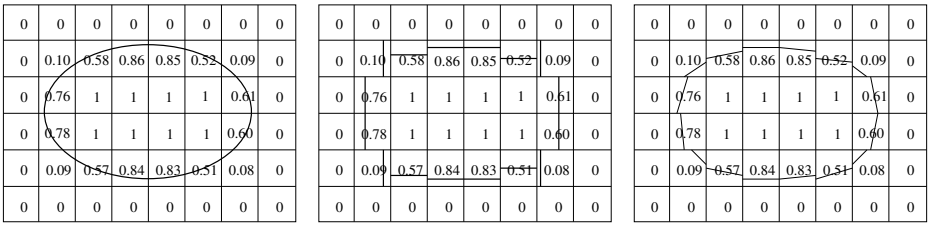


FIGURE 2.2. *Exact surface (left), SLIC (center) and PLIC (right) reconstruction.*

the second fluid phase to $F = 0$. Cells which are cut by the interface are assigned values $0 < F < 1$ according to the volume fraction of the first fluid. This function $F$ is transported over time with the flow. Note that this implicit approach also allows for a simple handling of topological changes but — unlike in MAC schemes — an explicit reconstruction of the free surface does not involve any additional consideration. Furthermore, it can easily be implemented for the three-dimensional case. The VOF was first introduced by DeBar [13], and got subsequently improved by Noh and Woodward [38], Hirt and Nichols [26], Lafaurie et al. [31], and Zaleski et al. [24].

The main difficulty in VOF schemes is to obtain a reconstruction with sufficient accuracy and smoothness, compare Figure 2.2. This is especially important when the computation of curvature is needed, i.e., for surface tension. Here, two well-known schemes are the Simple Line Interface Calculation (SLIC) [38] technique and the Piecewise Linear Interface Calculation (PLIC) [26] method. But for any meaningful curvature calculation higher order spline interpolations must be used, see Wittum et al. [19, 20].

At the end of the 1980s the level-set (LS) technique was introduced by Osher and Sethian [40]. Here, the basic idea is to describe an interface $\Gamma$ as the zero level-set of a continuous higher-dimensional scalar field $\phi(\mathbf{x}, t)$; i.e., for points $\vec{x}$ on the free surface $\Gamma$ at time $t$ we have $\phi(\vec{x}, t) = 0$. Hence, the two fluid phases can easily be identified by looking at the sign of $\phi$. Similar to the VOF method, the level-set technique greatly reduces the complexity of a description of the interface, especially when topological changes such as pinching and merging occur, compare Figure 2.3. Yet, in contrast to VOF, the approximation with the level-set function is globally continuous. Thus, the solution is less affected by numerical diffusion during the transport process and the stability of the numerical scheme is increased. First applications of level-set-methods in the area of fluid-dynamics were described by Mulder, Osher and Sethian [36] and Sussman, Smereka, and Osher [47]. Further developments are found in the books of

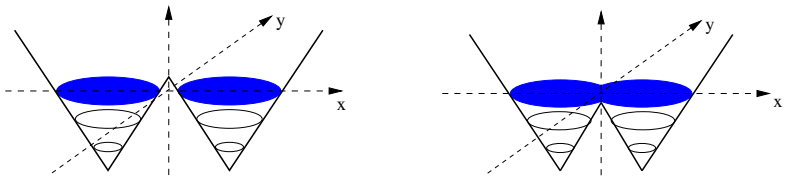


FIGURE 2.3. *Area bounded by zero level-set before (left) and after merging (right).*

Sethian [43] and Osher and Fedkiw [39].

For the treatment of surface tension on the free surface Brackbill et. al [5] and Unverdi et. al [51] independently developed the Continuum Surface Force (CSF) discretization technique. Here, the interfacial tension condition is transformed into a body force for cells which contain the interface. A variant is the continuous surface stress scheme [31]. These approaches were first presented in the context of VOF and MAC methods but are nowadays also used in level-set methods, see Sussman et. al [47]. They have proven to be quite robust. However, to our knowledge, the convergence behavior of surface tension terms approximated by the level-set method within a full two-phase flow solver is not completely understood. Other approaches [3, 15, 16, 27, 28] to the numerical treatment of surface tension employ a variational approach using the Laplace–Beltrami operator on the free surface. Via integration by parts, one derivative can be shifted to the test space and a discretization with linear finite elements then leads to a first order convergence for curvature in the appropriate energy-norm.

Another important issue – especially for level-set methods – is mass conservation. In many applications the level-set function may become distorted; i.e., its gradient may become very large or very small near the interface. To avoid these situations a so-called reinitialization step must be performed. Here, the current level-set function is replaced by a smoother, less distorted function which has the same zero level-set. Reinitialization with a signed distance function has been used in various applications, see for example Chopp et al. [9] (minimal surfaces), Sussman et al. [47] (free boundary problems in two-phase flow), Chen et al. [8] (crystal growth), Merriman et al. [34] (motion of multiple junctions), and Adelsteinsson and Sethian [1] (fast marching method). However, these simple reinitialization techniques introduce numerical diffusion to the solution which leads to difficulties with volume conservation. To this end, several modified reinitalization methods have been developed to improve mass conservation, see Sussman and Fatemi [45], Sussman and Puckett [46], Russo and Smereka [42], Enright et al. [17] and Peng et al. [41]. But again, the relation between these techniques and the convergence behavior of surface tension terms within a full two-phase flow solver is not completely understood.

**3. Model.** We now discuss the mathematical model for three-dimensional flow of two immiscible incompressible fluids. The behavior of the fluids is governed by the incompressible Navier–Stokes equations defined on an open set $\Omega = \Omega_1 \cup \Omega_2 \cup \Gamma_f \subset \mathbb{R}^3$ with Lipschitz boundary $\Gamma := \partial\Omega$. Here $\Omega_1$ and $\Omega_2$ denote the subdomains of fluid one and fluid two, respectively. The free interface between the two fluid phases is denoted by $\Gamma_f := \partial\Omega_1 \cap \partial\Omega_2$. The two fluid domains $\Omega_1$ and $\Omega_2$, and the free interface $\Gamma_f$ depend on time. The temporal evolution of each of the fluids is described by the

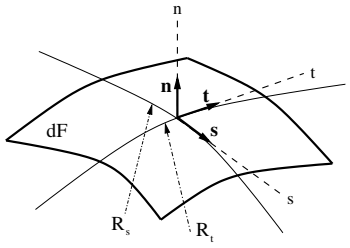


FIGURE 3.1. *Curvature for surface tension computation.*

Navier–Stokes equations in its respective subdomain $\Omega_i$ for $t \in [0, T]$; i.e., we have

$$\begin{array}{rcll} \rho_i \dfrac{D\vec{u}_i}{Dt} & = & -\nabla p_i + \nabla \cdot (\mu_i \mathbf{S}_i) + \rho_i \vec{g} & \text{in } \Omega_i, \\ \nabla \cdot \vec{u}_i & = & 0 & \text{in } \Omega_i, \\ \vec{u}_i|_\Gamma & = & 0 & \text{in [0,T] ,} \\ \vec{u}_i|_{t=0} & = & u_{0_i} & \text{in } \Omega_i, \end{array} \tag{3.1}$$

where $i \in \{1, 2\}$ indicates the considered fluid phase. Furthermore, $\vec{u}_i$ denotes the velocity-field, $p_i$ the pressure, $\mu_i$ the dynamic viscosity, $\rho_i$ the density, and $\vec{g}$ the volume force. The viscous stress tensor is given by $\mathbf{S}_i := \nabla \vec{u}_i + \{\nabla \vec{u}_i\}^T$ and $\frac{D(\vec{u}_i)}{Dt} := \partial_t(\vec{u}_i) + (\vec{u}_i \cdot \nabla)\vec{u}_i$ is the material derivative. The values of $\mu_i$ and $\rho_i$ in each phase are assumed to be constant. However, the system (3.1) is *not* complete without some additional conditions imposed on the free interface $\Gamma_f$. The basic assumption in continuum mechanics for the free boundary $\Gamma_f$ is that it can be regarded as a sharp interface; i.e., the interface is a two-dimensional surface which separates the two flow regions. For immiscible fluids this can be justified by dimension analysis. Hence, our two-phase flow model contains a jump in density and a jump in viscosity across the free surface.

A consequence of this assumption is that the interface possesses no mass. Consequently, the net stress vanishes along the interface. Furthermore, the velocity must be continuous across the free surface, i.e., $\vec{u}_1 = \vec{u}_2$ on $\Gamma_f$. For a more detailed discussion see [32]. The surface tension boundary conditions at the interface $\Gamma_f$ between the two fluid phases are given by

$$(\mathbf{T}_1 - \mathbf{T}_2) \cdot \vec{n} = \sigma\kappa\vec{n}, \qquad (\mathbf{T}_1 - \mathbf{T}_2) \cdot \vec{t} = \frac{\partial \sigma}{\partial \vec{t}}, \qquad \text{and} \qquad (\mathbf{T}_1 - \mathbf{T}_2) \cdot \vec{s} = \frac{\partial \sigma}{\partial \vec{s}},$$

where $\mathbf{T}_i := -p\mathbf{I} + \mu_i \mathbf{S}_i$ denotes the stress tensor, $\sigma$ is the surface tension coefficient determined by the physical properties of the considered fluids, and $\vec{n} := \vec{n}_1 = -\vec{n}_2$ denotes the surface normal on $\Gamma_f$, i.e., the outer normal on $\partial\Omega_1$. The local curvature $\kappa$ is given by

$$\kappa = \frac{1}{R_t} + \frac{1}{R_s} \tag{3.2}$$

with the radii $R_t$ and $R_s$ of curvature along coordinates $t$ and $s$ as shown in Figure 3.1. Since we assume that the material properties like $\mu_i$ and $\rho_i$ are constant in each $\Omega_i$, the surface tension coefficient $\sigma$ is constant throughout the *whole* domain $\Omega$.

Therefore, the conditions on the interface $\Gamma_f$ reduce to

$$\vec{u}_1 = \vec{u}_2, \tag{3.3}$$
$$(\mathbf{T}_1 - \mathbf{T}_2) \cdot \vec{n} = \sigma\kappa\vec{n}, \tag{3.4}$$
$$(\mathbf{T}_1 - \mathbf{T}_2) \cdot \vec{t} = 0, \tag{3.5}$$
$$(\mathbf{T}_1 - \mathbf{T}_2) \cdot \vec{s} = 0. \tag{3.6}$$

Throughout this paper, we only consider surface tension forces in normal direction. Hence, we complete the Navier–Stokes equations (3.1) in each of the subdomains $\Omega_i$ with the interface conditions (3.3) and (3.4) only, the tangential conditions (3.5) and (3.6) are not explicitly enforced. Altogether, the complete model for two-phase flow problems including surface tension considered in this paper is given by

$$\begin{aligned}
\rho_i \frac{D\vec{u}_i}{Dt} &= -\nabla p_i + \nabla \cdot (\mu_i \mathbf{S}_i) + \rho_i \vec{g} && \text{in } \Omega_i, \\
\nabla \cdot \vec{u}_i &= 0 && \text{in } \Omega_i, \\
\vec{u}_i|_\Gamma &= 0 && \text{in [0,T] }, \\
\vec{u}_i|_{t=0} &= u_{0_i} && \text{in } \Omega_i \\
\vec{u}_1 &= \vec{u}_2 && \text{on } \Gamma_f, \\
(\mathbf{T}_1 - \mathbf{T}_2) \cdot \vec{n} &= \sigma\kappa\vec{n} && \text{on } \Gamma_f,
\end{aligned} \tag{3.7}$$

for $i \in \{1, 2\}$.

**3.1. Formulation of surface tension within the momentum equations.** To couple the two-phase-flow Navier–Stokes equations (3.1) with the free surface boundary conditions (3.4), we start from the integral form of (3.1)

$$\int_{\Omega_i} \rho_i \frac{D(\vec{u}_i)}{Dt}\, d\vec{x} = \int_{\partial\Omega_i} \mathbf{T}_i \cdot \vec{n}\, dF + \int_{\Omega_i} \rho_i \vec{g}\, d\vec{x} \tag{3.8}$$

for $i \in \{1, 2\}$. Summation of both momentum equations (3.8) yields

$$\begin{aligned}
&\rho_1 \int_{\Omega_1} \frac{D\vec{u}}{Dt} d\vec{x} + \rho_2 \int_{\Omega_2} \frac{D\vec{u}}{Dt}\, d\vec{x} = \\
&\int_{\partial\Omega} \mathbf{T} \cdot \vec{n}\, dF - \int_{\Gamma_f} [\mathbf{T}] \cdot \vec{n}\, dF + \rho_1 \int_{\Omega_1} \vec{g}\, d\vec{x} + \rho_2 \int_{\Omega_2} \vec{g}\, d\vec{x}
\end{aligned} \tag{3.9}$$

where $\mathbf{T} := \mathbf{T}_1\chi_{\Omega_1} + \mathbf{T}_2\chi_{\Omega_2}$ and $[\mathbf{T}]$ denotes the jump in the stress tensor $T$, i.e., $[\mathbf{T}] = \mathbf{T}_1 - \mathbf{T}_2$, at the interface $\Gamma_f$. With (3.4) and the Gauß Theorem, we obtain

$$\int_{\partial\Omega} \mathbf{T} \cdot \vec{n}\, dF = \int_\Omega \nabla \cdot \mathbf{T}\, d\vec{x}. \tag{3.10}$$

Since $\mathbf{T}$ is discontinuous at the free surface due to the jump in the viscosity, relation (3.10) is to be understood in the sense of distributions.

With (3.3), the velocity-field $\vec{u} = \vec{u}_1\chi_{\Omega_1} + \vec{u}_2\chi_{\Omega_2}$ for viscous fluids is continuous on $\Omega$ and we can write

$$\rho_1 \int_{\Omega_1} \frac{D\vec{u}}{Dt}\, d\vec{x} + \rho_2 \int_{\Omega_2} \frac{D\vec{u}}{Dt}\, d\vec{x} = \int_\Omega \rho \frac{D\vec{u}}{Dt}\, d\vec{x}. \tag{3.11}$$

Now we substitute (3.10) and (3.11) into (3.9) and obtain the integral formulation of incompressible two-phase flow with surface tension

$$\int_\Omega \rho \frac{D\vec{u}}{Dt}\, d\vec{x} = \int_\Omega \nabla \cdot \mathbf{T}\, d\vec{x} - \int_{\Gamma_f} \sigma\kappa\vec{n}\, dF + \int_\Omega \rho\vec{g}\, d\vec{x}. \tag{3.12}$$

The advantage of (3.12) is that the boundary condition for surface tension is implicitly contained in the momentum equation. However, it is formulated as a free boundary integral. For the further discretization, we will convert this free boundary integral to a volume integral via the CSF approach. Then, we can easily couple the momentum equation with a level-set formulation which does not require an explicit reconstruction of the free surface. Note that this is in contrast to VOF methods where a reconstruction of $\Gamma_f$ is required even with a CSF formulation. Note further that since there is no need for an explicit reconstruction, the computation of surface tension can easily be parallelized.

**3.2. The CSF-approach for surface tension.** The CSF model was first proposed by Brackbill et al. [5] for VOF methods and by Unverdi et al. [51] for MAC methods. We now give a short description of the CSF approach in the framework of the level-set method.

The aim is to provide a coupled formulation of (3.7), i.e., we will couple the Navier–Stokes equations (3.1), the boundary conditions for surface tension (3.4) and the continuity condition (3.3). We start from the two-phase flow equations in integral form (3.12). There, surface tension is included in the right-hand side as a source-term. Since our discretization will be based on the differential expression of the Navier–Stokes equations, this free boundary integral needs to be converted into a volume integral and, by passing to the limit of infinitesimally small volumes, we then obtain the associated differential expression of the Navier–Stokes equations.

To this end, we construct a function $\phi$ such that the interface between two different fluids is the zero level-set of $\phi$. The interface is then given by

$$\Gamma_f(t) = \{\vec{x} : \phi(\vec{x}, t) = 0\}$$

for all times $t \in [0, T]$. Note that there are arbitrarily many ways to define the level-set function $\phi$ away from the free surface. Here, we choose $\phi$ as a signed-distance function such that

$$\phi(\vec{x}, t) \begin{cases} < 0 & \text{if } \vec{x} \in \Omega_1 \\ = 0 & \text{if } \vec{x} \in \Gamma_f \\ > 0 & \text{if } \vec{x} \in \Omega_2 \end{cases} \tag{3.13}$$

hold and the Eikonal equation $|\nabla \phi| = 1$ is fulfilled. With the help of this level-set function $\phi$ we can now easily define the density $\rho$ and the viscosity $\mu$ on the whole domain, i.e., of both fluid-phases. To this end, we set

$$\rho(\phi) := \rho_2 + (\rho_1 - \rho_2)H(\phi) \quad \text{and} \quad \mu(\phi) := \mu_2 + (\mu_1 - \mu_2)H(\phi)$$

where $H(\phi)$ denotes the Heavyside function which is defined as

$$H(\phi) := \begin{cases} 0 & \text{if } \phi < 0 \\ \frac{1}{2} & \text{if } \phi = 0 \\ 1 & \text{if } \phi > 0. \end{cases}$$

Now, with these $\phi$-dependent formulations, equation (3.12) can be expressed as

$$\int_\Omega \rho(\phi) \frac{D\vec{u}}{Dt} \, d\vec{x} = \int_\Omega \nabla \cdot \mathbf{T} \, d\vec{x} - \int_{\Gamma_f} \sigma \kappa \vec{n} \, dF + \int_\Omega \rho(\phi) \vec{g} \, d\vec{x}. \tag{3.14}$$

Chang et al. [6] showed that in the context of level-set methods the free boundary integral can be converted into a volume integral using a Dirac $\delta$-functional. Since $\vec{n} = \frac{\nabla\phi}{|\nabla\phi|}$ and $|\nabla\phi| = 1$, this results in the identity

$$\int_{\Gamma_f} \sigma\kappa\vec{n}\ dF = \int_{\Omega} \sigma\kappa(\phi(\vec{x}))\delta(\phi(\vec{x}))\nabla\phi(\vec{x})\ d\vec{x} \tag{3.15}$$

where $\delta$ denotes the one-dimensional Dirac $\delta$-functional, i.e. $\int_{\mathbb{R}} f(x)\delta(x)\ dx = f(0)$ for $f : \mathbb{R} \to \mathbb{R}$, such that $\int_{\mathbb{R}^3} \delta(\phi(\vec{x}))\ d\vec{x} = \int_{\Gamma_f} 1\ ds$. We substitute (3.15) into (3.14) and obtain

$$\int_{\Omega} \left( \rho(\phi)\frac{D\vec{u}}{Dt} - \nabla\cdot\mathbf{T} + \sigma\kappa(\phi)\delta(\phi)\nabla\phi - \rho(\phi)\vec{g} \right)\ d\vec{x} = 0$$

which now only involves a volume integral. Since this relation holds for any arbitrary volume $\Omega$, we can pass to the limit of infinitesimally small volumes and obtain the associated differential equation

$$\rho(\phi)\frac{D\vec{u}}{Dt} - \nabla\cdot\mathbf{T} + \sigma\kappa(\phi)\delta(\phi)\nabla\phi - \rho(\phi)\vec{g} = 0.$$

With the definition of the stress tensor $\mathbf{T}$, this yields the equation

$$\rho(\phi)\frac{D\vec{u}}{Dt} + \nabla p = \nabla\cdot(\mu(\phi)\mathbf{S}) - \sigma\kappa(\phi)\delta(\phi)\nabla\phi + \rho(\phi)\vec{g}. \tag{3.16}$$

Furthermore, we need to take into account the time-dependence of the free-surface $\Gamma_f$. In essence, the free-surface is advected with the flow-field. Hence, we can model the time-evolution of $\Gamma_f$ via a simple transport of the level-set function $\phi$ due to the underlying fluid velocity-field $\vec{u}$, i.e., via the pure transport equation

$$\phi_t + \vec{u}\cdot\nabla\phi = 0 \tag{3.17}$$

with initial value $\phi_0(\vec{x}) = \phi(\vec{x}, 0)$.

**4. Numerical method.** In this section we present our overall numerical scheme for the treatment of the two-phase flow equations with surface tension in three space dimensions as given in (3.16) and (3.17). The scheme is based on the well-known projection method employs an explicit time-stepping scheme and a finite difference discretization in space on a staggered grid which is also used for the level-set function. We begin with a short review of the classical projection method before we cover the details of the incorporation of the level-set method into the projection scheme. Here, we need to be concerned with three major issues: the movement of the interface and the conservation properties of the overall scheme, the discontinuity of the material properties at the interface, and the accurate approximation of the surface tension.

The movement of the free interface is modeled via the additional transport equation (3.17) for the level-set function $\phi$. Here, we have to deal with the problem of volume and mass conservation of the two different fluid phases. To this end, we implement an appropriate reinitialization scheme where we enforce the signed-distance property of $\phi$. Moreover, we employ a higher order WENO scheme for the treatment of the convective terms in the transport equation (3.17), the Navier–Stokes equations with surface tension (3.16) and in the reinitalization phase to improve the overall conservation properties of the numerical scheme. Furthermore, the discontinuity of the

density and viscosity at the interface may lead to spurious oscillations and instabilities, therefore we need to be concerned with the development of an appropriate smoothing scheme for the material properties. Finally, it is essential to employ a higher order interpolation scheme for the level-set function to obtain an accurate approximation of the surface tension. In the following we cover these issues in detail.

**4.1. Projection Method.** A widely used and very successful scheme for the solution of the incompressible Navier–Stokes equations is the so-called projection method due to Chorin [10]. Here, the starting point is the consideration of the time-discrete Navier–Stokes equations using a forward Euler scheme; i.e., we will compute the solution $\vec{u}^{n+1}$ at time $t_{n+1}$ from the solution $\vec{u}^n$ of the previous time step. To obtain the solution of (3.16) in a single Euler-step, we employ a two step approach. First, we compute an intermediate velocity field $\vec{u}^*$ which may not be divergence free via an explicit transport. Then, in a second step, we compute a correction $\nabla p^{n+1}$ of the intermediate velocity field via the pressure Poisson equation which leads to a divergence free velocity field $\vec{u}^{n+1}$; i.e., we treat the pressure implicitly.

The intermediate velocity field $\vec{u}^*$ is given by

$$\frac{\vec{u}^* - \vec{u}^n}{\delta t} = -(\vec{u}^n \cdot \nabla)\vec{u}^n + \vec{g} + \frac{1}{\rho(\phi^n)}\Big(\nabla \cdot (\mu(\phi^n)\mathbf{S}^n) - \sigma\kappa(\phi^n)\delta(\phi^n)\nabla\phi^n\Big) \tag{4.1}$$

where the superscript $n$ indicates the respective time step. With this definition, we can rewrite the time-discrete Navier–Stokes equations as

$$\frac{\vec{u}^{n+1} - \vec{u}^*}{\delta t} + \frac{\nabla p^{n+1}}{\rho(\phi^{n+1})} = 0, \tag{4.2}$$

$$\nabla \cdot \vec{u}^{n+1} = 0. \tag{4.3}$$

Now, we apply the divergence operator to equation (4.2), multiply by $-1$, and with (4.3), we arrive at the pressure Poisson equation with the density field as diffusion coefficient

$$-\nabla \cdot \Big(\frac{1}{\rho(\phi^{n+1})}\nabla \hat{p}^{n+1}\Big) = -\nabla \cdot \vec{u}^* \tag{4.4}$$

with $\hat{p}^{n+1} := \delta t p^{n+1}$. We obtain appropriate boundary conditions for the pressure Poisson equation by projecting the vector equation (4.2) onto the outer unit normal of the domain boundary

$$\left.\frac{\partial p^{n+1}}{\partial \vec{n}}\right|_\Gamma = \frac{\rho}{\delta t}(\vec{u}^*_\Gamma - \vec{u}^{n+1}_\Gamma) \cdot \vec{n}.$$

Thus, if we require $\vec{u}^{n+1}_\Gamma = \vec{u}^*_\Gamma$, we obtain homogeneous Neumann boundary conditions for the pressure

$$\left.\frac{\partial \hat{p}^{n+1}}{\partial \vec{n}}\right|_\Gamma = 0.$$

To ensure the existence of a solution, we also need to fulfill the compatibility condition

$$0 = \int_\Omega \nabla \cdot \vec{u}\ d\vec{x} = \int_\Gamma \vec{u} \cdot \vec{n}\ ds$$

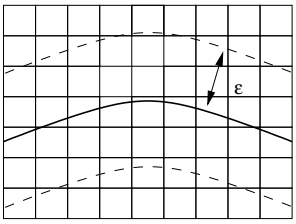


FIGURE 4.1. *Numerical smoothing region.*

which expresses the fact that the velocity on the boundary $\Gamma$ must have a vanishing total flux. Note that the employed space discretization must comply with this requirement. Now, we can solve the system (4.4) for $\hat{p}^{n+1} = \delta t p^{n+1}$, at least up to a constant. We fix this remaining degree of freedom in the pressure $p^{n+1}$ via the additional condition

$$\int_\Omega p^{n+1}\, d\vec{x} = 0$$

to ensure the comparability of the pressure results at different times $t_n$. Finally, we obtain a correction of the intermediate velocity field $\vec{u}^*$ such that the velocities at time $t_{n+1}$

$$\vec{u}^{n+1} = \vec{u}^* - \frac{1}{\rho(\phi^{n+1})}\nabla\hat{p}^{n+1} \tag{4.5}$$

are now divergence free.

The extension of Chorin's projection method to higher order multi-step time-discretizations is straightforward. Throughout this paper we use a second order Adams–Bashford scheme for the integration of the momentum equation (4.1), i.e.

$$\frac{\vec{u}^* - \vec{u}^n}{\delta t_n} = \frac{1}{2}\Big(\frac{\delta t_n + 2\delta t_{n-1}}{\delta t_{n-1}} L^n - \frac{\delta t_n}{\delta t_{n-1}} L^{n-1}\Big) \tag{4.6}$$

where $\delta t_n = t_n - t_{n-1}$ denotes the length of the $n$th time step and $L^n$ the right-hand side of (4.1) evaluated at time $t_n$.

**4.2. Smoothing.** So far we have neglected the discontinuity of the density at the interface. Yet, a jump in the diffusion coefficient in (4.4) can have a substantial adverse effect on the stability and accuracy of our numerical scheme. Hence, we need to develop an appropriate smoothing scheme for the material properties, so that the density and the viscosity are at least continuous across the interface. This smoothing scheme, however, should not affect the approximation properties globally. Another issue here is of course the discretization of the Dirac delta functional.

To this end, let us consider the interface as having a fixed thickness $\epsilon$ which is proportional to the spatial meshsize $h$, see Figure 4.1. Then, we replace the density $\rho(\phi)$ and the viscosity $\mu(\phi)$ by

$$\rho^\epsilon(\phi) = \rho_2 + (\rho_1 - \rho_2)H^\epsilon(\phi) \quad \text{and} \quad \mu^\epsilon(\phi) = \mu_2 + (\mu_1 - \mu_2)H^\epsilon(\phi) \tag{4.7}$$

where $H^\epsilon$ denotes a smoothed Heavyside function

$$H^\epsilon(\phi) := \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \frac{1}{2}(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin(\frac{\pi\phi}{\epsilon})) & \text{if } |\phi| \le \epsilon \\ 1 & \text{if } \phi > \epsilon. \end{cases} \tag{4.8}$$

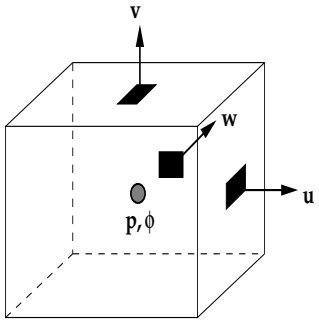


FIGURE 4.2. *Location of variables in a staggered grid cell.*

The associated smoothed delta functional is given by

$$\delta^\epsilon(\phi) := \partial_\phi H^\epsilon = \begin{cases} \frac{1}{2\epsilon}(1+\cos(\frac{\pi\phi}{\epsilon})) & \text{for } |\phi| < \epsilon \\ 0 & \text{elsewhere.} \end{cases} \tag{4.9}$$

Let us now consider a level-set function $\phi$ in this context. The smoothing employed in (4.8) and (4.9) requires that the values of $\phi$ are symmetric to the zero level-set. Hence, a level-set function with a constant gradient in the $\epsilon$-neighborhood of the interface should be employed. Due to our choice of a signed-distance function (3.13) with $|\nabla\phi| = 1$, the smoothing scheme given in (4.8) and (4.9) is directly applicable.

The use of the smoothed delta functional (4.9) reduces the quality of the approximation locally within the smoothing region by one order. Consider a function $g$ and its approximation $g_h$ with a pointwise approximation error $g - g_h$ of order $O(h^p)$. Then, the smoothed error $(g - g_h)\delta^\epsilon$ is of the order $O(h^p)O(\epsilon^{-1}) \approx O(h^{p-1})$ since $\epsilon \approx h$.

**4.3. Momentum Equation.** Let us now consider the numerical treatment of the momentum equation. To this end, we need to specify the space discretization scheme. Here, we employ a widely used finite difference scheme on a staggered grid to discretize the velocity field $\vec{u} := (u, v, w)^T$ and the pressure $p$; i.e., the cell centers correspond to the pressure nodes $p_{i,j,k}$ whereas the cell-face centers give the velocity nodes $u_{i+\frac{1}{2},j,k}$, $v_{i,j+\frac{1}{2},k}$ and $w_{i,j,k+\frac{1}{2}}$, respectively. Furthermore, we discretize the level-set function on the same grid, i.e., in cell centers, see Figure 4.2.

**4.3.1. Discretization of viscous terms.** The computation of the intermediate velocity $\vec{u}^*$ requires the discretization of the viscous terms $\nabla\cdot(\mu(\phi^n)\mathbf{S}^n)$ in the velocity nodes. Hence, the $x$-component of the viscous term

$$2(\mu(\phi)u_x)_x + (\mu(\phi)(u_y + v_x))_y + (\mu(\phi)(u_z + w_x))_z \tag{4.10}$$

is discretized in the nodes $(i \pm \frac{1}{2}, j, k)$, whereas the $y$-component

$$(\mu(\phi)(u_y + v_x))_x + 2(\mu(\phi)u_y)_y + (\mu(\phi)(v_z + w_y))_z \tag{4.11}$$

is discretized at $(i, j \pm \frac{1}{2}, k)$ and the $z$-component

$$(\mu(\phi)(u_z + w_x))_x + (\mu(\phi)(v_z + w_y))_y + 2(\mu(\phi)w_z)_z \tag{4.12}$$

at $(i, j, k \pm \frac{1}{2})$. Note that for the ease of notation we omit the superscript $n$.

Since the velocities are continuous for viscous fluids, the first derivatives can be computed with central differences. For example, the first derivatives of the velocity $u$ in the $x$-direction are approximated by

$$\begin{aligned}
[u_x]_{i,j,k} &= (\delta x)^{-1}(u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}),\\
[u_y]_{i+\frac{1}{2},j+\frac{1}{2},k} &= (\delta y)^{-1}(u_{i+\frac{1}{2},j+1,k} - u_{i+\frac{1}{2},j,k}),\\
[u_z]_{i+\frac{1}{2},j,k+\frac{1}{2}} &= (\delta z)^{-1}(u_{i+\frac{1}{2},j,k+1} - u_{i+\frac{1}{2},j,k}).
\end{aligned}$$

The first derivatives of the other velocities $v$ and $w$ are discretized in a similar fashion. Furthermore, the second order derivatives in (4.10), (4.11) and (4.12) can also be approximated using central differences, since the jump in the viscosity across the interface has been regularized by the smoothed Heavyside function $H^\epsilon$, see (4.7).

In summary, the terms in (4.10) at position $(i+\frac{1}{2},j,k)$ are approximated by

$$\begin{aligned}
[(2\mu(\phi)u_x)_x]_{i+\frac{1}{2},j,k} &= 2(\delta x)^{-1}\Big(\mu(\phi_{i,j,k})\,[u_x]_{i,j,k} - \mu(\phi_{i+1,j,k})\,[u_x]_{i+1,j,k}\Big)\\
[(\mu(\phi)(u_y+v_x))_y]_{i+\frac{1}{2},j,k} &= (\delta y)^{-1}\Big(\mu(\phi_{i+\frac{1}{2},j+\frac{1}{2},k})\,[u_y+v_x]_{i+\frac{1}{2},j+\frac{1}{2},k}\Big) -\\
&\quad (\delta y)^{-1}\Big(\mu(\phi_{i+\frac{1}{2},j-\frac{1}{2},k})[u_y+v_x]_{i+\frac{1}{2},j-\frac{1}{2},k}\Big),\\
[(\mu(\phi)(u_z+w_x))_z]_{i+\frac{1}{2},j,k} &= (\delta z)^{-1}\Big(\mu(\phi_{i+\frac{1}{2},j,k+\frac{1}{2}})\,[u_z+w_x]_{i+\frac{1}{2},j,k+\frac{1}{2}}\Big) -\\
&\quad (\delta z)^{-1}\Big(\mu(\phi_{i+\frac{1}{2},j,k-\frac{1}{2}})[u_z+w_x]_{i+\frac{1}{2},j,k-\frac{1}{2}}\Big).
\end{aligned}$$

Since the level-set function $\phi$ is discretized in the cell-centers $(i,j,k)$, the values in the cell-face centers are not directly available. Therefore, we need to employ an appropriate interpolation scheme to compute the required values e.g. $\phi_{i+\frac{1}{2},j,k+\frac{1}{2}}$ in the cell-face centers. To this end, we use a higher order Lagrange interpolation scheme, see (4.13).

**4.3.2. Discretization of convective terms.** To be able to treat convection dominated flow problems, it is essential to employ a higher order discretization for the convective terms. Here, we use a fifth order WENO scheme [29, 30] for the treatment of all convective terms. Note that the use of a higher order scheme also improves the conservation properties of the overall numerical method.

The main advantage of a WENO scheme is that it can handle large gradients and even shocks very accurately since the approximation takes local smoothness into account. The weighted combination of multiple stencils then gives a higher order approximation. The weighting scheme used in a WENO method is based on the smoothness of the local stencils such that the largest weight is assigned to the smoothest stencil. Hence, the most significant contribution to the approximation comes from the smoothest region. Note that in contrast to ENO methods the employed weighting scheme is continuous which is very beneficial to the accurate approximation of the curvature.

In summary, we employ the following fifth order WENO scheme for the treatment of the convective terms in the momentum equations (3.16), in the transport equation (3.17) for the level-set function $\phi$, and in the reinitialization of $\phi$. In the following, we present the scheme for the approximation of the $x$-component of the convective term $\vec{u}^n \cdot \nabla\phi^n$ in the transport equation in the cell centers $(i,j,k)$. To this end, let us consider the $\phi_x^n$-term, i.e., $[\phi_x^n]_{i,j,k}$. Again, we drop the superscript $n$ for the ease of notation.

We obtain $[\phi_x]_{i,j,k}$ via the upwind procedure

$$[\phi_x]_{i,j,k} = \begin{cases} [\phi_x^-]_{i,j,k} & \text{if } u_{i,j,k} > 0 \\ [\phi_x^+]_{i,j,k} & \text{if } u_{i,j,k} < 0 \\ 0 & \text{otherwise} \end{cases}$$

where $[\phi_x^-]$ denotes the left-biased stencil and $[\phi_x^+]$ refers to the right-biased stencil. These biased stencils in our fifth order WENO scheme are defined as

$$\left[\phi_x^{\pm}\right]_{i,j,k} := \omega_1^{\pm}\Big(\frac{q_1^{\pm}}{3} - \frac{7q_2^{\pm}}{6} + \frac{11q_3^{\pm}}{6}\Big) + \omega_2^{\pm}\Big(\frac{-q_2^{\pm}}{6} + \frac{5q_3^{\pm}}{6} + \frac{q_4^{\pm}}{3}\Big) + \omega_3^{\pm}\Big(\frac{q_3^{\pm}}{3} + \frac{5q_4^{\pm}}{6} - \frac{q_5^{\pm}}{6}\Big)$$

where the $q^{\pm}$-terms are given by the following stencils

$$q_1^- = \frac{\phi_{i-2,j,k} - \phi_{i-3,j,k}}{\delta x}, \quad q_2^- = \frac{\phi_{i-1,j,k} - \phi_{i-2,j,k}}{\delta x}, \quad q_3^- = \frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{\delta x},$$
$$q_4^- = \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\delta x}, \quad q_5^- = \frac{\phi_{i+2,j,k} - \phi_{i+1,j,k}}{\delta x}$$

and

$$q_1^+ = \frac{\phi_{i+3,j,k} - \phi_{i+2,j,k}}{\delta x}, \quad q_2^+ = \frac{\phi_{i+2,j,k} - \phi_{i+1,j,k}}{\delta x}, \quad q_3^+ = \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\delta x},$$
$$q_4^+ = \frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{\delta x}, \quad q_5^+ = \frac{\phi_{i-1,j,k} - \phi_{i-2,j,k}}{\delta x}.$$

The respective weights $\omega^{\pm}$ are defined as

$$\omega_1^{\pm} = \frac{\alpha_1^{\pm}}{\alpha_1^{\pm} + \alpha_2^{\pm} + \alpha_3^{\pm}}, \quad \omega_2^{\pm} = \frac{\alpha_2^{\pm}}{\alpha_1^{\pm} + \alpha_2^{\pm} + \alpha_3^{\pm}}, \quad \omega_3^{\pm} = \frac{\alpha_3^{\pm}}{\alpha_1^{\pm} + \alpha_2^{\pm} + \alpha_3^{\pm}}$$

with

$$\alpha_1^{\pm} = \frac{1}{10}\frac{1}{(\tilde{\epsilon} + IS_1^{\pm})^2}, \quad \alpha_2^{\pm} = \frac{6}{10}\frac{1}{(\tilde{\epsilon} + IS_2^{\pm})^2}, \quad \alpha_3^{\pm} = \frac{3}{10}\frac{1}{(\tilde{\epsilon} + IS_3^{\pm})^2}.$$

Here, $\tilde{\epsilon} > 0$ is a regularization parameter (we use $\tilde{\epsilon} = 10^{-6}$ throughout this paper) and $IS^{\pm}$ denotes the WENO smoothness indicators

$$IS_1^{\pm} = \frac{13}{12}(q_1 - 2q_2 + q_3)^2 + \frac{1}{4}(q_1 - 4q_2 + 3q_3)^2,$$
$$IS_2^{\pm} = \frac{13}{12}(q_2 - 2q_3 + q_4)^2 + \frac{1}{4}(q_2 - q_4)^2,$$
$$IS_3^{\pm} = \frac{13}{12}(q_3 - 2q_4 + q_5)^2 + \frac{1}{4}(3q_3 - 4q_4 + q_5)^2.$$

The other derivatives of $\phi$ are approximated in a similar fashion. Note that close to the boundary, where we do not have enough neighbors to compute all the stencils for the $q^{\pm}$-terms, we switch to a third order ENO scheme.

Now we need to approximate the velocity field $\vec{u}$ in the cell-centers $(i, j, k)$. To this end, we employ a third order Lagrange interpolation scheme, see Figure 4.3. For the approximation of the velocity $u$ in $x$-direction, this scheme gives

$$u_{i,j,k} = \frac{1}{16}\Big(-u_{i-\frac{3}{2},j,k} + 9u_{i-\frac{1}{2},j,k} + 9u_{i+\frac{1}{2},j,k} - u_{i+\frac{3}{2},j,k}\Big). \tag{4.13}$$

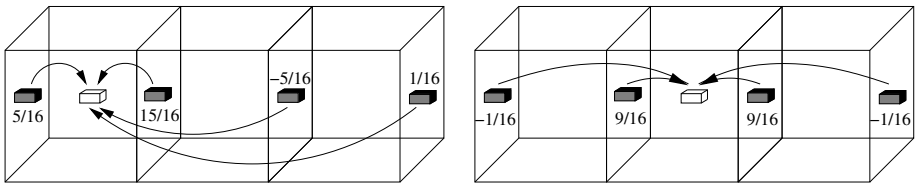


FIGURE 4.3. *Left-weighted (left) and centered (right) Lagrange interpolation scheme.*

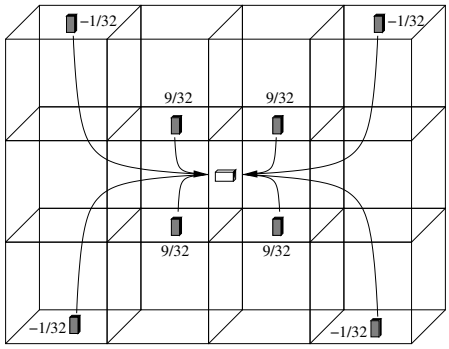


FIGURE 4.4. *Two-dimensional Lagrange interpolation.*

At the boundary of the computational domain we use a right- or left-weighted Lagrange interpolation

$$\begin{aligned}\text{left-weighted}: \quad u_{i,j,k} &= \frac{1}{16}\Big(5u_{i-\frac{1}{2},j,k} + 15u_{i+\frac{1}{2},j,k} - 5u_{i+\frac{3}{2},j,k} + u_{i+\frac{5}{2},j,k}\Big),\\ \text{right-weighted}: \quad u_{i,j,k} &= \frac{1}{16}\Big(u_{i-\frac{5}{2},j,k} - 5u_{i-\frac{3}{2},j,k} + 15u_{i-\frac{1}{2},j,k} + 5u_{i+\frac{1}{2},j,k}\Big).\end{aligned}$$

Note that for the evaluation of these stencils some velocity values need to be interpolated in the cell-face centers, for example the value of the $y$-component $v$ of the velocity field $\vec{u}$ at the position $(i+\frac{1}{2}, j, k)$. To this end, we use a two-dimensional Lagrange interpolation, see Figure 4.4. The velocity value $v$ at $(i+\frac{1}{2}, j, k)$ is given by

$$\begin{aligned}v_{i+\frac{1}{2},j,k} = \frac{1}{32} \quad & (-v_{i-1,j+\frac{3}{2},k} - v_{i-1,j-\frac{3}{2},k} - v_{i+2,j+\frac{3}{2},k} - v_{i+2,j-\frac{3}{2},k} + \\ & 9\ v_{i,j+\frac{1}{2},k} + 9\ v_{i,j-\frac{1}{2},k} + 9\ v_{i+1,j+\frac{1}{2},k} + 9\ v_{i+1,j-\frac{1}{2},k}).\end{aligned}$$

Close to the boundary of the domain, we switch to bilinear interpolation.

**4.3.3. Surface tension, surface normals and curvature.** Finally, we must consider the approximation of the surface tension for the overall discretization of the momentum equation (3.16). Recall that the surface tension force enters as a singular source term which we converted to a volume integral using a delta functional and the CSF-approach. Hence, the main ingredient in the approximation of the surface tension is the use of the approximate delta functional $\delta^\epsilon$ given in (4.9). This leads to a volumetric approximation of the surface tension force with support in an $\epsilon$-neighborhood of the free surface, i.e.,

$$\sigma\kappa\delta(\phi)\vec{n} \approx \sigma\kappa\delta^\epsilon(\phi)\vec{n}.$$

Recall that the unit normal $\vec{n}$ on the interface is given by $\frac{\nabla\phi}{|\nabla\phi|}$ and the curvature in three dimensions is defined as $\kappa = \nabla \cdot \vec{n}$. Note that even though we employ a signed-distance function, i.e., $|\nabla\phi| = 1$ in an $\epsilon$-neighborhood of the free surface, we discretize $\frac{\nabla\phi}{|\nabla\phi|}$ rather than $\nabla\phi$. Similarly, we discretize $\nabla \cdot \vec{n}$ rather than $\Delta\phi$ (compare section 4.5). This procedure allows for a more stable discretization of the surface normals as well as the curvature at cusps and edges. The approximate unit normals are computed as

$$\vec{n}_{i,j,k} := \frac{([\phi_x]_{i,j,k}, [\phi_y]_{i,j,k}, [\phi_z]_{i,j,k})^T}{\sqrt{([\phi_x]^2_{i,j,k} + [\phi_y]^2_{i,j,k} + [\phi_z]^2_{i,j,k})}}$$

using central differences. For the discretization of $\nabla \cdot \vec{n}$ we employ central differences for the first order terms as well as the second order derivatives.

**4.4. Discretization of transport equation for the level-set function.** The discretization of the transport equation (3.17) for the level-set function is based on a second order Adams–Bashford scheme (4.6) for time discretization and the fifth order WENO scheme given above, compare section 4.3.2, for the discretization of the convective term $\vec{u}^n \cdot \nabla\phi^n$. Again, we use the third order Lagrange interpolation scheme (4.13) for the evaluation of the velocities at the cell-centers.

Note that in general the transport of the level-set function will destroy the signed-distance property. Hence, a transported level-set function should not be used for the approximation of curvature or the surface tension. Instead, we refer to the transported level-set function as an intermediate level-set function $\phi^*$; i.e., the time-stepping scheme gives $\phi^*$ and not $\phi^{n+1}$. To obtain a valid signed-distance function $\phi^{n+1}$ from $\phi^*$ a reinitialization must be employed. This reinitialization process will now be discussed in the following section.

**4.5. Reinitalization of level-set function.** The reinitialization of the transported level-set function $\phi^*$ is necessary, since $\phi^*$ does not have the signed-distance property. This, however, is crucial for the approximation of the surface tension. Therefore, a post-processing of the intermediate level-set function $\phi^*$ is necessary. There exist a number of variants for the reinitialization of the level-set function, the one we employ in our implementation is due to Sussman et al. [47]. Here, we give a short review of this scheme for the sake of completeness.

Consider a given function $\phi^*(\vec{x})$ whose zero level-set is the fluid interface. To generate the appropriate signed-distance function $\phi^{n+1}(\vec{x})$ with the same zero level-set as $\phi^*(\vec{x})$, we evolve the following pseudo-transient Hamilton–Jacobi problem to steady state

$$d_\tau + \text{sign}(\phi^*)(|\nabla d| - 1) = 0 \tag{4.14}$$

with initial value $d(\vec{x}, 0) = \phi^*(\vec{x})$. Note that the term $\text{sign}(\phi^*)|\nabla d|$ can be interpreted as motion along the normal direction away from the zero level-set. Since our numerical approach relies on the signed-distance property only in the $\epsilon$-neighborhood of the free surface it is sufficient to compute the solution of (4.14) up to $\tau \approx \epsilon$. Furthermore, the largest deviation from the exact signed-distance due to the transport of the level-set function is in the $\epsilon$-neighborhood. Throughout this paper, we solve the Hamilton–Jacobi system (4.14) up to $\tau = 2\ \epsilon$ for the reinitalization of the level-set function.

For the numerical treatment of (4.14) it is advisable to employ a smoothed signum function $S$, since this leads to better conservation properties as well as a more stable

approximation. In our implementation we employ two different variants of a mollified signum function, one due to Sussman et al. [47]

$$S(\phi^*) = \frac{\phi^*}{\sqrt{(\phi^*)^2 + \delta x^2}} \tag{4.15}$$

and another one due to Peng et al. [41]

$$S(\phi^*) = \frac{\phi^*}{\sqrt{(\phi^*)^2 + |\nabla\phi^*|^2(\delta x^2)}}. \tag{4.16}$$

Note that (4.16) takes the gradient of $\phi^*$ into account which leads to the conservation of the sign of the level-set function, i.e.,

$$\text{sign}(\phi^{n+1}) = \text{sign}(\phi^*)$$

in the entire domain. The Hamilton-Jacobi reinitialization (4.14) is discretized using a third order TVD Runge–Kutta [44] method in time and a fifth order WENO scheme (compare section 4.3.2) in space. The employed third order TVD Runge-Kutta scheme applied to the model problem $\partial_t\psi = L\psi$ can be written as

$$\begin{aligned}
\psi^{(1)} &= \psi^n + \delta t L(\psi^n) \\
\psi^{(2)} &= \psi^n + \frac{\delta t}{4}\left[L(\psi^n) + L(\psi^{(1)})\right] \\
\psi^{n+1} &= \psi^n + \frac{\delta t}{6}\left[L(\psi^n) + 4L(\psi^{(2)}) + L(\psi^1)\right].
\end{aligned} \tag{4.17}$$

Note, that the mollified signum function (4.16) needs to be updated before each Euler-substep in the Runge–Kutta scheme.

**4.6. Poisson solver with non-constant coefficients.** Now that the momentum equations are discretized, it remains to discretize the pressure Poisson equation (4.4). Hence, let us consider the numerical treatment of

$$-\nabla\cdot\frac{1}{\rho^\epsilon(\phi^{n+1})}\nabla\hat{p}^{n+1} = -\nabla\cdot\vec{u}^* \tag{4.18}$$

where the density field $\rho^\epsilon(\phi^{n+1})$ denotes the smoothed density field according to section 4.2. Since the velocities in viscous fluid flow are sufficiently smooth, we can compute the right-hand side of (4.18) with central differences

$$[\nabla\cdot\vec{u}^*]_{i,j,k} = \frac{u^*_{i+\frac{1}{2},j,k} - u^*_{i-\frac{1}{2},j,k}}{\delta x} + \frac{v^*_{i,j+\frac{1}{2},k} - v^*_{i,j-\frac{1}{2},k}}{\delta y} + \frac{w^*_{i,j,k+\frac{1}{2}} - w^*_{i,j,k-\frac{1}{2}}}{\delta z}.$$

The left-hand side of (4.18) is discretized with a standard seven-point-stencil where the smoothed density field is evaluated via an interpolated level-set function, compare (4.13). Thus, the discrete scheme reads as follows

$$\begin{aligned}
\left[\nabla\cdot\frac{1}{\rho^\epsilon(\phi)}\nabla\hat{p}^{n+1}\right]_{i,j,k} = &\frac{1}{(\delta x)^2}\left(\frac{\hat{p}^{n+1}_{i+1,j,k} - \hat{p}^{n+1}_{i,j,k}}{\rho^\epsilon(\phi_{i+\frac{1}{2},j,k})} - \frac{\hat{p}^{n+1}_{i,j,k} - \hat{p}^{n+1}_{i-1,j,k}}{\rho^\epsilon(\phi_{i-\frac{1}{2},j,k})}\right) \\
&+\frac{1}{(\delta y)^2}\left(\frac{\hat{p}^{n+1}_{i,j+1,k} - \hat{p}^{n+1}_{i,j,k}}{\rho^\epsilon(\phi_{i,j+\frac{1}{2},k})} - \frac{\hat{p}^{n+1}_{i,j,k} - \hat{p}^{n+1}_{i,j-1,k}}{\rho^\epsilon(\phi_{i,j-\frac{1}{2},k})}\right) \\
&+\frac{1}{(\delta z)^2}\left(\frac{\hat{p}^{n+1}_{i,j,k+1} - \hat{p}^{n+1}_{i,j,k}}{\rho^\epsilon(\phi_{i,j,k+\frac{1}{2}})} - \frac{\hat{p}^{n+1}_{i,j,k} - \hat{p}^{n+1}_{i,j,k-1}}{\rho^\epsilon(\phi_{i,j,k-\frac{1}{2}})}\right).
\end{aligned}$$

According to section 4.1, we complement the pressure equation with homogeneous Neumann boundary conditions and enforce a vanishing mean for the pressure $p^{n+1} = (\delta t)^{-1}\hat{p}^{n+1}$. For the iterative solution of the resulting semi-definite diffusion equation we employ a parallel Jacobi-preconditioned BiCGSTAB method. Note that this solver is not optimal in the sense that the number of iterations grows as we increase the number of grid cells. More importantly, this solver is not robust with respect to the variation in the density field. The development of optimal and robust solvers is, however, very involved. One possible approach is the use of the so-called black-box multigrid scheme due to Dendy [14] which employs operator or matrix-dependent transfer operators [52]. This approach was further generalized in the so-called algebraic multigrid (AMG) methods [49]. The parallelization of AMG, however, is not straightforward. In our implementation we have included an interface to a parallel AMG solver [35]. In problems with large variations in the density and complicated geometry of the free surface, this solver clearly outperformed the Jacobi-preconditioned BiCGSTAB method. Note, however, that AMG based solvers require the assembly of the system matrix which is not necessary for the Jacobi-preconditioned BiCGSTAB method.

**4.7. Adaptive time step control.** In summary, we employ an explicit scheme for the velocities and an implicit approach for the pressure. Hence, we have to deal with a time step restriction due to the CFL-condition for explicit schemes. The CFL-condition takes convection, viscosity, surface tension and gravity into account and enforces that the discrete information can evolve no further than one grid cell since the discrete difference equations consider only fluxes between adjacent cells.

Here, the convective time step restriction is given by

$$\delta t_c \leq \min\Big(\frac{\delta x}{u_{\max}}, \frac{\delta y}{v_{\max}}, \frac{\delta z}{w_{\max}}\Big), \tag{4.19}$$

where $u_{\max}$, $v_{\max}$, and $w_{\max}$ refer to the maximal absolute values of the velocities $\vec{u}$ on the grid e.g. $u_{\max} := \max_{i,j,k} |u_{i+\frac{1}{2},j,k}|$. The viscous time step restriction is given by

$$\delta t_v \leq \Big(\max\Big\{\frac{\mu_1}{\rho_1}, \frac{\mu_2}{\rho_2}\Big\}\Big(\frac{2}{(\delta x)^2} + \frac{2}{(\delta y)^2} + \frac{2}{(\delta z)^2}\Big)\Big)^{-1}. \tag{4.20}$$

Volume forces $g = (g_1, g_2, g_3)^T$ can be included in the convection estimate (4.19) since e.g. $u_{\max} + |g_1|\delta t$ is an upper bound for the overall horizontal component of the velocity at the end of a time step. Thus, we extend equation (4.19) by the volume force $g$ and obtain a composite time step restriction

$$\delta t^u_{gc} \leq \Big(\frac{u_{\max} + |g_1|\delta t^u_{gc}}{\delta x}\Big)^{-1}$$

with respect to the velocity $u$ in $x$-direction. Solving for $\delta t^u_{gc}$, this leads to

$$\delta t^u_{gc} \leq 2\left(\frac{u_{\max}}{\delta x} + \sqrt{\Big(\frac{u_{\max}}{\delta x}\Big)^2 + \frac{4|g_1|}{\delta x}}\right)^{-1}. \tag{4.21}$$

The time step estimates for the components $v$ and $w$ of the velocity including the volume force components $g_2$ and $g_3$, respectively, are analogous to (4.21).

However, the combined force due to convective and diffusive forces must also be considered. To this end, we insert the viscous time step restriction (4.20) into equation (4.21). Altogether with $V := \max\left\{\frac{\mu_1}{\rho_1}, \frac{\mu_2}{\rho_2}\right\}\left(\frac{2}{(\delta x)^2}+\frac{2}{(\delta y)^2}+\frac{2}{(\delta z)^2}\right)$ this yields the time step restriction $\delta t^u_{gcv}$ via

$$\delta t^u_{gcv} \leq 2\left(\left(\frac{u_{\max}}{\delta x}+V\right)+\sqrt{\left(\frac{u_{\max}}{\delta x}+V\right)^2+\frac{4|g_1|}{\delta x}}\right)^{-1} \tag{4.22}$$

for the $x$-component $u$ of the velocity $\vec{u}$.

Finally, it remains to consider the amount of acceleration due to surface tension, which is included as a volume force, compare section 4.3.3. Recall that the smoothed volume force due to surface tension is given by $\frac{1}{\rho}(\sigma\delta^\epsilon\kappa)$. Here, the maximal value of $\delta^\epsilon$ is $\frac{1}{\epsilon}$, where $\epsilon = \alpha\delta x$ describes the amount of smoothing in (4.9). The value of the regularized density at the free surface is $\frac{1}{2}(\rho_2+\rho_1)$. With the definition $\kappa_{\max} := \max_{i,j,k}|\kappa_{i,j,k}|$, we obtain $\frac{2\ \kappa_{\max}\sigma}{\alpha\delta x(\rho_2+\rho_1)}$ as an approximation of the velocity induced by surface tension. Since this velocity is included as a volume force in the right-hand side of the momentum equations, it can be handled in a similar fashion as $g_1$ in (4.22). This leads to the overall time step restriction for the $u$ component of the velocity field

$$\delta t^u_{gcvs} \leq 2\ C_u$$

where

$$C_u := \left(\left(\frac{u_{\max}}{\delta x}+V\right)+\sqrt{\left(\frac{u_{\max}}{\delta x}+V\right)^2+\frac{4|g_1|}{\delta x}+\frac{8\ \kappa_{\max}\sigma}{\alpha(\rho_2+\rho_1)(\delta x)^2}}\right)^{-1}.$$

The bounds for $\delta t^v_{gcvs}$ and $\delta t^w_{gcvs}$ with respect to the $v$ and $w$ velocities involving $C_v$ and $C_w$ are defined in an analogous way. Applying this scheme to all three velocity components, we obtain the appropriate time step restriction $\delta t^{u/v/w}_{gcvs} \leq 2\ C_{u/v/w}$ for each component separately. The final time step restriction can then be computed easily as the minimum of these three scalar bounds. This yields the overall time step restriction

$$\delta t \leq 2\ \xi\ \min_{\Omega}(C_u, C_v, C_w)$$

with $\xi \in (0,1]$ as a safety factor. Throughout this paper, we employ this adaptive time step control using a safety factor of $\xi = 0.3$ in our computations.

**4.8. Parallelization.** For the understanding of physical phenomena in the area of multi-phase flows, the investigation of fluid-fluid interactions on very small scales is essential. However, a direct numerical simulation (DNS) on such small scales requires extremely high resolutions. Hence, an enormous amount of computer memory and computing time is necessary to obtain a valid approximation. These issues make the use of massively parallel high-performance computers a must for multi-phase flow problems in three space dimensions.

To this end, we parallelized our two-phase flow solver with surface tension and the level-set reinitialization procedure by means of a classical domain decomposition method using the Message Passage Interface (MPI) [23]. Here, the main issue is of course the exchange of all relevant information during the computation between the

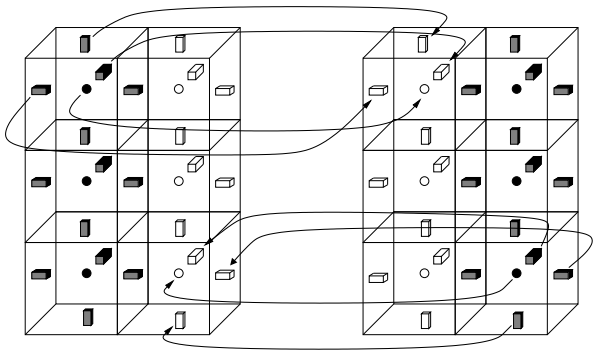

FIGURE 4.5. *Exchange of pressure, level-set and velocity values for one ghost cell slice. White bricks denote ghost cell values and black bricks refer to subdomain values.*

different processors. Since we consider uniform grids only, the decomposition of the computational domain and the resulting communication pattern is straightforward and requires no special attention, see [21]. The communication volume, however, is dependent on the employed finite difference stencil.

The width of the finite difference stencils we employ in our implementation ranges from three to seven grid cells. Therefore, we need to attach up to three slices of boundary ghost cells to each neighboring subdomain. These boundary values are – unless they fall outside the computational domain $\Omega$ – computed by the processes assigned to the neighboring subdomains and hence are not directly available. They must rather be sent by the neighboring processes in a communication phase inside the time-stepping loop. The schematic depicted in Figure 4.5 shows the exchange of the relevant values within one ghost cell slice. Note that in contrast to parallel single-phase flow solvers we also need to exchange the ghost cell values located at corners and edges of each subdomain for the computation of the two-dimensional Lagrange interpolation and the accurate approximation of curvature. Hence, the values for all ghost cells which have at least one node in common with the respective subdomain must be communicated in our two-phase level-set flow solver with surface tension which can be written in the following algorithmic form.

ALGORITHM 1 (Parallel level-set two-phase flow solver).

1. Initialize $\delta x$, $\delta y$, $\delta z$, $T$, $\alpha$, $\phi_0$ and $\Omega$.
2. Set local domain $\Omega^q$.
3. Set $t := 0$, $n := 0$, $h := \max(\delta x, \delta y, \delta z)$, and $\epsilon := \alpha h$.
4. Enforce $\phi_0$ to be a global signed-distance function, e.g. solve (4.14) with $\tau = 1$.
5. Set initial values $\vec{u}^n := \vec{u}_0$, $p^n := p_0$, and $\phi^n := \phi_0$ on local domain $\Omega^q$.
6. Set boundary values for $\vec{u}^n$ on local boundary $\partial\Omega^q \cap \partial\Omega$.
7. TIME-LOOP: While $t \leq T$:
   (a) TIME STEP RESTRICTION:
   Compute local time step $\delta t^q$ according to section 4.7.
   Compute global time step $\delta t := \min_{\hat{q}} \delta t^{\hat{q}}$.
   (b) MOMENTUM EQUATION I:
   Exchange ghost cell values for $u^n$ and $\phi^n$.
   Compute intermediate velocity field $\vec{u}^*(\vec{u}^n, p^n, \phi^n)$ on local domain $\Omega^q$ according to section 4.1.

Set boundary values for $\vec{u}^*$ on local boundary $\partial\Omega^q \cap \partial\Omega$.

(c) PRESSURE POISSON EQUATION I:

Exchange ghost cell values for $u^*$.

Set right-hand side $b := \nabla \cdot \vec{u}^*$ on local domain $\Omega^q$ for pressure Poisson equation.

(d) TRANSPORT EQUATION:

Compute intermediate level-set function $\phi^*(\vec{u}^n, \phi^n)$ according to section 4.4.

(e) Set $\psi^n := \phi^*$.

(f) REINITALIZATION-LOOP: While $\tau < 2\ \epsilon$:

i. Exchange ghost cell values for $\psi^n$, $\psi^{(1)}$, and $\psi^{(2)}$.

ii. Set $\psi^{(1)}$, $\psi^{(2)}$, and $\psi^{n+1}$ according to (4.17) on local domain $\Omega^q$.

(g) Set $\phi^{n+1} = \psi^{n+1}$ and $\hat{p}_{it} := p^n \delta t$.

(h) PRESSURE POISSON EQUATION II:

Solve for $\hat{p}^{n+1}(\vec{u}^*, \phi^{n+1})$ by preconditioned Krylov-method.

Exchange ghost cell values for $\hat{p}_{it}$ per iteration $it$ as required.

(i) Set $p^{n+1} := (\delta t)^{-1} \hat{p}_{it}$.

(j) MOMENTUM EQUATION II:

Set $\vec{u}^{n+1}$ according to section 4.1 on local domain $\Omega^q$.

Set boundary values for $\vec{u}^{n+1}$ on local boundary $\partial\Omega^q \cap \partial\Omega$.

(k) Set $t = t + \delta t$ and $n = n + 1$.

**5. Numerical Results.** In this section we present the results of our numerical experiments. Here, we consider three-dimensional problems only. First we investigate the mass conservation due to reinitialization, where we compare the influence of different higher order schemes and the smoothing of the signum function. In our second example we investigate the order of convergence obtained for the curvature by means of a diagonally transported unit sphere. Here we also compare the results obtained with and without the use of a smoothed delta functional. Then we examine the global order of convergence of our overall numerical scheme for two-phase flow problems including surface tension on the basis of a rising bubble test problem. Finally, regarding real world applications we compare our numerical scheme with experimental data given for the collapsing water column benchmark and study the qualitative behavior of the dynamic contact angle in dependence of the surface tension.

**5.1. Investigation of mass conservation due to reinitialization.** In the following we investigate different higher order ENO and WENO discretization schemes employed in the Hamilton–Jacobi reinitialization procedure given in section 4.5. In particular, we are interested in the influence of these schemes with respect to mass conservation; i.e. conservation of the sign of the level-set function. To this end, we consider a cube in three dimensions located inside the domain $\Omega = [0, 1]^3$. This domain is discretized with a uniform $40^3$ grid and as initial condition we use $\phi_{i,j,k} = 1$ for the gridpoints $i, j, k \in \{7, \ldots, 34\}$ and $\phi_{i,j,k} = -1$ elsewhere. Since the initial condition for $\phi$ is not a signed distance function we iterate the reinitialization equation (4.14) until $\tau = 1$. Then the characteristic information has evolved up from the zero level-set through the complete domain and we obtain an approximate global signed distance function. On the basis of this cube test problem, we can observe the influence of numerical diffusion and thus the loss in mass introduced by the reinitialization procedure with different higher order schemes and different smoothing for the signum function.

Figure 5.1 shows the zero level-set for $\tau = 1$, where at the top line the reinitializa-

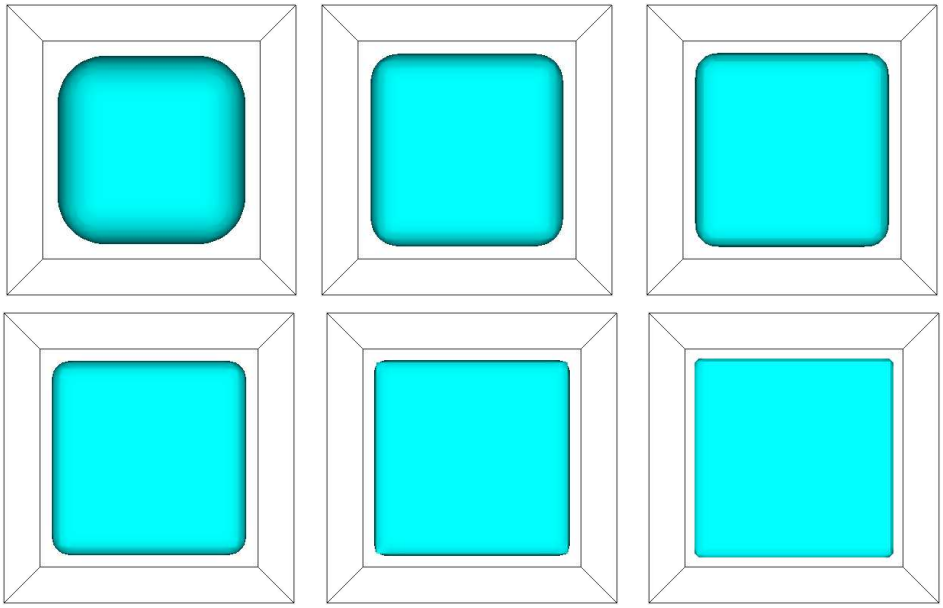

FIGURE 5.1. *Reinitialization with a first (left), a second (center), and a third order (right) ENO-scheme (top row) and with a fifth order WENO-scheme using* (4.15) *as mollified signum function (left bottom row) and* (4.16) *as mollified signum (center bottom row), initial condition (right bottom row).*

TABLE 5.1
*Mass after the reinitialization in percent of the initial mass*

| ENO-1 with (4.15) | ENO-2 with (4.15) | ENO-3 with (4.15) | WENO-5 with (4.15) | WENO-5 with (4.16) |
|---|---|---|---|---|
| 84.80% | 95.66% | 96.41% | 98.39% | 100.0% |

tion function is discretized in space using a first (left), second (center), and third order ENO-scheme (right) including the smoothed signum function (4.15). The left picture from the bottom line of Figure 5.1 shows the corresponding zero level-set obtained using a fifth order WENO scheme together with (4.15). The center picture of the bottom line shows the level-set solution using a fifth order WENO together with the smoothed signum function (4.16). The zero level-set of the initial condition is shown on the lower right of Figure 5.1. From these pictures, we can clearly observe that the WENO schemes outperform the ENO schemes. Furthermore, the combination of the fifth order WENO scheme with the smoothed signum function (4.16) shows a substantially superior performance than its combination with (4.15). In fact, we find perfect conservation of the sign of the level-set function.

In Table 5.1 we give the measured values for the discrete mass which was computed by

$$\int_{\Omega} \rho(\phi(\vec{x}))\ d\vec{x} \approx (\Delta x)^3 \sum_{i,j,k} \rho(\phi_{i,j,k},),$$

without smoothing the density function $\rho$. These results underline the observed influence of the discretization schemes with respect to the reinitialization procedure.

TABLE 5.2

*Convergence study for approximation of curvature. Given are the $L_h^1$-, the $L_h^2$- and $L_h^\infty$-norms of the errors $\Delta\kappa_h$ and $\Delta\kappa_h^\epsilon$ and the respective convergence rates $\gamma_h$.*

| | $L_h^1$-error | | | | $L_h^2$-error | | | | $L_h^\infty$-error | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $N^3$ | $\Delta\kappa_h$ | $\gamma_h$ | $\Delta\kappa_h^\delta$ | $\gamma_h^\delta$ | $\Delta\kappa_h$ | $\gamma_h$ | $\Delta\kappa_h^\delta$ | $\gamma_h^\delta$ | $\Delta\kappa_h$ | $\gamma_h$ | $\Delta\kappa_h^\delta$ | $\gamma_h^\delta$ |
| $25^3$ | $3.35_{-1}$ | — | $6.47_{-1}$ | — | $7.24_{-2}$ | — | $1.59_{-1}$ | — | $5.92_{2}$ | — | $1.24_{-1}$ | — |
| $50^3$ | $4.08_{-2}$ | 3.03 | $1.64_{-1}$ | 1.98 | $8.85_{-3}$ | 3.03 | $4.29_{-2}$ | 1.89 | $7.10_{-3}$ | 3.05 | $3.51_{-2}$ | 1.82 |
| $100^3$ | $7.88_{-3}$ | 2.37 | $6.53_{-2}$ | 1.33 | $1.78_{-3}$ | 2.31 | $1.80_{-2}$ | 1.25 | $1.01_{-3}$ | 2.81 | $1.21_{-2}$ | 1.53 |
| $200^3$ | $1.93_{-3}$ | 2.03 | $3.22_{-2}$ | 1.02 | $4.39_{-4}$ | 2.01 | $8.95_{-3}$ | 1.01 | $2.20_{-4}$ | 2.19 | $6.10_{-3}$ | 0.99 |

Figure 5.1 as well as Table 5.1 point out a significant improvement in mass conservation, when the fifth order WENO scheme is applied together with the modified signum function (4.16). We can clearly observe the expected conservation of the sign of the level-set function during the reinitalization. Hence, in our further experiments we always employ (4.16) together with a fifth order WENO scheme during the reinitialization.

**5.2. Convergence study for the curvature for a transport problem.** In this section we investigate the convergence behavior of our numerical scheme with respect to curvature considering the simple test-case of an advected unit sphere. Since the curvature of the unit sphere is known analytically, we can directly compare the approximated curvature with the exact solution.

To this end we consider the domain $\Omega = [0, 4]^3$ using a sequence of uniformly refined equidistant grids. Furthermore, we use periodic boundary conditions on $\partial\Omega$ and a stationary velocity field

$$u(x, y, z) = 1, \quad v(x, y, z) = 1, \quad w(x, y, z) = 0.$$

As initial condition for the level-set function we use the exact distance function

$$\phi(x, y, z) = 1 - \sqrt{(x-2)^2 + (y-2)^2 + (z-2)^2}$$

for a sphere with radius $r = 1$ centered at $(2, 2, 2)$. The order of convergence $\gamma_h$ is given by

$$\gamma_h := \frac{\log\left(\frac{\|\kappa_{2h} - \kappa\|}{\|\kappa_h - \kappa\|}\right)}{\log 2}, \tag{5.1}$$

where $\kappa$ is the exact curvature and $\kappa_h$ and $\kappa_{2h}$ are the discrete curvatures computed on two successive grids; i.e., the grids with meshsize $h$ and $2h$. We consider the time interval $[0, T]$ with $T = 4$, so that the unit sphere passes through the whole domain along the $x$-$y$-diagonal, i.e., we simulate exactly one period. The time step size in this experiment was $\delta t = h^2$. The curvature is computed in all grid cells with $|\phi_h| < \epsilon$ where $\epsilon = \alpha\ h$ with $\alpha = 1.5$. Here, we also computed the following differences

$$\Delta\kappa_h := \kappa_h(\phi_h) - \kappa(\phi) \quad \text{and} \quad \Delta\kappa_h^\delta := \Delta\kappa_h \delta^\epsilon(\phi_h),$$

where the superscript $\delta$ denotes if the approximation error $\Delta\kappa_h$ was weighted by the approximate delta functional $\delta^\epsilon$. Recall that we need to employ a smoothed delta functional when we take surface tension effects into account. On the basis of these

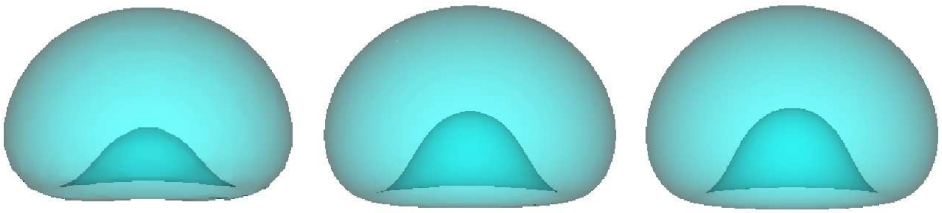

FIGURE 5.2. *Rising bubble at time t=0.075s for $N = 36^3$, $N = 72^3$ and $N = 144^3$.*

TABLE 5.3
*Parameter for rising bubble problem. Fluid$_1$ denotes the bubble and fluid$_2$ the surrounding fluid.*

| | | | |
|---|---|---|---|
| fluid$_1$: | $\mu_1 = 2.5_{-4} kg/ms$, $\rho_1 = 1.0_0 kg/m^3$ | surface tension: | $\sigma = 5.0_{-3} N/m$ |
| fluid$_2$: | $\mu_2 = 5.0_{-4} kg/ms$, $\rho_2 = 1.0_1 kg/m^3$ | surface thickness: | $\epsilon = 1.6h$ |
| forces: | $(g_1, g_2, g_3) = (0, -9.81, 0) m/s^2$ | boundary conditions: | slip |

values, we finally approximated the norms $\|\Delta\kappa_h^{(\delta)}\|_{L^1}$, $\|\Delta\kappa_h^{(\delta)}\|_{L^2}$ and $\|\Delta\kappa_h^{(\delta)}\|_{L^\infty}$ by

$$\begin{aligned} \|\Delta\kappa_h^{(\delta)}\|_{L_h^1} &:= \textstyle\sum_{|\phi_h|<\epsilon} |\Delta\kappa_h^{(\delta)}| h^2 \\ \|\Delta\kappa_h^{(\delta)}\|_{L_h^2} &:= \sqrt{\textstyle\sum_{|\phi_h|<\epsilon} (\Delta\kappa_h^{(\delta)})^2 h^2} \\ \|\Delta\kappa_h^{(\delta)}\|_{L_h^\infty} &:= \max_{|\phi_h|<\epsilon} |\Delta\kappa_h^{(\delta)}|. \end{aligned}$$

The measured values for this test problem are given in Table 5.2. From these numbers we can clearly observe the anticipated second order convergence for the approximation of the curvature when we do *not* employ a smoothed delta functional. The rates obtained *with* the smoothed delta functional show essentially a first order convergence as expected since the smoothing of the delta functional is of order $\epsilon = \alpha\, h$, i.e., we obtain an approximation of order $O(h)$ since $\Delta\kappa_h^\delta = \Delta\kappa_h \delta^\epsilon \approx O(h^2) O(\epsilon^{-1}) \approx O(h)$. Note that a higher order approximation of the delta functional might increase the order of convergence, but on the other hand may lead to stability problems for the numerical scheme.

**5.3. Convergence study of a rising bubble.** To determine the convergence behavior of the complete two-phase flow solver including surface tension, we now consider the rising bubble problem. We anticipate to find first order convergence near the free surface due to the smoothing of the delta functional whereas away from the free surface we expect to obtain second order approximation since there, the approximate delta functional is not active. Hence, we anticipate to find a global approximation order close to two.

In this experiment we use a sequence of three refined equidistant grids on the computational domain $\Omega = [0, 0.15m]^3$, namely we use the grids with meshwidth $h = 1/145$, $2h = 1/73$ and $4h = 1/37$ together with the equidistant time steps $\delta t_h = 6.25_{-5}$, $\delta t_{2h} = 2.50_{-4}$, and $\delta t_{4h} = 1.00_{-3}$ respectively. The initial condition for the level-set function $\phi$ corresponds to a spherical bubble with center $(0.075m, 0.05m, 0.075m)$ and radius $r = 0.025m$. All further physical properties are listed in Table 5.3.

We consider the absolute errors [48]

$$E_{L_h^1} = \frac{1}{N} \sum |I_{2h,4h}^h \psi_{2h,4h} - \psi_h| \quad \text{and} \quad E_{L_h^\infty} = \max |I_{2h,4h}^h \psi_{2h,4h} - \psi_h| \qquad (5.2)$$

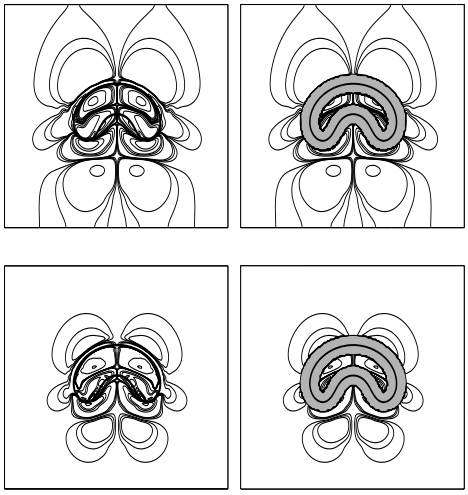

FIGURE 5.3. *Contour plot of the error in the velocity in x-direction between $u_h$ and $u_{4h}$ (top row) and $u_h$ and $u_{2h}$ (bottom row) on the cut-plane $(i, j, N/2)$. Depicted is the free surface with respect to the finest mesh (bold line), the contour lines for the values -0.06, -0.04, -0.02, -0.015, -0.01, -0.0015, -0.001, -0.0005, 0.0005, 0.001, 0.0015, 0.01, 0.015, 0.02, 0.04, 0.06. On the right-hand side the smoothing region on the coarse mesh, i.e. $|\phi_{4h}| < 1.6 \cdot 4h$, is indicated by the gray shaded area.*

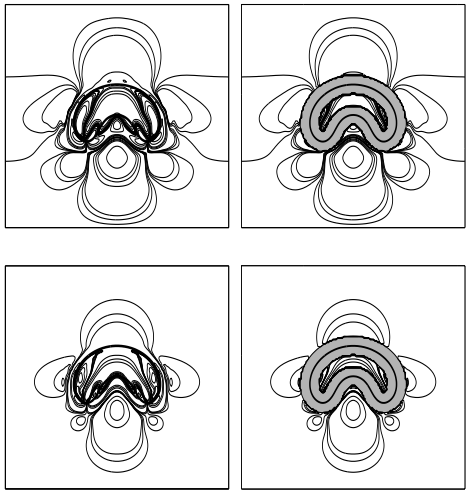

FIGURE 5.4. *Contour plot of the error in the velocity in y-direction between $v_h$ and $v_{4h}$ (top row) and $v_h$ and $v_{2h}$ (bottom row) on the cut-plane $(i, j, N/2)$. Depicted is the free surface with respect to the finest mesh (bold line), the contour lines for the values -0.06, -0.04, -0.02, -0.015, -0.01, -0.0015, -0.001, -0.0005, 0.0005, 0.001, 0.0015, 0.01, 0.015, 0.02, 0.04, 0.06. On the right-hand side the smoothing region on the coarse mesh, i.e. $|\phi_{4h}| < 1.6 \cdot 4h$, is indicated by the gray shaded area.*

TABLE 5.4
*Convergence study for rising bubble problem at time $t = 0.075s$.*

| | $E_{L_h^\infty}$-error | | | $E_{L_h^1}$-error | | |
|---|---|---|---|---|---|---|
| | $37^3/145^3$ | $73^3/145^3$ | $\gamma_h$ | $37^3/145^3$ | $73^3/145^3$ | $\gamma_h$ |
| $u$ | $1.43_{-1}$ | $4.95_{-2}$ | 1.54 | $1.23_{-3}$ | $3.50_{-4}$ | 1.81 |
| $v$ | $1.42_{-1}$ | $4.64_{-2}$ | 1.61 | $2.01_{-3}$ | $6.08_{-4}$ | 1.72 |
| $w$ | $1.43_{-1}$ | $4.95_{-2}$ | 1.54 | $1.23_{-3}$ | $3.50_{-4}$ | 1.81 |
| $\phi$ | $3.19_{-3}$ | $1.11_{-3}$ | 1.51 | $8.00_{-4}$ | $2.25_{-4}$ | 1.82 |

where we used a third order spline interpolation $I^h_{2h,4h}$ to interpolate the coarse grid values $\psi_{2h,4h}$ to the finest grid before computing the difference of two solutions. Here, $N$ is the total number of gridpoints on the finest grid. We determine the numerical order of convergence $\gamma_h$ analogously to (5.1). In Figures 5.3 and 5.4 several contour plots on the cut-plane $(i, j, N/2)$ of the error $\psi_{2h,4h} - \psi_h$ are given for the velocity in $x$-direction, i.e. $\psi = u$, and the velocity in $y$-direction, i.e. $\psi = v$. Depicted are the free-surface on the finest mesh, the contour-lines of the error and the smoothing region with respect to the coarse mesh with spacing $4h$. From these plots we can clearly observe the expected fast convergence away from the free surface (outside the smoothing region) and the larger error gradients near the interface (inside the smoothing region). The measured errors and convergence rates at time $t = 0.075s$ are given in Table 5.4, see also Figure 5.2. These numbers show that we obtain the anticipated convergence behavior of the overall scheme including surface tension. The

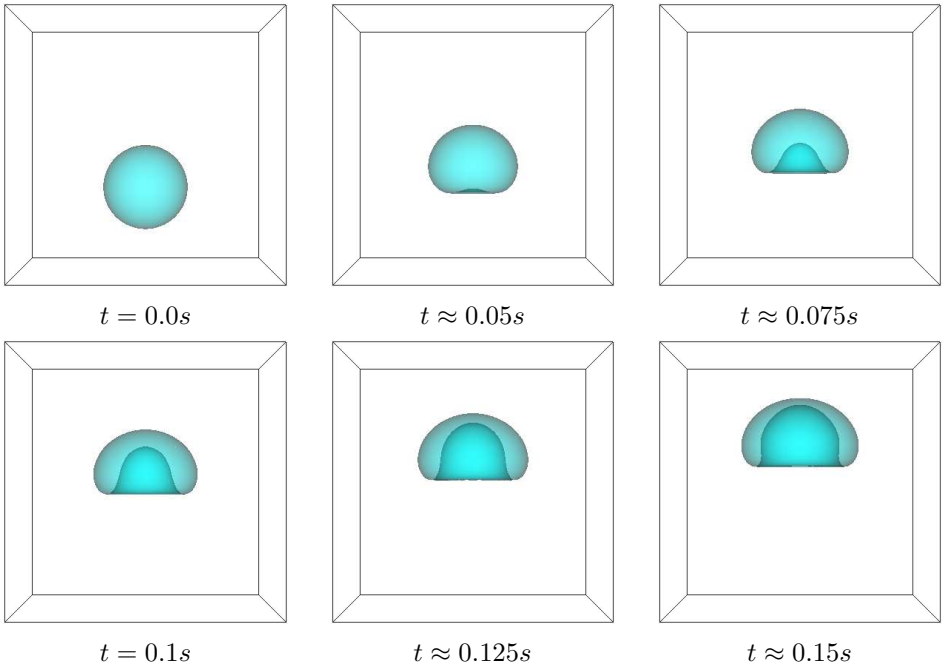


FIGURE 5.5. *Rising bubble on finest grid with meshwidth* $h = 1/145$.

TABLE 5.5
*Study for mass conservation at time* $t = 0.075s$

| Conserved mass in percent of initial mass | | | |
|---|---|---|---|
| | $4h$ | $2h$ | $h$ |
| present mass | 96.06% | 98.95% | 99.81% |
| mean mass | 98.91% | 99.55% | 99.88% |

rates obtained for the $L_h^1$-norm are between 1.7 and 1.8 and slightly larger than the rates obtain for the $L_h^\infty$-norm. This is of course due to the fact that the largest error occurs close the free surface. Figure 5.5 shows the temporal evolution of the rising bubble on the finest grid, where the free surface is depicted slightly transparent.

We also investigated the overall mass conservation of our scheme for this model problem. Here, we approximated the mass conservation in the $n$th time step, i.e. at time $t_n$, by

$$M_h^n = \frac{\sum\limits_{i,j,k} \rho^\epsilon(\phi_{i,j,k}^n)h^3}{\sum\limits_{i,j,k} \rho^\epsilon(\phi_{i,j,k}^0)h^3}$$

which describes the relationship of the current mass at time $t_n$ to the initial mass. Furthermore, we also approximated the temporal mean by

$$\Delta M_h^n = \frac{1}{t_n} \sum_{q=0}^{n} M_h^q \delta t_q$$

where $\delta t_q$ refers to the time step size in the $q$th time step. The computed values with

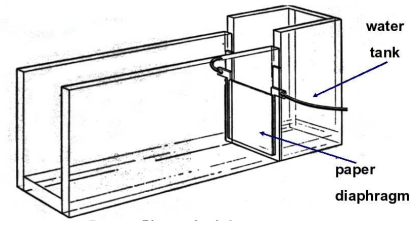


FIGURE 5.6. *Diagram of a typical apparatus from J.C. Martin and W.J. Moyce.*

TABLE 5.6
*Parameter for collapsing water column. Fluid$_1$ denotes the water column and fluid$_2$ the surrounding air.*

| | | | |
|---|---|---|---|
| fluid$_1$: | $\mu_1 = 1.002_{-3} kg/ms$, $\rho_1 = 9.982_2 kg/m^3$ | surface tension: | $\sigma = 5.0_{-3} N/m$ |
| fluid$_2$: | $\mu_2 = 1.810_{-5} kg/ms$, $\rho_2 = 1.205_0 kg/m^3$ | surface thickness: | $\epsilon = 1.6h$ |
| forces: | $(g_1, g_2, g_3) = (0, -9.81, 0) m/s^2$ | CFL-number: | 0.3 |

$t_n = 0.075s$ are given in Table 5.5. From these numbers we can observe the good conservation properties of our scheme as well as the convergence as the grid is refined.

**5.4. Collapsing Water Column.** To validate our numerical scheme we also compared our simulations to experimental data. To this end, we computed the well-known collapsing water column benchmark problem and compared our results with the experimental data from Martin and Moyce [33]. A sketch of the experimental setup is given in Figure 5.6.

In our numerical simulation we rebuild the experimental setup in the computer. We consider the computational domain $0.2286m \times 0.085725m \times 0.05715m$ and employ a uniform grid of size $80 \times 30 \times 20$ for the discretization. The initial water column has a height of $0.05715m$ and a width of $0.05715m$ and is located on the right-hand side of the reservoir. As boundary conditions we used outflow boundaries for the wall on the top and on the left hand side, on all other walls we used classical slip boundary conditions. See Table 5.6 for further parameters used in the simulation of this example.

The result of our numerical simulation are given in Figure 5.7 and Figure 5.8. From the snapshots of the free surface depicted in Figure 5.7 we can observe the time evolution of the collapsing column, which is in good agreement with the experimental data. This is also clear from the graphs depicted in Figure 5.8, where we plotted the position of the leading edge of the collapsing column (left) as well as the height of the column at the right-hand boundary (right) over time. For comparison the graphs also give the experimental data obtained by Martin and Moyce [33]. Here, we can observe that our numerical scheme is in very good agreement with the experiment. However, the simulation seems to be ahead of the physical data with respect to the position of the leading edge. We believe that this shift is due to the unsynchronized start of the simulation and the physical experiment. The removal of the dam in the real world experiment cannot be instantaneous, whereas in the numerical simulation it is. This accounts for the time-gap between the simulation and the experiment. This conjecture is supported by the fact, that a positive shift of the measured numerical data by $0.007s$ brings the numerical data in perfect alignment with the experimental data.

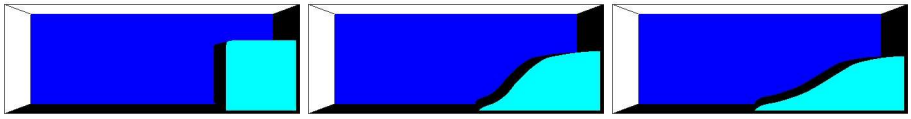

FIGURE 5.7. *Transient free surfaces of the collapsing water column problem.*

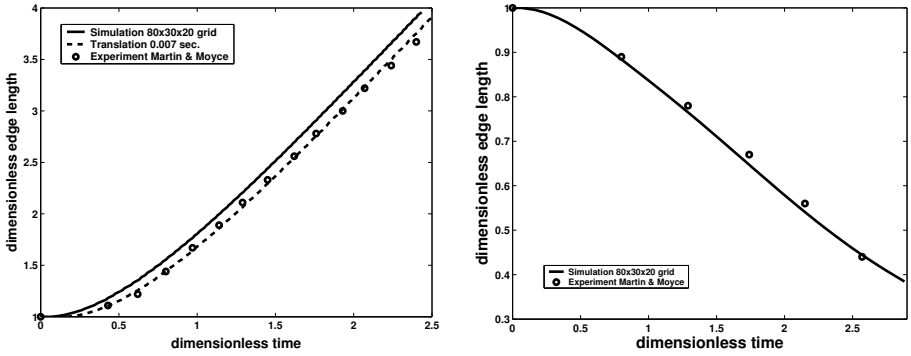


FIGURE 5.8. *Numerical results compared with experimental results of J.C. Martin and W.J. Moyce.*

**5.5. Influence of surface tension and gravity on the dynamic contact angle.** In our last numerical experiment we consider the evolution of the dynamic contact angle for a semi-spherical drop moving down an inclined surface with inclination of 45°. Note that the inclination is encoded via a rotated gravitational force. The computational domain in this experiment is set to be of length= $15mm$, height= $2mm$ and width= $5mm$. Here, we completed four different simulations. Two simulations without surface tension ($\sigma = 0.0N/m$) and two with surface tension using $\sigma = 0.007275N/m$. Furthermore, we employed two different values for the gravitational force, namely $g = -6.93m/s^2$ and $g = -9.81m/s^2$, see Table 5.7 for the complete list of parameters.

Since the moving contact line causes a singularity in the stress tensor, it is widely believed that slip occurs in the neighborhood of the contact line. Therefore, one usually employs a specialized slip condition, globally or near the contact line. The simplest relation used to model this slip condition is the so-called Navier slip condition which has the form

$$u = \beta \frac{\partial u}{\partial y}. \tag{5.3}$$

Here, the slip coefficient $\beta$ has the dimension of length, and can be interpreted as the distance from the boundary where an extrapolated velocity profile will vanish. In most applications, this slip length is much smaller than any realistic meshsize for numerical simulation. Numerically, we can impose a slip length on the order of the

TABLE 5.7
*Parameter for sliding drop. Fluid$_1$ denotes the drop and fluid$_2$ the surrounding air.*

| | | | |
|---|---|---|---|
| fluid$_1$: | $\mu_1 = 1.002_{-3} kg/ms$, $\rho_1 = 9.982_2 kg/m^3$ | surface tension: | $\sigma = 5.0_{-3} N/m$ |
| fluid$_2$: | $\mu_2 = 1.810_{-5} kg/ms$, $\rho_2 = 1.205_0 kg/m^3$ | surface thickness: | $\epsilon = 1.6h$ |

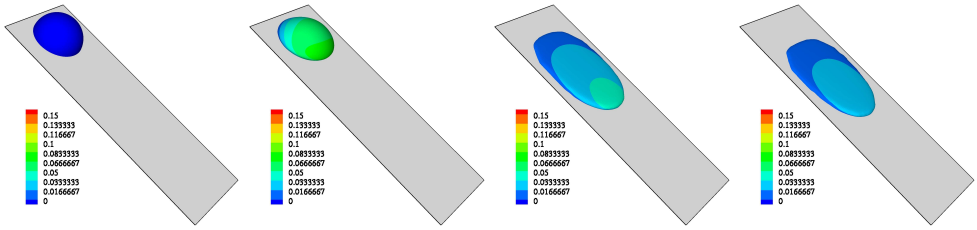

FIGURE 5.9. *Sliding drop for g=6.93* without *surface tension. The color-contour shows the velocity in x-direction.*

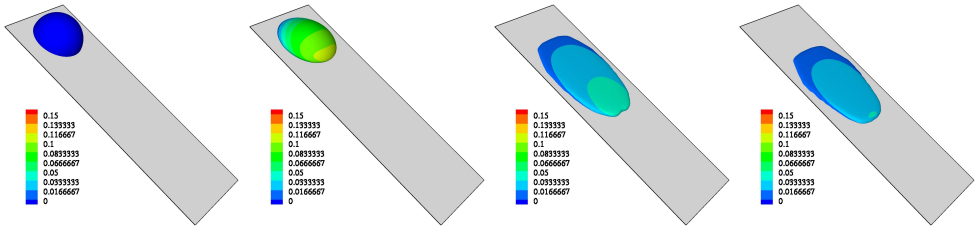

FIGURE 5.10. *Sliding drop for g=9.81* without *surface tension. The color-contour shows the velocity in x-direction.*

meshsize only. In our experiment we simply choose $\beta = \frac{\delta y}{2}$ and enforce (5.3) globally.

The results of our experiments *without* surface tension are given in Figure 5.9 and Figure 5.10, where several snapshots of the free surface are depicted at different time steps. Since the surface tension force is not considered, the drop collapses rapidly and the viscous terms get dominant over time causing a slow-down in the movement.

The results of our experiments *including* surface tension are given in Figure 5.11 and Figure 5.12, where several snapshots of the free surface are depicted at different time steps. Furthermore, we measured the contact angles and the $x$-velocities along the solid wall at the upper leading edge of the drop and at the lower leading edge. The respective graphs are also given in Figure 5.11 and Figure 5.12. Note that the small oscillations visible in the graphs are due to the fact that we cannot measure the contact angle directly on the free surface since the zero level-set usually does not pass through the grid points. Therefore, we approximate the position of the free surface by choosing the center point between two adjacent grid points where the level-set function changes sign. More precisely, we focus on the level-set values located at the vertical center cut-plane $(i, j, k = N/2)$ directly above the substrate. If two neighboring level-set values $\phi_{i,1,N/2}$ and $\phi_{i+1,1,N/2}$ have different sign, we compute the contact-angle at the location $(i + \frac{1}{2}, 1, N/2)$ using central differences in tangential direction and weighted one-sided differences in normal direction. Hence, we expect to find small oscillations on the scale of the meshsize. The graphs given in Figure 5.11 and Figure 5.12 clearly show that the contact angle increases with increasing velocity of the moving contact line. This qualitative behavior is also observed in physical experiments [4, 18].

**6. Concluding remarks.** In this paper we presented a parallel incompressible Navier–Stokes solver for two-phase flow problems with surface tension in three dimensions. Our scheme employs a standard staggered grid finite difference discretization and a projection method with a fifth order WENO scheme.

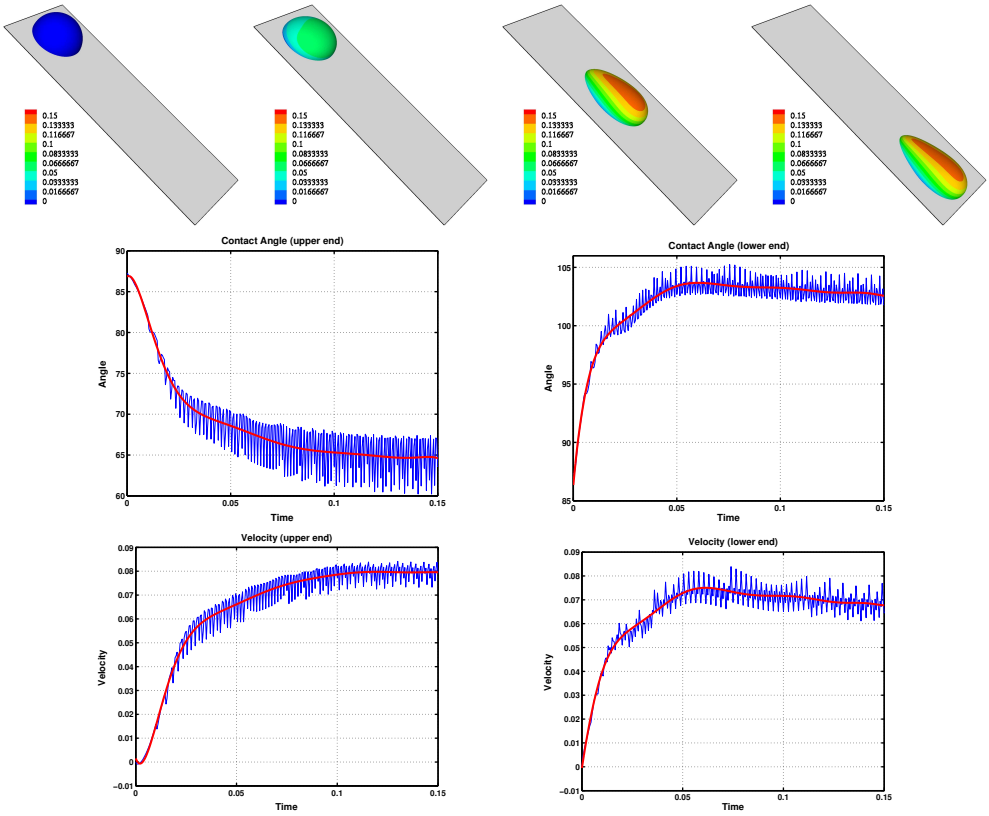


FIGURE 5.11. *Sliding drop for g=6.93* with *surface tension (upper row). The color-contour shows the velocity in x-direction. The dynamic contact angle (center row) at the upper and lower edge of the drop and the respective velocities in x-direction (lower row).*

The free surface between the two fluid phases is tracked with a level-set approach. Here, the interface conditions are implicitly incorporated into the momentum equations by the CSF method and surface tension is evaluated using a smoothed delta functional and an appropriate third order interpolation scheme. Altogether, our approach exhibits a second order convergence for the approximation of the curvature when no smoothing is applied (or away from the smoothing region) and a first order convergence near the free surface when smoothing is employed. The results of our numerical experiments clearly show the anticipated convergence behavior. Furthermore, we compared our results to experimental data and observed a very good agreement.

Finally, we have considered the dynamic contact angle problem for the test case of a sliding drop on an inclined surface. Here, we clearly observed the expected qualitative and physical behavior of the moving contact line. In fact, the angle increases with increasing velocity of the moving contact line. Hence, a direct numerical simulation of the dynamic contact angle problem seems feasible if surface tension effects are taken into account.

## REFERENCES

[1] D. ADALSTEINSSON AND J. A. SETHIAN, *A fast levelset method for propagating interfaces*, J.

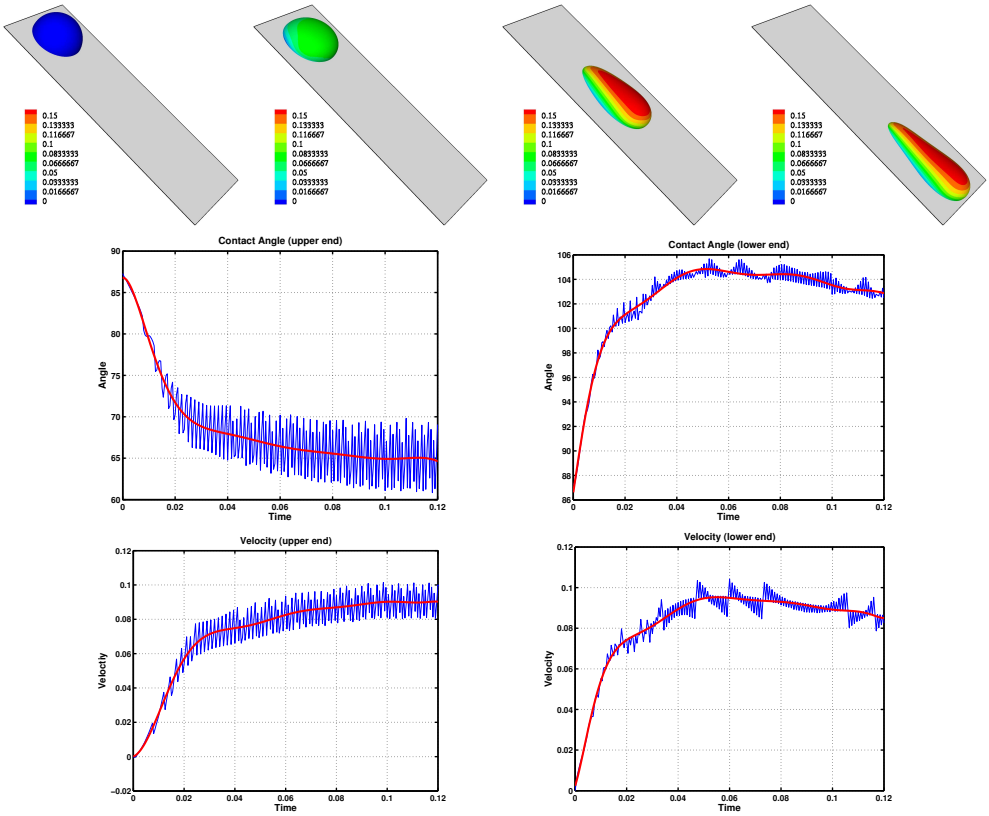


FIGURE 5.12. *Sliding drop for g=9.81* with *surface tension (upper row). The color-contour shows the velocity in x-direction. The dynamic contact angle (center row) at the upper and lower edge of the drop and the respective velocities in x-direction (lower row).*

Comput. Phys., 118 (1995), pp. 269–277.

[2] A. AMSDEN AND F. HARLOW, *A simplified MAC technique for incompressible fluid flow calculations*, J. Comput. Phys., 6 (1970), p. 322.

[3] E. BÄNSCH, *Finite element discretization of the Navier–Stokes equations with a free capillary surface*, preprint, Universität Bremen, 1999.

[4] T. D. BLAKE AND M. J. HAYNES, *Kinetics of liquid/liquid displacement*, J. Colloid Interface Sci., 30 (1969), p. 421.

[5] J. BRACKBILL, D. KOTHE, AND C. ZEMACH, *A continuum method for modeling surface tension*, J. Comput. Phys., 100 (1992), p. 335.

[6] Y. CHANG, T. HOU, B. MERRIMAN, AND S. OSHER, *A level set formulation of Eulerian interface capturing methods for incompressible fluid flows*, J. Comput. Phys., 124 (1996), p. 449.

[7] S. CHEN, D. JOHNSON, P. RAAD, AND D. FADDA, *The surface marker and micro cell method*, International Journal for Numerical Methods in Fluids, (1997).

[8] S. CHEN, B. MERRIMAN, S. OSHER, AND P. SMEREKA, *A simple level set method for solving Stefan problems*, J. Comput. Phys., 135 (1997), pp. 8–29.

[9] D. L. CHOPP, *Computing minimal surfaces via level set curvature flow*, J. Comput. Phys., 106 (1993), pp. 77 – 91.

[10] A. CHORIN, *Numerical solution of the Navier–Stokes equations*, Math. Comput., 22 (1968), p. 745.

[11] B. DALY, *Numerical study of two fluid Rayleigh–Taylor instability*, The Physics of Fluids, 10 (1967), p. 297.

[12] ———, *Numerical study of the effect of surface tension on interface instability*, The Physics of Fluids, 12 (1969), p. 1340.

[13] R. DEBAR, *Fundamentals of the KRAKEN code*, report ucir-760, Lawrence Livermore National

Laboratory, 1974.
[14] J. E. Dendy, *Black box multigrid*, J. Comput. Phys., 48 (1982), pp. 366–386.
[15] G. Dziuk, *Partial Differential Equations and Calculus of Variations*, vol. 1357, Springer, 1988, ch. Finite Elements for the Beltrami Operator on Arbitrary Surfaces, pp. 142–155.
[16] ———, *An algorithm for evolutionary surfaces*, Numer. Math., 58 (1991), pp. 603–611.
[17] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, *A hybrid particle level set method for improved interface capturing*, J. Comput. Phys., 183 (2002), pp. 83–116.
[18] M. Fermigier and P. Jenffer, *Experimental investigation of the dynamic contact angle in liquid-liquid systems*, J. Colloid Interface Sci., 146 (1991), pp. 226–241.
[19] I. Ginzburg and G. Wittum, *Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants*, J. Comput. Phys., (2001).
[20] I. Ginzburg, G. Wittum, and S. Zaleski, *Adaptive multigrid computations of multiphase flows*, in Numerical Flow Simulation III, E. Hirschel, ed., vol. 82 of Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer, 2002, pp. 77–96.
[21] M. Griebel, T. Dornseifer, and T. Neunhoeffer, *Numerical Simulation in Fluid Dynamics, A Practical Introduction*, SIAM, Philadelphia, 1998.
[22] M. Griebel, T. Neunhoeffer, and H. Regler, *Algebraic multigrid methods for the solution of the Navier–Stokes equations in complicated domains*, Int. J. Numer. Methods for Heat and Fluid Flow, 26 (1998), pp. 281–301.
[23] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI—Portable Parallel Programming with the Message-Passing Interface*, The MIT Press, 2nd ed., 1999.
[24] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski, *Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows*, J. Comput. Phys., 152 (1999), pp. 423–456.
[25] F. Harlow and J. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, The Physics of Fluids, 8 (1965), p. 2182.
[26] C. Hirt and B. Nichols, *Volume of fluid VOF method for the dynamics of free boundaries*, J. Comput. Phys., (1981), p. 201.
[27] L. Ho and A. Patera, *A Legendre spectral element method for simulation of unsteady incompressible viscous free-surface flows*, Comput. Methods Appl. Mech. Eng., (1990), pp. 355–366.
[28] L. Ho and E. Rønquist, *Spectral element solution of steady incompressible viscous free-surface flows*, Finite Elements in Analysis and Design, 16 (1994), pp. 207–227.
[29] G. Jiang and D. Peng, *Weighted ENO schemes for Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 21 (2000), pp. 2126–2143.
[30] G. Jiang and C. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), p. 202.
[31] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, *Modelling merging and fragmentation in multiphase flows with SURFER*, J. Comput. Phys., 113 (1994), p. 134.
[32] L. Landau and E. Lifshitz, *Fluid Mechanics*, Pergamon Press, Oxford/London/Paris/Frankfurt, 1959.
[33] J. Martin and W. Moyce, *An experimental study of the collapse of liquid columns on a rigid horizontal plane*, Philos. Trans. R. Soc. Lond. Ser. A, 244 (1952), pp. 312–324.
[34] B. Merriman, J. Bence, and S. Osher, *Motion of multiple junctions: A level set approach*, J. Comput. Phys., 122 (1994), pp. 334–363.
[35] B. Metsch, *Ein paralleles graphenbasiertes algebraisches Mehrgitterverfahren*, Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2004.
[36] W. Mulder, S. Osher, and J. A. Sethian, *Computing interface motion in compressible gas dynamics*, J. Comput. Phys., 100 (1992), pp. 209–228.
[37] T. Neunhoeffer, *Numerische Simulation von Erstarrungsprozessen unterkühlter Flüssigkeiten unter Berücksichtigung von Dichteunterschieden*, Dissertation, Fakultät für Informatik der Technischen Universität München, 1997.
[38] W. Noh and P. Woodward, *SLIC–Simple line interface calculation*, in Fifth International Conference on Fluid Dynamics, Lecture Notes in Physics, A. V. Vooren and P. Zandbergen, eds., vol. 59, Springer, 1976, p. 330.
[39] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, no. 153 in Applied Mathematical Sciences, Springer, New York/Berlin/Heidelberg, 2003.
[40] S. Osher and J. Sethian, *Fronts propagating with curvature–dependent speed: Algorithms based on Hamilton–Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49.
[41] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, *A PDE-based fast local level set method*, J. Comput. Phys., 155 (1999), pp. 410–438.

[42] G. Russo and P. Smereka, *A remark on computing distance functions*, J. Comput. Phys., 163 (2003), pp. 51–67.
[43] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 1999.
[44] C. Shu and S. Osher, *Efficient implementation of essentially non–oscillatory shock–capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
[45] M. Sussman and E. Fatemi, *An efficient, interface preserving level set re-distancing algorithm and its application to interfacial incompressible fluid flow*, SIAM J. Sci. Comput., 20 (1999), p. 1165.
[46] M. Sussman and G. Puckett, *A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows*, J. Comput. Phys., 162 (2000), pp. 301–337.
[47] M. Sussman, P. Smereka, and S. Osher, *A level set approach for computing solutions to incompressible two–phase flow*, J. Comput. Phys., 114 (1994), pp. 146–159.
[48] M. Sussman and M. Y.Hussaini, *A discontinuous spectral element method for the level set equation*, J. Sci. Comput., 19 (2003), pp. 479–500.
[49] U. Trottenberg, C. W. Osterlee, and A. Schüller, *Multigrid*, Academic Press, San Diego, 2001, Appendix A: An Introduction to Algebraic Multigrid by K. Stüben, pp. 413–532.
[50] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y. Jan, *A front tracking method for the computations of multiphase flow*, J. Comput. Phys., 169 (2001), p. 708.
[51] S. Unverdi and G. Tryggvason, *A front-tracking method for viscous, incompressible multi-fluids flow*, J. Comput. Phys., 100 (1992), pp. 25–37.
[52] P. M. Zeeuw, *Matrix-dependent prolongations and restrictions in a black-box multigrid solver*, J. Comput. Appl. Math., 33 (1990), pp. 1–27.