DELFT UNIVERSITY OF TECHNOLOGY

REPORT 03-03

A Mass-Conserving Level-Set (MCLS) Method for Modeling of Multi-Phase Flows

S.P. van der Pijl, A. Segal, C. Vuik

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2003

A Mass-Conserving Level-Set (MCLS) Method for Modeling of Multi-Phase Flows

S.P. van der Pijl* A. Segal C. Vuik

February 25, 2003

Abstract

The Mass-Conserving Level-Set (MCLS) method is proposed to model multi-phase flows. The aim is to model high density-ratio flows with complex interface topologies, such as mixtures of bubbles and droplets. Aspects which are taken into account are: a sharp front (density changes rapidly), arbitrary shaped interfaces, surface tension, buoyancy and coalescence of drops/bubbles. Attention is paid to mass-conservation and integrity of the interface.

A survey of available computational methods is performed in [1]. The proposed computational method is a combination of Level-Set and Volume-of-Fluid methods. The flow is computed with a pressure correction method with a Marker-and-Cell layout. Interface conditions are satisfied by means of the continuous surface force/stress (CSF/CSS) methodology and the GhostFluid method for incompressible flows.

The Level-Set method is an elegant method. The major disadvantage is that it is not rigorously mass-conserving. This means that additional effort is necessary to conserve mass. The MCLS method introduces a Volume-of-Fluid function, which is advected without the necessity to reconstruct the interface. There is a strong relationship between the Volume-of-Fluid function and the Level-Set function. In the spirit of the Level-Set methodology, the advection of the VOF-function is, unlike other VOF methods, purely implicit at every time. This makes the method straightforward to apply to arbitrarily shaped interfaces, which may collide and break up.

1 Introduction

Multi-phase flows occur commonly in engineering fluid mechanics. In this work modeling of incompressible two-phase flows is considered. The present research aims to model high density-ratio flows with complex interface topologies, typically air/water flows. A sharp front exists between the phases, where fluid

*Research granted by NWO.

density and viscosity change rapidly. This front is modeled as an interface between the phases where fluid properties are discontinuous. The interface is a moving (internal) boundary. Also, surface tension forces act on the interface. A numerical method has to take this into account as well as to be able to handle arbitrarily shaped interfaces which may collide and break up.

Various methods have been put forward to treat multi-phase flows. A classification is given in [1]. Most generally speaking, the interface representation can be explicit ('moving, boundary conforming mesh') or implicit ('fixed mesh') or a combination of both. Purely moving, boundary-conforming meshes will be cumbersome, since the objective is to simulate arbitrarily shaped interfaces. It is therefore not very suited to apply to this work.

A combination of fixed and moving mesh methods include the *hybrid front-tracking/front-capturing* method ([2–6]) and the closely related *immersed boundary* method ([7–10]). Although the interface is tracked by an interface grid, the flow is solved on a stationary grid. The interface conditions are satisfied by regularizing (smoothing) the interface discontinuities and interpolating interface forces from the interface grid to the stationary grid. For this purpose, the interface forces are transformed into volume forces and distributed over certain mesh widths. This is sometimes referred to as the *continuous surface force* (CSF) or *continuous surface stress* (CSS) approach ([11]). As far as solving the flow field is concerned, the interface position is not prescribed explicitly. The interface grid on the other hand is advected by interpolating the velocity field from the stationary grid to the interface grid. A drawback of the method might be that the interface grid can get hard to evaluate, especially when interfaces collide or break up. Furthermore, interpolating from the interface grid to the stationary grid is carried out by smearing of interface forces and discontinuities over certain mesh widths. This can introduce so-called parasitic currents ([6]) which are unphysical regions of recirculated fluid near the interface. In the *cut-cell* approach ([12–17]) on the other hand, the interface conditions are satisfied without smoothing of the interface. The control volumes of the stationary grid are readjusted near the interface, so that it is locally boundary conforming. It has been successfully applied to fluid-solid interfaces, where a velocity is prescribed on the interface. However, in case of two-phase flows, the interface separates two fluids and the interface conditions express continuity of mass and momentum over the interface. The cut-cell approach is extended to handle fluid-fluid interfaces in [18] by iteratively modifying the interface shape, such that the interface conditions are satisfied. Since control volumes have to be readjusted, the cut-cell method requires a large amount of logic decisions. At first sight the method seems to be cumbersome to apply in three-dimensional space. Also, the iterative procedure is not trivial.

The combined implicit/explicit methods are sometimes referred to as 'front'- or 'surface tracking' methods. The interface grid which is needed to track the moving boundary will be difficult to evaluate when the interface has arbitrary shape and topology. Instead, for the current research the implicit methods have been selected to represent the interface. These methods are also called 'volume tracking' methods. The interface is implicitly defined by marking it by particles

or by a function.

With the *Marker-and-Cell* (MAC) method ([19]), one of the two phases is indicated by marker particles. With the *Surface Marker and Micro Cell* (SMMC) method ([20]) the interface itself is marked by particles. The particles are advected by the flow field. For both methods it is hard to compute interface quantities like the normal vector and curvature in order to model surface tension forces, since the connectivity of particles is unknown (otherwise these would be front-tracking methods).

The two volume-tracking methods that are most suitable for the current research are the *Volume-of-Fluid* (VOF) method and the *Level-Set* method. For both methods a marker function is used to define the interface. In case of the Volume-of-Fluid method, a marker function, say Ψ, indicates the fractional volume of a certain fluid, say fluid '1', in a computational cell. It can be seen as the concentration of the marker particles of the MAC-method, when the number of particles goes to infinity. Hence Ψ is defined in a grid cell Ω by

$$\Psi = \frac{1}{\mathrm{vol}(\Omega)} \int_{\Omega} \chi \, \mathrm{d}\Omega, \tag{1}$$

where χ is the characteristic function

$$\chi = \begin{cases} 1 & \text{in fluid '1',} \\ 0 & \text{outside fluid '1'.} \end{cases} \tag{2}$$

The value of Ψ will be 0 or 1 in cells without the interface and $0 \leq \Psi \leq 1$ in cells containing the interface. In other words, the value of Ψ changes rapidly across the interface. This causes difficulties in advecting Ψ. The step-like behavior of Ψ will be lost by the straightforward application of a numerical scheme. Hence the interface will get ill defined. Volume-of-fluid methods can suffer from 'flotsam and jetsam', which are the small remnants of mixed-fluid zones ([21]). Besides that, the step-like behavior of the marker function makes computing interface derivatives (normals, curvature) elaborate. The major advantage of VOF methods is that mass is rigorously conserved, provided the discretization is conservative.

Advecting Ψ is not a straightforward task. This can either be performed algebraically or geometrically. Algebraic methods try to discretize the advection equation for the marker function Ψ by incorporating the step-like behavior of Ψ. Examples of such approaches are: *Constrained Interpolation Profile* ([22–24]) and *Flux-Corrected Transport* (FCT) (([25, 26]). Geometric methods on the other hand first reconstruct the interface from Ψ, after which it is advected. Geometric VOF methods differ in the accuracy of the interface reconstruction. With *Simple Line Calculation* (SLIC) the interface is assumed to be piecewise tangential to a coordinate direction ([21]). In case of the *donor-acceptor* (SOLA-VOF) method the interface is stair-stepped ([27, 28]). And with the Piecewise Linear Interface Calculation (PLIC) method the interface is piecewise linear ([26, 29–33]). Reconstruction of the interface will be increasingly difficult if the

discrete interface representation is more accurate. This is a drawback especially of the PLIC method, which on the other hand is the most accurate method.

An alternative to the Volume-of-Fluid methods is the Level-Set method ([34–37]). The interface is now defined by the zero level-set of the marker function, say Φ:

$$\Phi \begin{cases} < 0 & \text{outside fluid '1'}, \\ = 0 & \text{at the interface}, \\ > 0 & \text{inside fluid '1'}. \end{cases} \tag{3}$$

The function Φ is chosen such that it is smooth near the interface. This eases the computation of interface derivatives. Also, methods available from hyperbolic conservation laws can be used to advect the interface. The interface is (implicitly) advected by advecting Φ as if it were a material constant:

$$\frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \nabla \Phi = 0. \tag{4}$$

Although this makes the Level-Set method an elegant method, the major disadvantage is that it is not rigorously mass-conserving. This means that additional effort is necessary to conserve mass, or at least to improve mass conservation. One approach would be to approximate Eqn. (4) more accurately by higher order schemes or by grid refinement. In [38–40] higher order ENO discretization of Eqn. (4) is adopted and in [41–43] the grid is refined adaptively near the interfaces.

The interface forces and discontinuities are often smeared out over certain mesh widths in a CSS/CSF way. To keep the smeared-out interface thickness constant in the course of time, it is necessary that the Level-Set function is a so-called distance function at all time instants. This is achieved by re-initialization in [44]. However, when the interface is advected through the flow field, the violation of mass-conservation is increased due to re-initialization. Therefore, improved re-initialization is carried out in [38–40].

Mass-conservation is improved due to all aforementioned efforts, but never exactly satisfied. In [45] the mass-conservation properery of the Level-Set method is improved by adding passively advected marker particles. These particles are used near the interface. It can be seen as a coupling of the Level-Set and Marker-and-Cell methods. It is required to keep track of the particles and to redistribute the particles, which increases the amount of bookkeeping. The velocity field has to be interpolated at the particle positions. Furthermore, the interface has to be reconstructed from the particles. This might not be a straightforward task.

The *Coupled Level-Set Volume-of-Fluid* (CLSVOF) method ([46]) is a coupling of the Level-Set method with the Volume-of-Fluid PLIC method. Besides the advection of the Level-Set function Φ also the Volume-of-Fluid function Ψ is advected. However, there is no straightforward relationship between the Level-Set function and the Volume-of-Fluid function; both advections are independent. After each update of Φ and Ψ, coupling of both functions takes place. This coupling is not easily achieved. Since the PLIC approach is employed, a

drawback of this method might be that besides mass-conservation, also the elaborateness of the VOF methods is imported. The mass-conservation properties are shown to be comparable to VOF methods.

This work also combines the Level-Set method with the Volume-of-Fluid method. A VOF-function Ψ is introduced, which is advected without the necessity to reconstruct the interface. There exists a strong relationship between the Volume-of-Fluid function and the Level-Set function Φ. In the spirit of the Level-Set methodology, the advection of the VOF-function is, unlike PLIC and consequently CLSVOF, purely implicit at all time. This makes the method straightforward to apply to arbitrarily shaped interfaces, which may collide and break up.

2 Governing Equations

Consider two fluids '0', and '1' in domain $\Omega \in \mathbb{R}^2$ which are separated by an interface S. Both fluids are assumed to be incompressible, i.e.:

$$\nabla \cdot \mathbf{u} = 0, \tag{5}$$

where $\mathbf{u} = (u, v)^t$ is the velocity vector and u and v are the velocities in x- and y-direction respectively. The flow is governed by the incompressible Navier-Stokes equations:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^t\right) + \rho \mathbf{g}, \tag{6}$$

where ρ, p, μ and $\mathbf{g}$ are the density, pressure, viscosity and gravity vector respectively. With Eqn. (5), this is rewritten as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^t\right) + \mathbf{g}, \tag{7}$$

The density and viscosity are constant within each fluid. Using the Level-Set function Φ as defined in Eqn. (3), these can be expressed as

$$\rho = \rho_0 + (\rho_1 - \rho_0) H(\Phi) \tag{8}$$

and

$$\mu = \mu_0 + (\mu_1 - \mu_0) H(\Phi), \tag{9}$$

where the subscript indicates the corresponding fluid and H is the Heaviside step function.

2.1 Interface conditions

The interface conditions express continuity of mass and momentum at the interface:

$$\begin{aligned} [\mathbf{u}] &= 0 \\ [p\mathbf{n} - \mathbf{n} \cdot \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^t\right)] &= \sigma \kappa \mathbf{n}, \end{aligned} \tag{10}$$

where the brackets denote jumps across the interface, $\mathbf{n}$ is a normal vector at the interface, σ is the surface tension coefficient and κ is the curvature of the interface. Clearly, the velocity $\mathbf{u}$ is continuous at the interface. Concerning the derivatives of the velocity, if $\mathbf{s}$ is parallel to the interface, $u_n = \mathbf{n} \cdot \mathbf{u}$ and $u_s = \mathbf{s} \cdot \mathbf{u}$, in [47, 48] it is shown that in general

$$\begin{aligned} \left[\frac{\partial u_n}{\partial n}\right] &= 0, & \left[\frac{\partial u_n}{\partial s}\right] &= 0 \\ \left[\frac{\partial u_s}{\partial n}\right] &= -[\mu]\frac{\partial u_n}{\partial s}, & \left[\frac{\partial u_s}{\partial s}\right] &= 0. \end{aligned} \tag{11}$$

Note that if the viscosity μ is continuous at the interface, Eqn. (11) shows that the derivatives of the velocity components are continuous too. The viscosity is forced to be continuous by smoothing Expression 9:

$$\mu = \mu_0 + (\mu_1 - \mu_0) H_\alpha(\Phi), \tag{12}$$

where H_α is the smoothed (or regularized) Heaviside step function

$$H_\alpha(x) = \begin{cases} 0 & x < -\alpha \\ \frac{1}{2}\left(1 + \sin(\frac{x}{\alpha}\frac{1}{2}\pi)\right) & |x| \leq \alpha \\ 1 & x > \alpha \end{cases} \tag{13}$$

and α is a parameter proportional to the mesh width. Here α is chosen as (following [44]) $\alpha = \frac{3}{2}h$, where h is the mesh width. According to [38], the viscosity is then smoothed over three mesh widths, provided $|\nabla\Phi| = 1$. Furthermore, if the Level-Set function Φ is a distance function, i.e. $|\nabla\Phi| = 1$, then

$$\mu = \mu_0 + (\mu_1 - \mu_0) H_\alpha(\Phi). \tag{14}$$

Note that only the viscosity is smoothed, not the density ρ. Instead of taking into account the pressure-jump at the interface due to the surface tension forces, the continuous surface force/stress (CSF,CSS, [11]) methodology is adopted.

3 Computational Approach

The Navier-Stokes equations are solved by the pressure-correction method ([49]). The unknowns are stored in a Marker-and-Cell layout ([19]). For the interface representation a combination of the Level-Set and Volume-of-Fluid methodologies are adopted. The interface conditions are satisfied by means of the continuous surface force/stress (CSF/CSS) methodology and the GhostFluid method for incompressible flow ([47, 50]).

3.1 Pressure Correction

The Navier-Stokes equations (Eqn. (7)) are discretized in a finite difference fashion by employing the pressure correction methodology. The unknowns u, v

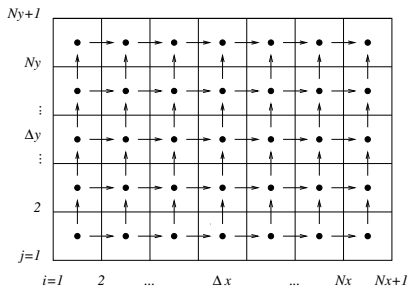

Figure 1: Locations of u (right-arrow), v (uparrow) and p (dot)

and p are located as indicated in Fig. 1. If superscript ^{n} denotes time-level n, first a tentative velocities u^* is computed by

$$\begin{aligned} \frac{u^*-u^n}{\Delta t} &= -u^n\left(\frac{\partial u}{\partial x}\right)^* - v^n\left(\frac{\partial u}{\partial y}\right)^* + \\ &\quad \frac{1}{\rho}\left(\left(\frac{\partial \mu \frac{\partial u}{\partial x}}{\partial x} + \frac{\partial \mu \frac{\partial u}{\partial y}}{\partial y}\right)^* + \left(\frac{\partial \mu \frac{\partial u}{\partial x}}{\partial x} + \frac{\partial \mu \frac{\partial v}{\partial x}}{\partial y}\right)^n\right). \end{aligned} \tag{15}$$

Note that there are no pressure gradient terms at the previous time level included in Eqn. (15). These are absent, because the density ρ^{n+1} is not equal to the density ρ^n due to the moving interface, so that it is impossible to write

$$\frac{1}{\rho}\nabla p^{n+1} - \frac{1}{\rho}\nabla p^n = \frac{1}{\rho}\nabla \delta p, \tag{16}$$

which is the usual formulation for single-phase flows if $\delta p = p^{n+1} - p^n$. Furthermore, gravity terms are only included if the pressure gradient is present. Since the pressure gradient is absent in Eqn. (15) no gravity is included there. The equation for v is treated similarly:

$$\begin{aligned} \frac{v^*-v^n}{\Delta t} &= -u^n\left(\frac{\partial v}{\partial x}\right)^* - v^n\left(\frac{\partial v}{\partial y}\right)^* + \\ &\quad \frac{1}{\rho}\left(\left(\frac{\partial \mu \frac{\partial v}{\partial x}}{\partial x} + \frac{\partial \mu \frac{\partial v}{\partial y}}{\partial y}\right)^* + \left(\frac{\partial \mu \frac{\partial u}{\partial y}}{\partial x} + \frac{\partial \mu \frac{\partial v}{\partial y}}{\partial y}\right)^n\right). \end{aligned} \tag{17}$$

Due to the regularization of μ, the velocities u and v and their derivatives are continuous. The gradients can therefore be approximated by central differences. Velocities which have to be evaluated at different locations then where they are stored are approximated by averaging ([19]). Note that the stress tensor is split in a part on time level * (implicit) and ^{n} (explicit). The reason is that in regions where μ is constant (away from the interface) the explicit part will cancel since the continuity equation (Eqn. (5)) yields:

$$\frac{\partial \frac{\partial u}{\partial x}}{\partial x} + \frac{\partial \frac{\partial v}{\partial x}}{\partial y} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0 \tag{18}$$

and

$$\frac{\partial \frac{\partial u}{\partial y}}{\partial x} + \frac{\partial \frac{\partial v}{\partial y}}{\partial y} = \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0. \tag{19}$$

The velocities at the new time instant $^{n+1}$ are computed by:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\nabla p + \mathbf{g}, \tag{20}$$

under the constraint of Eqn. (5)

$$\nabla \cdot \mathbf{u}^{n+1} = 0. \tag{21}$$

These are symbolically discretized as

$$\left\{ \begin{array}{rcl} \mathbf{u}^{n+1} & = & \mathbf{u}^* + \Delta t \left(-\frac{1}{\rho} G p + \mathbf{g}\right) \\ D\mathbf{u}^{n+1} & = & 0, \end{array} \right. \tag{22}$$

where D represents the discretization of the divergence and G is the discrete gradient operator which remains to be specified. Since the pressure p is discontinuous at the interface (Eqn. (10)), special care has to be taken with the gradients Gp. These are computed by the GhostFluid method for incompressible flows ([47, 50, 51]) which is discussed in the next section. From the last expressions follows for p:

$$D\frac{1}{\rho}Gp = D\left(\frac{1}{\Delta t}\mathbf{u}^* + \mathbf{g}\right). \tag{23}$$

Due to our approach, no boundary conditions are needed for the pressure when velocity boundary conditions are applied.

3.1.1 GhostFluid method

The interface conditions have to be satisfied by prescribing the interface discontinuities at the interface. Due to the regularization of μ, only pressure jumps have to be taken into account. This jump is according to Eqn. (10):

$$[p] = \sigma\kappa|_{\Phi=0}\,, \tag{24}$$

where σ is the surface tension coefficient and κ is the curvature of the interface, which can be expressed as (e.g. [39])

$$\kappa = \nabla \cdot \frac{\nabla\Phi}{|\nabla\Phi|}. \tag{25}$$

The curvature is approximated by central differences. So for a given interface position, the jump $[p]$ can be computed. The GhostFluid method for incompressible flows serves as a straightforward extension of a finite difference formulation. It is based on an interface representation by means of the Level-Set

function Φ. The discontinuities are computed at Φ locations and interpolated at the interface. Note that the *Immersed Interface Method* ([48, 52–55]) method is a similar method, but specific to front tracking, where interface discontinuities are known on the interface (grid) itself.

In [51], the pressure derivatives $\frac{1}{\rho}\frac{\partial p}{\partial x}$ at u locations $(i+1, j+\frac{1}{2})$ are approximated by

$$\left(\beta\frac{\partial p}{\partial x}\right)_{i+1,j+\frac{1}{2}} = \hat{\beta}_{i+1,j+\frac{1}{2}}\frac{p_{i+1\frac{1}{2},j+\frac{1}{2}} - p_{i+\frac{1}{2},j+\frac{1}{2}} - [p]_{i+1,j+\frac{1}{2}}}{\Delta x} \tag{26}$$

and similar for $\beta\frac{\partial p}{\partial y}$, where $\beta = \frac{1}{\rho}$. Here it is used that the jump of the pressure-gradients $[\frac{1}{\rho}\nabla p] = \mathbf{0}$ ([47, 50]). The quantity $\hat{\beta}$ is the harmonic weighted average of β.

In this paper the GhostFluid method is applied with the discontinuous density. However, the Continuous Surface Force/Stress (CSF/CSS) methodology is adopted to take into account the surface tension forces. Hence the pressure jump at the interface reduces to

$$[p] = 0. \tag{27}$$

3.1.2 Continuous Surface Force/Stress (CSF/CSS)

From the GhostFluid methodology follows that there exist surface tension forces of the form (Eqn. (26))

$$-\hat{\beta}\sigma\kappa\mathbf{n}, \tag{28}$$

where $\mathbf{n}$ is the normal vector at the interface, $\beta = \frac{1}{\rho}$ and $\hat{\beta}$ is the harmonic average, so that

$$\hat{\beta} = \frac{1}{\frac{1}{2}\frac{1}{\beta_0} + \frac{1}{2}\frac{1}{\beta_1}} = \frac{1}{\frac{1}{2}(\rho_0 + \rho_1)} = \frac{1}{\bar{\rho}}. \tag{29}$$

By $\bar{\rho}$ is meant the average density, which is constant in the whole domain Ω. Adopting the CSS/CSF methodology, the interface forces acting on S are transformed into volume forces acting in Ω by writing (e.g. [39])

$$\int_S \sigma\kappa\mathbf{n}\,\mathrm{d}S = \int_\Omega \sigma\kappa\delta(\Phi)\nabla\Phi\,\mathrm{d}\Omega, \tag{30}$$

where δ is the delta function. This results in additional terms in Eqn. (7) of the form

$$-\frac{1}{\bar{\rho}}\sigma\kappa\delta(\Phi)\nabla\Phi. \tag{31}$$

Note that the force $\sigma\kappa\mathbf{n}$ is conserved, since for the force in x-direction for example (assume a monotone function $\Phi(x)$, $x \in \mathbb{R}$)

$$\begin{aligned}
\int_{-\infty}^{\infty} \rho(\Phi)\frac{\sigma\kappa(\Phi)\delta(\Phi)\nabla\Phi}{\bar{\rho}} \cdot \mathbf{e}_x \,\mathrm{d}x &= \int_{-\infty}^{\infty} \rho(\Phi)\frac{\sigma\kappa(\Phi)\delta(\Phi)}{\bar{\rho}}\frac{\partial\Phi}{\partial x}\,\mathrm{d}x \\
&= \int_{-\infty}^{0} \frac{\rho_0}{\bar{\rho}}\sigma\kappa(\Phi)\delta(\Phi)\,\mathrm{d}\Phi \\
&\quad + \int_{0}^{\infty} \frac{\rho_1}{\bar{\rho}}\sigma\kappa(\Phi)\delta(\Phi)\,\mathrm{d}\Phi \\
&= \frac{\frac{1}{2}\rho_0+\frac{1}{2}\rho_1}{\bar{\rho}}\sigma\kappa(0) \quad = \quad \sigma\kappa(0).
\end{aligned} \tag{32}$$

By regularizing the delta function in the same manner is the Heaviside step function (Eqn. (13))

$$\delta_\alpha(\Phi) = \begin{cases} \frac{1}{2\alpha}\left(1+\cos(\frac{\Phi}{\alpha}\pi)\right) & |\Phi| \leq \alpha \\ 0 & |\Phi| > \alpha \end{cases} \tag{33}$$

the pressure jump is reduced to $[p] = 0$. Consequently, Eqn. (7) becomes

$$\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla\cdot\mu\left(\nabla\mathbf{u}+\nabla\mathbf{u}^t\right) + \mathbf{g} - \frac{1}{\bar{\rho}}\sigma\kappa\delta_\alpha(\Phi)\nabla\Phi. \tag{34}$$

Here α has the same value as in Eqn. (13), i.e. $\alpha = \frac{3}{2}h$. The additional term $\frac{\sigma\kappa\delta_\alpha(\Phi)\nabla\Phi}{\bar{\rho}}$ is a force just like the gravity vector $\mathbf{g}$ and appears, like $\mathbf{g}$, in Eqns. (20) and (22) in a straightforward discrete sense:

$$D\frac{1}{\rho}Gp = D\left(\frac{1}{\Delta t}\mathbf{u}^* + \mathbf{g} - \frac{1}{\bar{\rho}}\sigma\kappa\delta_\alpha(\Phi)\nabla\Phi\right) \tag{35}$$

and

$$\mathbf{u}^{n+1} = \mathbf{u}^* + \Delta t\left(-\frac{1}{\rho}Gp + \mathbf{g} - \frac{1}{\bar{\rho}}\sigma\kappa\delta_\alpha(\Phi)\nabla\Phi\right). \tag{36}$$

If the interface forces were *not* regularized, but included as a pressure jump in Eqn. (26), the pressure-jump terms would enter the pressure-correction step in a similar manner. Note also that $\frac{\sigma\kappa}{\rho}$ is regularized and not $\sigma\kappa$. This keeps the density ρ (piecewise) constant, which guarantees a straightforward application of the pressure-correction methodology. The GhostFluid method, described previously is used to deal with the discontinuous density.

3.2 Interface advection

The strategy of modeling two-phase flows is to compute the flow with a given interface position and subsequently evolve the interface in the given flow field. In the foregoing has been described how the flow is computed with a given interface position. The evolution of the interface is considered in the following.

3.2.1 Level-Set

The interface is implicitly defined by a Level-Set function Φ. More precisely, the interface, say S, is the zero level-set of Φ:

$$S(t) = \left\{ \mathbf{x} \in \mathbb{R}^2 | \Phi(\mathbf{x}, t) = 0 \right\}. \tag{37}$$

The interface is evolved by advecting the Level-Set function in the flow field as if it were a material constant (Eqn. (4)):

$$\frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \nabla \Phi = 0. \tag{38}$$

A symmetry condition for Φ is imposed at the boundaries. It will be clear that accuracy of the approximation of Eqn. (38) determines the accuracy of the interface representation. But, since mass is not rigorously conserved, the accuracy will also determine the mass errors. For this purpose, the discretization of the gradient of Φ can be either first order upwind, or second or third order ENO ([38, 39, 44]). In case of the first-order spatial discretization, a forward Euler temporal discretization is sufficient. In case of the higher order spatial discretization, a Runge-Kutta scheme is implemented (e.g. [40]).

Re-initialization If an initial signed distance function is advected through a non-uniform flow, it does not necessarily correspond to a distance function any longer. For a distance function holds

$$|\nabla \Phi| = 1. \tag{39}$$

If the interface is smoothed over certain mesh widths, keeping Φ a distance function ensures that the front has finite thickness at all time ([38, 44]). This is especially important when the surface tension forces are distributed over a number of grid cells (CSS/CSF approach). Also, in the regularization of μ (Eqn. (14)) it is used that $|\nabla \Phi| = 1$.

Function Φ is reinitialized each time step by ([38, 44]):

$$\begin{array}{rcl} \frac{\partial \Phi}{\partial t'} & = & \mathrm{sign}(\Phi|_{t'=0}) \left(1 - \sqrt{\frac{\partial \Phi}{\partial x_j} \frac{\partial \Phi}{\partial x_j}}\right) \\ \Phi(\mathbf{x}, 0) & = & \Phi|_{t'=0}(\mathbf{x}), \end{array} \tag{40}$$

where t' is an artificial time and the initial condition $\Phi|_{t'=0}$ is the updated Level-Set function by advection. For the limit $t' \to \infty$, the reinitialized Φ satisfies $|\nabla \Phi| = 1$, so it is a distance function. Note that due to numerical diffusion the re-initialization procedure can increase the smoothness of Φ. Consequently, the zero level-set might shift and mass errors occur. Therefore in [39, 40, 43] the re-initialization is improved, such that mass errors due to re-initialization are negligible.

Mass-errors due to the finite-accurate approximation of the Level-Set advection can never be circumvented by improving re-initialization. Therefore a different approach is chosen, which is described in the next section.

3.2.2 MCLS

In order to conserve mass, the Level-Set method as described above is combined with a Volume-of-Fluid method. In that sense, first the usual Level-Set advection is performed: first-order advection and unmodified re-initialization. Low order advection and re-initialization will ensure numerical smoothness of Φ. The obtained Level-Set function $\Phi^{n+1,*}$ will certainly not conserve mass. Therefore, corrections to $\Phi^{n+1,*}$ are made such that mass is conserved. This requires three steps:

1. the relative volume of a certain fluid in a computational cell (called 'volume-of-fluid' function Ψ) is to be computed from the Level-Set function Φ^n;
2. the volume-of-fluid function has to be advected during a time step towards Ψ^{n+1};
3. with this new volume-of-fluid function Ψ^{n+1}, corrections to $\Phi^{n+1,*}$ are sought such that $\Psi(\Phi^{n+1}) = \Psi^{n+1}$ holds.

These three steps will be explained subsequently.

Step 1: Volume-of-Fluid function A relationship between the Level-Set function Φ and the so-called volume-of-fluid function Ψ is found by considering the fractional volume of a certain fluid in a computational cell Ω_k. In this paper, a Cartesian mesh is employed consisting of computational cells $\Omega_k, k = 1, 2, \ldots$. By $\mathbf{x}_k = (x_k, y_k)^t$ the center node of Ω_k is meant and Δx and Δy are the mesh sizes in x and y direction respectively. In computational cell Ω_k, the volume-of-fluid function Ψ_k is by definition (Eqn. (1))

$$\Psi_k = \frac{1}{\mathrm{vol}(\Omega_k)} \int_{\Omega_k} \chi \, \mathrm{d}\Omega. \tag{41}$$

Employing the Level-Set function Φ of Eqn. (3), the characteristic function χ (Eqn. (2)) becomes:

$$\chi = H(\Phi), \tag{42}$$

where H is the Heaviside step function. The connection between Φ and Ψ is therefore:

$$\Psi_k = \frac{1}{\mathrm{vol}(\Omega_k)} \int_{\Omega_k} H(\Phi) \, \mathrm{d}\Omega. \tag{43}$$

In case of the PLIC method, the interface is assumed to be linear within each computational cell. This piecewise linear interface is reconstructed from the values of Ψ in neighboring cells. However, in terms of the Level-Set function, a piecewise linear interface means that Φ is *linearized* around Φ_k, which is the value of Φ in $\mathbf{x}_k$:

$$\Phi = \Phi_k + \nabla\Phi_k \cdot (\mathbf{x} - \mathbf{x}_k). \tag{44}$$

Substituting this in Eqn. (43) and taking $\xi = \frac{x-x_k}{\Delta x}$ and $\eta = \frac{y-y_k}{\Delta y}$ yields

$$\Psi_k = \int_{\xi=-\frac{1}{2}}^{\frac{1}{2}} \int_{\eta=-\frac{1}{2}}^{\frac{1}{2}} H(\Phi_k + \Delta x \left.\frac{\partial \Phi}{\partial x}\right|_k \xi + \Delta y \left.\frac{\partial \Phi}{\partial y}\right|_k \eta)\, \mathrm{d}\eta\, \mathrm{d}\xi. \tag{45}$$

Note that in contrast with other approaches, the Heaviside step function is not regularized. For the ease of integration the Heaviside step function is expressed as

$$H(\Phi) = \frac{1}{2} + \lim_{\alpha \to 0} \frac{1}{\pi} \arctan\left(\frac{\Phi}{\alpha^2}\right). \tag{46}$$

After some mathematical manipulations, the volume-of-fluid function Ψ_k is evaluated as

$$\Psi_k = \begin{cases} 0 & \Phi_k \leq -\Phi_{max\,k} \\ \frac{1}{2}\frac{(\Phi_{max\,k} + \Phi_k)^2}{\Phi_{max\,k}^2 - \Phi_{mid\,k}^2} & -\Phi_{max\,k} < \Phi_k < -\Phi_{mid\,k} \\ \frac{1}{2} + \frac{\Phi_k}{\Phi_{max\,k} + \Phi_{mid\,k}} & -\Phi_{mid\,k} \leq \Phi_k \leq \Phi_{mid\,k} \\ 1 - \frac{1}{2}\frac{(\Phi_{max\,k} - \Phi_k)^2}{\Phi_{max\,k}^2 - \Phi_{mid\,k}^2} & \Phi_{max\,k} < \Phi_k < \Phi_{max\,k} \\ 1 & \Phi_k \geq \Phi_{max\,k}, \end{cases} \tag{47}$$

where

$$\Phi_{max\,k} = \frac{1}{2}(|D_{x\,k}| + |D_{y_k}|) \tag{48}$$

and

$$\Phi_{mid\,k} = \frac{1}{2}\Big|\, |D_{y_k}| - |D_{x\,k}| \,\Big| \tag{49}$$

with

$$\begin{aligned} D_{x\,k} &= \Delta x \left.\tfrac{\partial \Phi}{\partial x}\right|_k \\ D_{y_k} &= \Delta y \left.\tfrac{\partial \Phi}{\partial y}\right|_k, \end{aligned} \tag{50}$$

which are approximated by central differencing. In Fig. 2 Ψ is plotted as function of Φ.

Step 2: Volume-of-Fluid advection At a certain time instant the volume-of-fluid function can be computed by means of Eqn. (47). The volume-of-fluid function after a time step is found by considering the amount of fluid that flows through the boundaries of a computational cell (see Fig. 3):

$$\begin{aligned} \Psi_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = \Psi_{i+\frac{1}{2},j+\frac{1}{2}}^{n} - \frac{1}{\Delta x \Delta y} \Big(& F_{x\,i+1,j+\frac{1}{2}} - F_{x\,i,j+\frac{1}{2}} + \\ & F_{y_{i+\frac{1}{2},j+1}} - F_{y_{i+\frac{1}{2},j}} \Big), \end{aligned} \tag{51}$$

where, in a Volume-of-Fluid manner, $F_{x\,i+1,j+\frac{1}{2}}$ and $F_{y_{i+\frac{1}{2},j}}$ are the amount of the $(\Phi > 0)$-fluid that flows from the so-called *donating regions* $\Omega_{D\,i+1,j+\frac{1}{2}}$ and

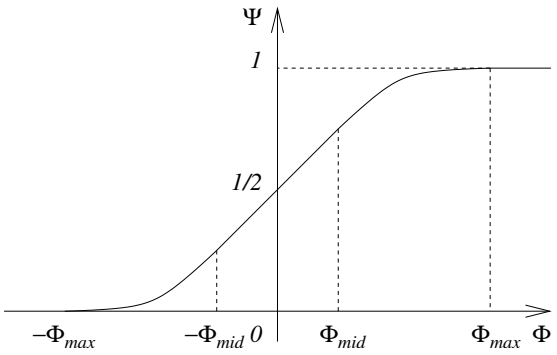

Figure 2: Fractional volume Ψ as function of Level-Set value Φ

$\Omega_{D_{i+\frac{1}{2},j}}$ through the faces $(i+1, j+\frac{1}{2})$ and $(i+\frac{1}{2}, j)$ respectively during the time step Δt:

$$\begin{aligned} F_{x_{i+1,j+\frac{1}{2}}} &= \int\limits_{\Omega_{D_{i+1,j+\frac{1}{2}}}} H(\Phi)\,\mathrm{d}\Omega \\ F_{y_{i+\frac{1}{2},j}} &= \int\limits_{\Omega_{D_{i+\frac{1}{2},j}}} H(\Phi)\,\mathrm{d}\Omega \end{aligned} \tag{52}$$

and similar for the other fluxes. Consider a face of a computational cell through which fluid flows during a time step. The donating region of this boundary is defined as the region from which this fluid originates at the beginning of the time-step. In other words, depending on the sign of the velocity at the face, the donating region can either be on the left or at the right neighboring cell. Formally, the flux can therefore be split in a contribution from both neighbors, called F^+ and F^- respectively (see Fig. 3). Of course if $F^+_{\ldots} \neq 0$ then $F^-_{\ldots} = 0$ and vice versa. In this fashion, both fluxes are:

$$\begin{aligned} F_{x_{i+1,j+\frac{1}{2}}} &= F^+_{x\ i+1,j+\frac{1}{2}} + F^-_{x\ i+1,j+\frac{1}{2}} \\ F_{y_{i+\frac{1}{2},j}} &= F^+_{y\ i+\frac{1}{2},j} + F^-_{y\ i+\frac{1}{2},j}. \end{aligned} \tag{53}$$

Note that u is the component of velocity in x-direction and v the component in y direction. The other fluxes are similar.

The fluxes are again computed by linearizing Φ (just like Eqn. (45)):

$$F^+_{x\ i,j+\frac{1}{2}} = \Delta x \Delta y \int\limits_{\xi=\frac{1}{2}-\nu^+}^{\frac{1}{2}} \int\limits_{\eta=-\frac{1}{2}}^{\frac{1}{2}} H(\Phi_L + D_{x_L}\xi + D_{y_L}\eta)\,\mathrm{d}\eta\,\mathrm{d}\xi \tag{54}$$

and

$$F^-_{x\ i,j+\frac{1}{2}} = -\Delta x \Delta y \int\limits_{\xi=-\frac{1}{2}}^{-\frac{1}{2}-\nu^-} \int\limits_{\eta=-\frac{1}{2}}^{\frac{1}{2}} H(\Phi_R + D_{x_R}\xi + D_{y_R}\eta)\,\mathrm{d}\eta\,\mathrm{d}\xi \tag{55}$$

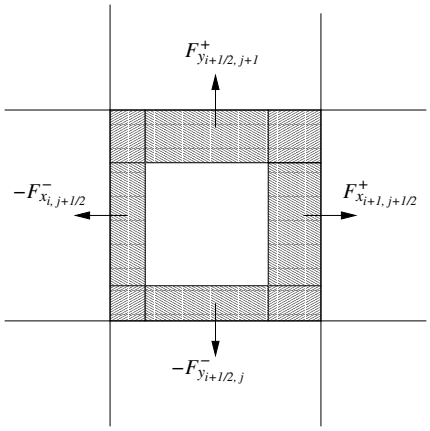

Figure 3: Donating regions for fluxes F_x and F_y

and similar for the other fluxes, where

$$\begin{array}{rcll} \nu^+ & = & \frac{\max(u_{i,j+\frac{1}{2}},0)\Delta t}{\Delta x} & \geq \quad 0 \\ \nu^- & = & \frac{\min(u_{i,j+\frac{1}{2}},0)\Delta t}{\Delta x} & \leq \quad 0 \end{array} \tag{56}$$

and

$$\begin{array}{rcl} \Phi_L & = & \Phi_{i-\frac{1}{2},j+\frac{1}{2}} \\ \Phi_R & = & \Phi_{i+\frac{1}{2},j+\frac{1}{2}} \\ D_{x_L} & = & \Delta x \left.\frac{\partial \Phi}{\partial x}\right|_{i-\frac{1}{2},j+\frac{1}{2}} \\ D_{y_L} & = & \Delta y \left.\frac{\partial \Phi}{\partial y}\right|_{i-\frac{1}{2},j+\frac{1}{2}} \\ D_{x_R} & = & \Delta x \left.\frac{\partial \Phi}{\partial x}\right|_{i+\frac{1}{2},j+\frac{1}{2}} \\ D_{y_R} & = & \Delta y \left.\frac{\partial \Phi}{\partial y}\right|_{i+\frac{1}{2},j+\frac{1}{2}} . \end{array} \tag{57}$$

With some scaling, these fluxes become

$$F^+_{x\ i,j+\frac{1}{2}} = \nu^+ \Delta x \Delta y \int_{\xi=-\frac{1}{2}}^{\frac{1}{2}} \int_{\eta=-\frac{1}{2}}^{\frac{1}{2}} H(\widehat{\Phi}_L + \widehat{D}_{x_L}\xi + D_{y_L}\eta)\,\mathrm{d}\eta\,\mathrm{d}\xi \tag{58}$$

and

$$F^-_{x\ i,j+\frac{1}{2}} = \nu^- \Delta x \Delta y \int_{\xi=-\frac{1}{2}}^{\frac{1}{2}} \int_{\eta=-\frac{1}{2}}^{\frac{1}{2}} H(\widehat{\Phi}_R + \widehat{D}_{x_R}\xi + D_{y_R}\eta)\,\mathrm{d}\eta\,\mathrm{d}\xi, \tag{59}$$

where

$$\begin{aligned}
\widehat{\Phi}_L &= \Phi_L + \tfrac{1}{2}(1-\nu^+)D_{x_L} \\
\widehat{\Phi}_R &= \Phi_R - \tfrac{1}{2}(1+\nu^-)D_{x_R} \\
\widehat{D}_{x_L} &= \nu^+ D_{x_L} \\
\widehat{D}_{x_R} &= -\nu^- D_{x_R}.
\end{aligned} \tag{60}$$

The integrals of Eqns. (58) and (59) are just Eqn. (45) with Φ_k replaced by $\widehat{\Phi}_L$ and $\widehat{\Phi}_R$ and D_x replaced by $\widehat{D}_{x_L}$ and $\widehat{D}_{x_R}$ respectively. Therefore, Eqns. (47), (48) and (49) are used to evaluate ${F_x^+}_{i,j+\frac{1}{2}}$ and ${F_x^-}_{i,j+\frac{1}{2}}$:

$$\begin{aligned}
{F_x^+}_{i,j+\frac{1}{2}} &= \nu^+ \Delta x \Delta y \; \Psi|_{\widehat{\Phi}_L, \widehat{D}_{x_L}, D_{y_L}} \\
{F_x^-}_{i,j+\frac{1}{2}} &= \nu^- \Delta x \Delta y \; \Psi|_{\widehat{\Phi}_R, \widehat{D}_{x_R}, D_{y_R}} .
\end{aligned} \tag{61}$$

The fluxes in y-direction are obtained in the same way.

In Fig. 3 it is shown that overlapping donating regions can exist in the corners of the cell. Fluid in those overlapping regions is fluxed more than once through different faces. As reported in for example [1], this can be solved by employing either a multidimensional scheme or flux-splitting. Here the second approach has been chosen, for reasons of simplicity. The order of fluxing is: first in x-direction, then in y-direction. Currently the flux-splitting of [46] is adopted:

$$\begin{aligned}
\Psi^*_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{\Psi^n_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{1}{\Delta x \Delta y}\left({F_x}^n_{i+1,j+\frac{1}{2}} - {F_x}^n_{i,j+\frac{1}{2}} \right)}{1 - \frac{\Delta t}{\Delta x}(u_{i+1,j+\frac{1}{2}} - u_{i,j+\frac{1}{2}})} \\
\Psi^{**}_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{\Psi^*_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{1}{\Delta x \Delta y}\left({F_y}^*_{i+\frac{1}{2},j+1} - {F_y}^*_{i+\frac{1}{2},j} \right)}{1 - \frac{\Delta t}{\Delta y}(v_{i+\frac{1}{2},j+1} - v_{i+\frac{1}{2},j})} \\
\Psi^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} &= \Psi^{**} - \Delta t \left(\Psi^* \frac{u_{i+1,j+\frac{1}{2}} - u_{i,j+\frac{1}{2}}}{\Delta x} + \Psi^{**} \frac{v_{i+\frac{1}{2},j+1} - v_{i+\frac{1}{2},j}}{\Delta y} \right).
\end{aligned} \tag{62}$$

Flux redistributing As reported in [46], undershoots and/or overshoots can still occur. These errors give rise to total mass errors of order 10^{-4}. This is also experienced in the current research. Mass errors are completely avoided by redistributing Ψ. The idea is to flux mass out of cells with $\Psi > 1$ and flux mass into cells with $\Psi < 0$. Since the trouble is in the doubly-fluxed regions, the fluxes are firstly taken from the diagonal $(i + \frac{1}{2} - \text{sign}(u), j + \frac{1}{2} - \text{sign}(v))$ neighboring cells.

Assume that cell k has a value $\Psi_k^{n+1} < 0$. Then define a flux F towards neighboring cell l, so that $\Psi_k = \Psi_k^{n+1} - F = 0$:

$$F = \Psi_k^{n+1}. \tag{63}$$

In order not to cause unphysical values of Ψ_l, limit F by

$$F = \min(\max(\Psi_k^{n+1}, -\Psi_l^{n+1}), 1 - \Psi_l^{n+1}), \tag{64}$$

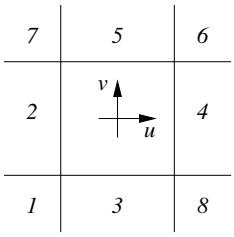

Figure 4: Order of flux redistributing for $u > 0$ and $v > 0$

so that $0 \leq \Psi_l = \Psi_l^{n+1} + F \leq 1$. Then make corrections to Φ_k and Φ_l by

$$\begin{aligned} \Psi_k &= \Psi_k^{n+1} - F \\ \Psi_l &= \Psi_k^{n+1} + F. \end{aligned} \tag{65}$$

In case of $\Psi_k^{n+1} > 1$, the flux F is

$$F = \min(\max(\Psi_k^{n+1} - 1, -\Psi_l^{n+1}), 1 - \Psi_l^{n+1}). \tag{66}$$

In two dimensions, there exist 8 neighboring cells l which can contribute to cell k. The order in which the neighboring cells l are subsequently chosen is depicted in Fig. 4. Note that the first step (in $-(\text{sign}(u), \text{sign}(v))$-direction) takes out most of the unphysical Ψ-values, since this is the direction the doubly-fluxed region was fluxed from. It is therefore important to take that neighbor first. The order of the other neighbors are arbitrary. The values u and v at Ψ locations are interpolated by averaging.

Step 3: Inverse function Having found a new fractional volume Ψ^{n+1}, the initial guess of the Level-Set function $\Phi^{n+1,*}$ (after Level-Set advection) is modified, such that mass is conserved within each computational cell. In other words, find $(\Phi_1, \Phi_2, \dots)$, such that

$$\Psi_k(\Phi_1, \Phi_2, \dots) - \Psi_k^{n+1} = 0 \qquad \forall k = 1, 2, \dots . \tag{67}$$

It will be clear that due to the behavior of Ψ no unique solution Φ exists. However, a (small) correction to Φ^* is searched, where Φ^* comes from Level-Set advection. A solution Φ is found by the repeating (i.e. iteratively) until convergence: leave Φ unmodified in a grid point when the Volume-of-Fluid constraint is satisfied and make corrections locally when this constraint is not satisfied. This is achieved by using the inverse function of Ψ_k as given in Eqn. (47) with respect to argument Φ_k and employing Picard-iterations. Starting with $(\Phi_1^{n+1,*}, \Phi_2^{n+1,*}, \dots)$, if at time step $n+1$ $(\Psi_1, \Psi_2, \dots)$ has to be equal to

$(\Psi_1^{n+1}, \Psi_2^{n+1}, \dots)$, then the m^{th} iteration is:

$$\begin{aligned} D_x &= \Delta x \left.\frac{\partial \Phi}{\partial x}\right|_k^{n+1,m} \\ D_y &= \Delta y \left.\frac{\partial \Phi}{\partial y}\right|_k^{n+1,m} \\ \Phi_{max} &= \tfrac{1}{2}(|D_x| + |D_y|) \\ \Phi_{mid} &= \tfrac{1}{2}\Big| \, |D_y| - |D_x| \, \Big| \end{aligned} \tag{68}$$

and if $\Psi_k^{n+1,m} \neq \Psi_k^{n+1}$ (else $\Phi_k^{n+1,m+1} = \Phi_k^{n+1,m}$)

$$\Phi_k^{n+1,m+1} = \begin{cases} -\Phi_{max} & \Psi_k^{n+1} \leq 0 \\ \mathcal{B} & 0 < \Psi_k^{n+1} < 1 - \Psi_{mid} \\ \mathcal{C} & 1 - \Psi_{mid} \leq \Psi_k^{n+1} \leq \Psi_{mid} \\ \mathcal{D} & \Psi_{mid} < \Psi_k^{n+1} < 1 \\ \Phi_{max} & \Psi_k^{n+1} \geq 1, \end{cases} \tag{69}$$

where

$$\begin{aligned} \mathcal{B} &= \sqrt{2\Psi_k^{n+1}(\Phi_{max}^2 - \Phi_{mid}^2)} - \Phi_{max} \\ \mathcal{C} &= (\Psi_k^{n+1} - \tfrac{1}{2})(\Phi_{max} + \Phi_{mid}) \\ \mathcal{D} &= -\sqrt{2(1 - \Psi_k^{n+1})(\Phi_{max}^2 - \Phi_{mid}^2)} + \Phi_{max} \end{aligned} \tag{70}$$

and

$$\Psi_{mid} = \frac{1}{2}\frac{\Phi_{max} + 3\Phi_{mid}}{\Phi_{max} + \Phi_{mid}}. \tag{71}$$

These iterations are repeated until

$$\max_k |\Psi_k^{n+1,m+1} - \Psi_k^{n+1}| \leq \epsilon, \tag{72}$$

where ϵ is a tolerance, typically

$$\epsilon = 1 \times 10^{-8}. \tag{73}$$

A graphical overview of the method is depicted in Fig. 5.

3.3 Time-step restrictions

Following [39, 47, 50], an adaptive time stepping procedure is chosen by considering the time-step restrictions due to convection, diffusion and surface tension effects. Since the Level-Set function Φ and the Volume-of-Fluid function Ψ are advected explicitly, the restriction due to advection is:

$$\Delta t_c = \frac{1}{\frac{|u|_{max}}{\Delta x} + \frac{|v|_{max}}{\Delta y}}. \tag{74}$$

The restriction due to surface tension in [47, 50] is

$$\Delta t_s = \frac{1}{\sqrt{\frac{\sigma |\kappa|_{max}}{\min(\rho_0, \rho_1)\min(\Delta x, \Delta y)^2}}}. \tag{75}$$

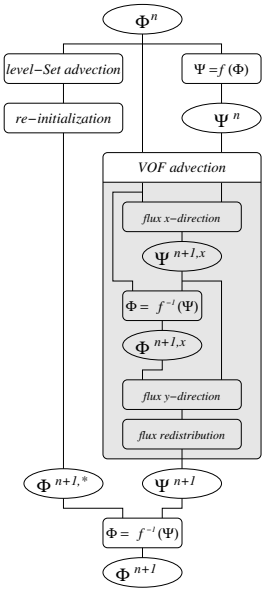

Figure 5: MCLS method: interface advection; Φ: Level-Set function; Ψ: Volume-of-Fluid function

Since the surface tension forces are regularized i.e. $\sigma\kappa$ is replaced by $\sigma\kappa\delta(\Phi)h$ and $h = \min(\Delta x, \Delta y)$, the restriction becomes

$$\Delta t_s = \frac{1}{\sqrt{\frac{|\sigma\kappa\delta(\Phi)|_{max}}{\min(\rho_0,\rho_1)\min(\Delta x,\Delta y)}}}. \tag{76}$$

Diffusion is accounted for implicitly, hence no time-step restriction is encountered. For the time step Δt finally holds (see e.g. [39])

$$\Delta t \leq \mathrm{CFL}\min(\Delta t_c, \Delta t_s), \tag{77}$$

where, again following [39, 47, 50], $\mathrm{CFL} = \frac{1}{2}$ is used.

4 Applications

The method is applied to show the behavior of the MCLS approach. First of all, the advection algorithm is tested by advecting an interface by a prescribed velocity field. Subsequently, the method is illustrated by considering a falling drop and a rising bubble respectively.

4.1 Advection tests

4.1.1 Linear advection

The first advection test is presented in Fig. 6. The velocity field is prescribed

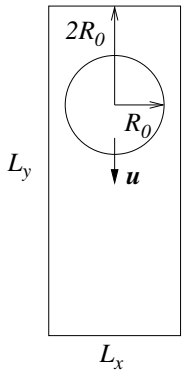

Figure 6: Linear advection test

by

$$\begin{aligned} u &= 0 \\ v &= -1. \end{aligned} \tag{78}$$

The dimensions of the computational domain are

$$\begin{aligned} L_x &= 10 \\ L_y &= 100, \end{aligned} \tag{79}$$

which is discretized by a 10x100-mesh. Initially a circle of radius R_0 is placed at $x = L_x/2$ and $y = L_y - 2R_0$. For the case of $R_0 = 4$ (a circle with a diameter of 8 mesh sizes), the relative mass is plotted in Fig. 7 as function of the traversed distance of the circle. First-order, second-order and third-order simulations are included. Also re-initialization and VOF advection are included. The truncation error of re-initialization is the same the order as the truncation error of advection. The tolerance in the VOF advection is taken to be:

$$\epsilon = 1\ 10^{-8}. \tag{80}$$

Globally speaking:

- mass is always lost without the VOF advection;
- mass losses are smaller for higher accuracy;
- re-initialization causes much higher mass losses;
- the MCLS method conserves mass up to a specified tolerance.

4.1.2 Zalesak's rotating disc

The advection test of Zalesak ([25]) is used often in literature to demonstrate the interface-advection algorithm (see e.g. [26, 31, 33, 56] for VOF methods and [39, 40, 45, 46] for Level-Set methods). A slotted disc (Fig. 8) is rotated through

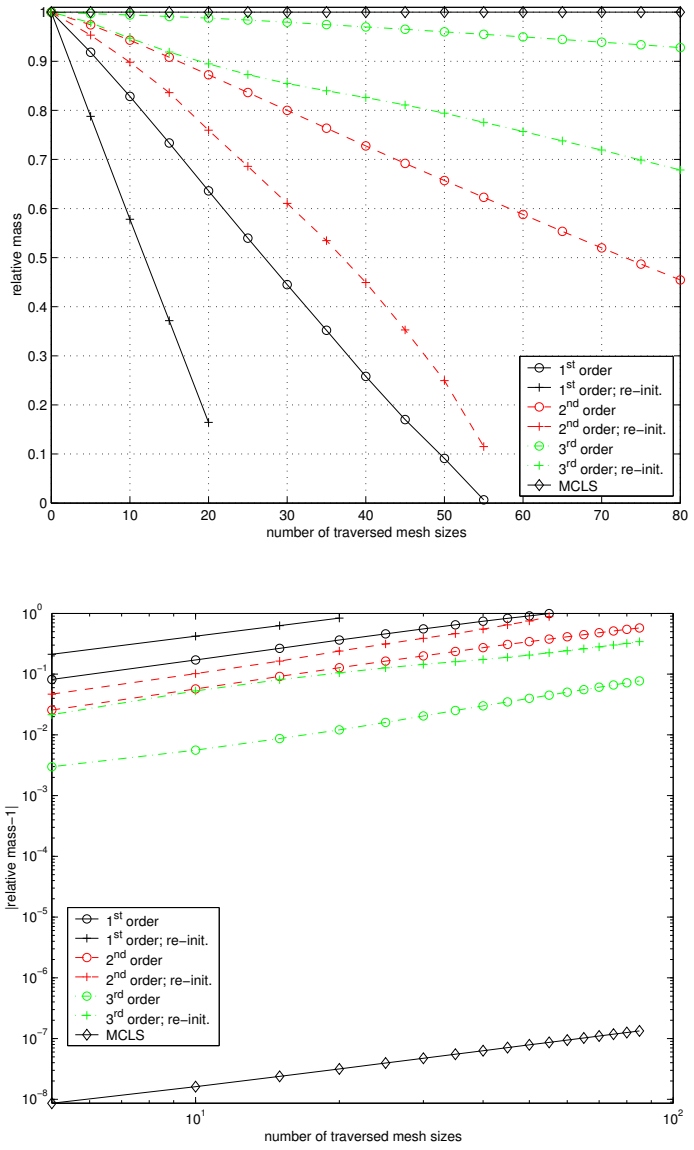

Figure 7: Relative mass for the linear advection test; $\epsilon = 1\ 10^{-8}$ (every 10^{th} iteration marked)

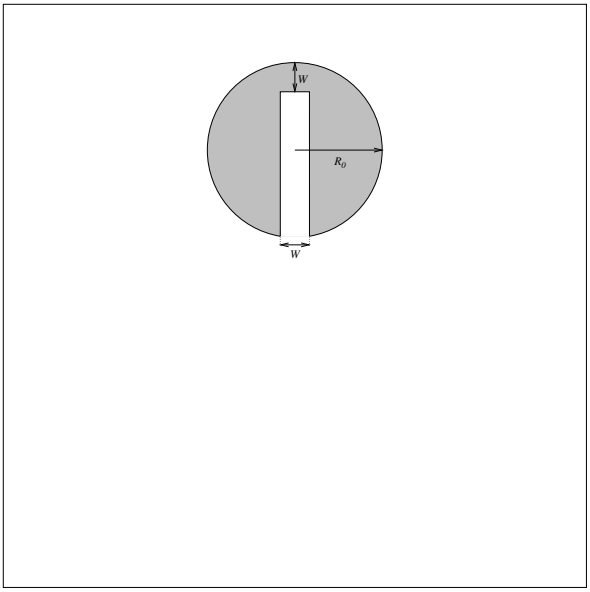

Figure 8: Zalesak's slotted disc advection test (on scale)

one revolution around the center of the computational domain. The velocity-field is prescribed by

$$\begin{aligned} u &= -(y - \tfrac{1}{2}L_y) \\ v &= x - \tfrac{1}{2}L_x. \end{aligned} \tag{81}$$

The center of the slotted disc $(x_0, y_0)^t$ is located at

$$\begin{aligned} x_0 &= \tfrac{1}{2}L_x \\ y_0 &= \tfrac{3}{4}L_y. \end{aligned} \tag{82}$$

The sizes are

$$\begin{aligned} L_x &= L_y \\ R_0 &= \tfrac{3}{20}L_x \\ W &= \tfrac{1}{3}R_0. \end{aligned} \tag{83}$$

In Figs. 9 and 10 results are shown for 50×50, 100×100, 150×150 and 200×200 mesh sizes. As might be expected from the linear advection discussed in the foregoing, mass is still lost in case of the high-order Level-Set method. For the MCLS method, mass is conserved up to the specified tolerance ϵ although mass is redistributed due to numerical diffusion. Results of the MCLS method are comparable with VOF/PLIC methods (see aforementioned references). Note that the Level-Set advection is first-order in case of the MCLS method.

If S represents the interface after one revolution, the length $l(S)$ of S is then

$$l(S) = \int_S \mathrm{d}S, \tag{84}$$

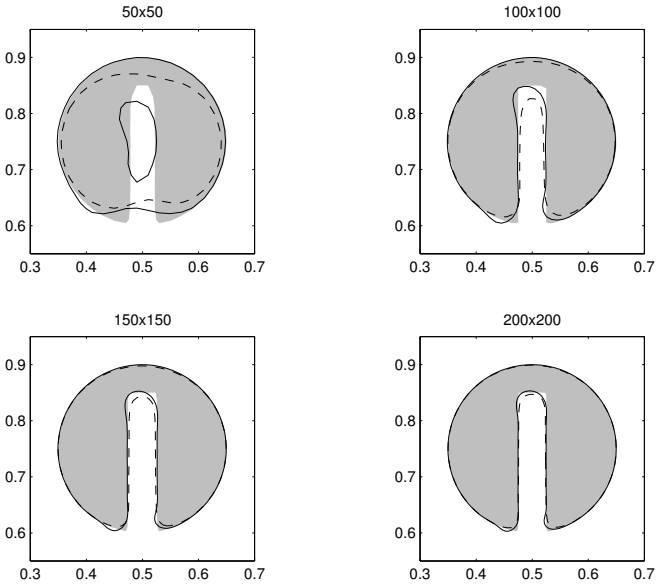

Figure 9: Results for Zalesak's advection test; shaded: initial contour; dashed lines: 3^{rd} order; solid lines: MCLS

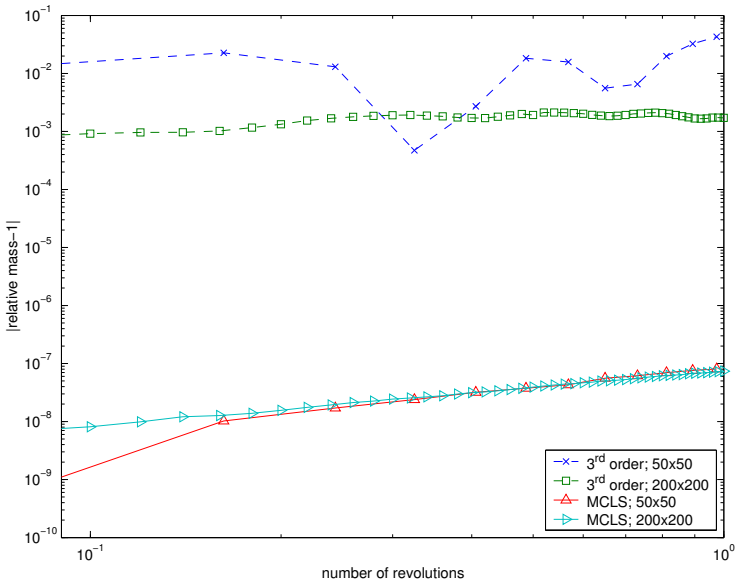

Figure 10: Relative masses for Zalesak's advection test; $\epsilon = 1 \times 10^{-8}$ (every 50^{th} iteration marked)

$\frac{l(S)}{l(S^0)}$	50×50	100×100	150×150	200×200
initial	0.86094	0.98187	0.98804	0.99102
3^{rd} order	0.49236	0.80940	0.91253	0.93318
MCLS	0.84106	0.95977	0.97020	0.97570

Table 1: Computed interface lengths after one revolution

which can be expressed as (see e.g. [38])

$$l(S) = \int_\Omega \delta(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega. \tag{85}$$

This is approximated by using central differences and regularization of the Dirac delta function (see Eqn. (12)):

$$l(S) = \int_\Omega \delta_\alpha(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega, \tag{86}$$

where

$$\delta_\alpha(x) = \begin{cases} 0, & |x| > \alpha \\ \frac{1+\cos(\frac{\pi x}{\alpha})}{2\alpha}, & |x| \leq \alpha. \end{cases} \tag{87}$$

Note that due to Eqn. (12), the exact value of Φ has no meaning in the Level-Set formulation; only the sign is relevant. The α in Eqn. (86) equals therefore the α of Eqn. (12). The exact length of the interface is

$$l(S^0) = \left(4 + 2\pi - 2\arctan(\frac{1}{2}\frac{W}{R_0}) - \frac{W}{R_0}\right) R_0. \tag{88}$$

In Table (1) the computed interface lengths are compared with the exact length. 'Initial' means at $t = 0$, when errors are made due to the regularization of the delta function. Furthermore, '3^{rd} order' and 'MCLS' correspond to the interface lengths after one revolution.

Since Φ^0 is a distance function, $|\Phi - \Phi^0|$ is a measure for the shift of the interface after one revolution. A norm of the error e can now be defined as

$$||e||_p = \left(\frac{\int_S |\frac{\Phi-\Phi^0}{L_x}|^p\,\mathrm{d}S}{\int_S \mathrm{d}S}\right)^{\frac{1}{p}} = \left(\frac{\int_\Omega |\frac{\Phi-\Phi^0}{L_x}|^p\ \delta_\alpha(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega}{\int_\Omega \delta_\alpha(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega}\right)^{\frac{1}{p}}, \tag{89}$$

where L_x is used to non-dimensionalize Φ and Φ^0 is the initial Level-Set function. The results are presented in Fig. 11.

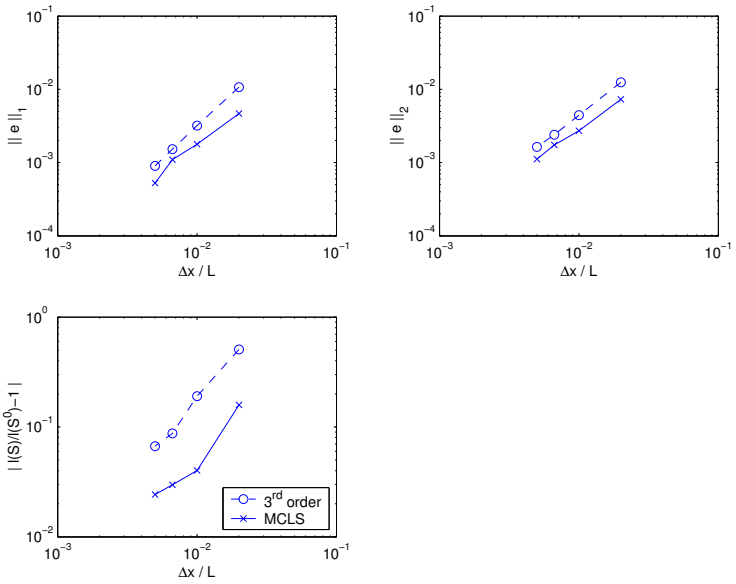

Figure 11: Errors for Zalesak's advection test

4.2 Air/water flow

In [47, 50] a two-dimensional rising air bubble in water is considered. The dimensions and sizes are

$$\begin{array}{rcl} L_x & = & 0.02\ m \\ L_y & = & 1\frac{1}{2}L_x \\ R_0 & = & \frac{1}{6}L_x \\ x_0 & = & \frac{1}{2}L_x \\ y_0 & = & \frac{1}{2}L_x. \end{array} \tag{90}$$

The gravity and material constants are

$$\begin{array}{rclcrcl} g & = & 9.8\ \frac{m}{s^2} & & \sigma & = & 0.0728\ \frac{kg}{s^2} \\ \rho_w & = & 10^3\ \frac{kg}{m^3} & & \rho_a & = & 1.226\ \frac{kg}{m^3} \\ \mu_w & = & 1.137\ 10^{-3}\ \frac{kg}{ms} & & \mu_a & = & 1.78\ 10^{-5}\ \frac{kg}{ms}, \end{array} \tag{91}$$

where subscripts $_w$ and $_a$ indicate water and air respectively. Results are shown in Fig. 12 for three different mesh sizes. These are 30×45, 40×60 and 60×90. We take $\epsilon = 1 \times 10^{-8}$. Relative mass losses are of the same order and in agreement with the advection tests. Note that mesh sizes are much smaller than in [47, 50]. The results are the same for $t \leq 0.025$ for all mesh sizes. Thereafter small differences occur. The results compare well with [47, 50]. The MCLS method seems to result in a more coherent structure at the highly curved part of the interface at $t = 0.05$. This is thought to be caused by the low resolution of the grids used here.

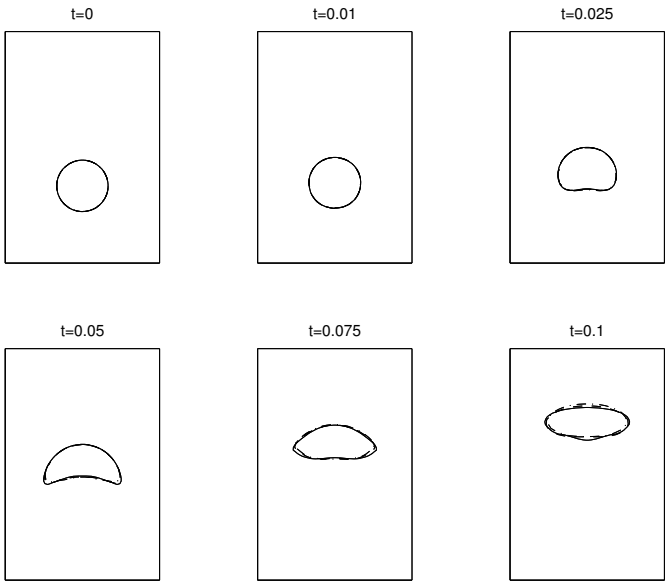

Figure 12: Rising bubble; $-\cdot-$: 30×45; $--$: 40×60; — : 60×90 mesh

In Fig. 13 results are shown for a falling droplet. The conditions are the same as for the rising bubble, except for the sign of Φ at $t = 0$ and

$$y_0 \quad = \quad L_x. \tag{92}$$

Mass conservation properties are the same as before. The result are the same until the droplet hits the bottom. Thereafter differences occur. This is thought to be due to limited number of grid cells available to capture the flow-phenomena near the wall. The results compare well with [47, 50]. Note that the results in [47, 50] span $t \leq 0.05$; no results after collision are presented.

5 Conclusion

A Mass Conserving Level-Set (MCLS) has presented. The method is based on a coupling of the Level-Set with the Volume-of-Fluid methodology. Advection tests were used to compare the method with the Level-Set method. Mass is conserved up to a specified (vanishing) tolerance. In this way the MCLS method is superior to the Level-Set method. On the other hand the implementation is much easier than for a Volume-of-Fluid method. The applicability of the MCLS method was illustrated by the application to air-water flows. It is possible to capture bubbles or droplets within a limited number of grid cells without mass losses up to the prescribed tolerance. This is an important feature, since future work will concern three-dimensional space, where the amount of grid

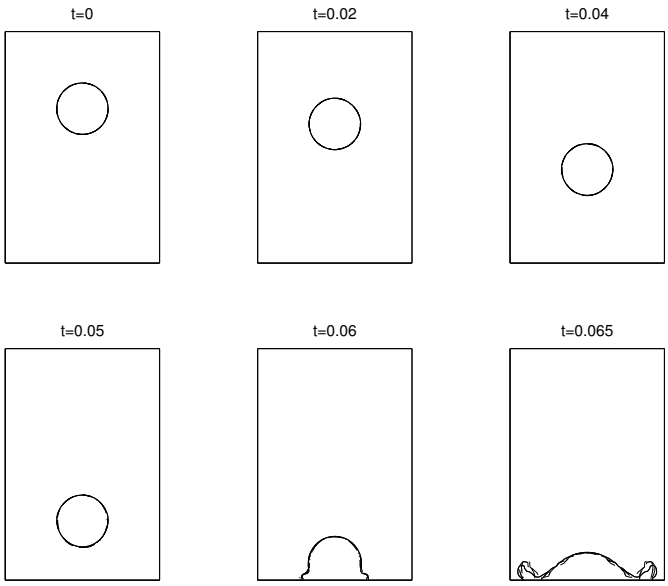

Figure 13: Falling droplet; $- \cdot -$: 30×45; $- -$: 40×60; — : 60×90 mesh

cells available to an individual entity will decrease considerably. Computing on a larger scale is intended in the near future.

References

[1] S.P. van der Pijl. Free-boundary methods for multi-phase flows. AMA report 02-13, Delft University of Technology, 2002. http://ta.twi.tudelft.nl.

[2] S.O.Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100:25–37, 1992.

[3] A. Esmaeeli and G. Tryggvason. Direct numerical simulations of bubbly flows, part 1, low reynolds number arrays. *Journal of FLuid Mechanics*, 377:313–345, 1998.

[4] G. Agresar, J.J. Linderman, G. Tryggvason, and K.G. Powell. An adaptive, Cartesian, front-tracking method for the motion, deformation and adhesion of circulation cells. *Journal of Computational Physics*, 143:346–380, 1998.

[5] B. Bunner and G. Tryggvason. Direct numerical simulations of three-dimensional bubbly flows. *Physics of Fluids*, 11:1967–1969, 1999.

[6] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. A front-tracking method for

the computation of multiphase flow. *Journal of Computational Physics*, 169:708–759, 2001.

[7] C.S. Peskin and B.F. Printz. Improved volume conservation in the computation of flows with immersed elastic boundaries. *Journal of Computational Physics*, 105:33–46, 1993.

[8] A.M. Roma, C.S. Peskin, and M.J. Berger. An adaptive version of the immersed boundary method. *Journal of Computational Physics*, 153:509–534, 1999.

[9] M.-C. Lai and C.S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, 160:705–719, 2000.

[10] E. Jung and C.S. Peskin. Two-dimensional simulations of valveless pumping using the immersed boundary method. *SIAM Journal on Applied Mathematics*, 23(1):19–45, 2001.

[11] J.U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100:335–354, 1992.

[12] W.Shyy, H.S.Udaykumar, M.M. Rao, and R.W. Smith. *Computational Fluid Dynamics with Moving Boundaries*. Series in computational and physical processes in mechanics and thermal sciences. Taylor and Francis, Washington DC, 1996.

[13] H.S. Udaykumar, H.-C. Kan, and W. Shyy R. Tran-Son-Tay. Multiphase dynamics in arbitrary geometries on fixed cartesian grids. *Journal of Computational Physics*, 137:366–405, 1997.

[14] T. Ye, R. Mittal, H.S. Udaykumar, and W. Shyy. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, 156:209–240, 1999.

[15] G. Yang, D.M. Causon, and D.M. Ingram. Calculation of compressible flows about complex moving geometries using a three-dimensional Cartesian cut cell method. *International Journal for Numerical Methods in Fluids*, 33:1121–1151, 2000.

[16] H.S. Udaykumar, R. Mittal, P. Rampunggoon, and A. Khanna. A sharp interface cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics*, 174:345–380, 2001.

[17] H.S. Udaykumar, R. Mittal, and P. Rampunggoon. Interface tracking finite volume method for complex solid-fluid interactions on fixed meshes. *Communications in Numerical Methods in Engineering*, 18:89–97, 2002.

[18] T. Ye, W. Shyy, , and J.N. Chung. A fixed-grid, sharp-interface method for bubble dynamics and phase change. *Journal of Computational Physics*, 174:781–815, 2001.

[19] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surfaces. *Physics of Fluids*, 8:2182–2189, 1965.

[20] S. Chen, B. Johnson, P.E. Raad, and D. Fadda. The surface marker and micro cell method. *International Journal for Numerical Methods in Fluids*, 25:749–778, 1997.

[21] W.F. Noh and P. Woodward. SLIC (Simple Line Interface Calculations). In A.I. van de Vooren and P.J. Zandbergen, editors, *Lecture Notes in Physics, Vol. 59*, pages 330–340, New York, 1976. Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics, Springer.

[22] T. Yabe, F. Xiao, and T. Utsumi. The constrained interpolation profile method for multiphase analysis. *Journal of Computational Physics*, 169:556–593, 2001.

[23] T. Nakamura, R. Tanaka, T. Yabe, and K. Takizawa. A numerical method for two-phase flow consisting of seperate compressible and incompressible regions. *Journal of Computational Physics*, 174:171–207, 2001.

[24] T. Yabe and F. Xiao. Description of complex and sharp interface with fixed grids in incopmressible and compressible fluid. *Computers and Mathematics with Applications*, 29:1:15–25, 1995.

[25] S.T. Zalesak. Fully multidimensional flux-corrected transport algorithm for fluids. *Journal of Computational Physics*, 31:335–362, 1979.

[26] M. Rudman. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids*, 24:671–691, 1997.

[27] C.W. Hirt and B.D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[28] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics*, 113:134–147, 1994.

[29] M. Rudman. A volume-tracking methods for incompressible multifluid flow with large density variations. *International Journal for Numerical Methods in Fluids*, 28:357–378, 1998.

[30] D. Geuyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics*, 152:423–456, 1999.

[31] W.J. Rider and D.B. Kothe. Reconstructing volume tracking. *Journal of Computational Physics*, 141:112–152, 1998.

[32] M. Renardy, Y. Renardy, and J. Li. Numerical simulation of moving contact line problems using a volume-of-fluid method. *Journal of Computational Physics*, 171:243–263, 2001.

[33] D.J.E. Harvie and D.F. Fletcher. A new volume of fluid advection algorithm: the defined donating region scheme. *International Journal for Numerical Methods in Fluids*, 35:151–172, 2001.

[34] W. Mulder, S. Osher, and J.A. Sethian. Computing interface motion in compressible gas dynamics. *Journal of Computational Physics*, 100:209–228, 1992.

[35] J.A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge monographs on applied and computational mathematics. Cambridge University Press, Cambridge, 1999.

[36] J.A. Sethian. Evolution, implementation, and application of level set and fast marching methods for advancing fronts. *Journal of Computational Physics*, 169:503–555, 2001.

[37] S. Osher and R.P. Fedkiw. Level set methods: An overview and some recent results. *Journal of Computational Physics*, 169:463–502, 2001.

[38] Y.C. Chang, T.Y. Hou, B. Merriman, and S. Osher. A level set formulation of eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, 124:449–464, 1996.

[39] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. An improved level set method for incompressible two-phase flows. *Computers and Fluids*, 27:663–680, 1998.

[40] M. Sussman and E. Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on Scientific Computing*, 20(4):1165–1191, 1999.

[41] H. Haj-Hariri and Q. Shi. Thermocapillary motion of deformable drops at finite Reynolds and Marangoni numbers. *Physics of Fluids*, 9:845–855, 1997.

[42] M. Sussman, A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, and M.L. Welcome. An adaptive level set approach for incompressible two-phase flows. *Journal of Computational Physics*, 148:81–124, 1999.

[43] L.L. Zheng and H. Zhang. An adaptive level set method for moving-boundary problems: Application to droplet spreading and solidification. *Numerical Heat Transfer, Part B*, 37:437–454, 2000.

[44] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

[45] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. UCLA CAM Report 02-04, University of California, Los Angeles, 2002.

[46] M. Sussman and E.G. Puckett. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162:301–337, 2000.

[47] M. Kang, R.P. Fedkiw, and D.Q. Nguyen. A boundary condition capturing method for multiphase incompressible flow. UCLA CAM Report 99-27, University of California, Los Angeles, 1999.

[48] Z. Li and M.-C. Lai. The immersed interface method for the navier-stokes equations with singular forces. *Journal of Computational Physics*, 171:822–842, 2001.

[49] J.J.I.M. van Kan. A second-order accurate pressure correction method for viscous incompressible flow. *SIAM J. Sci. Stat. Comp.*, 7:870–891, 1986.

[50] M. Kang, R.P. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, pages 323–360, 2000.

[51] X.-D. Liu, R.P. Fedkiw, and M. Kang. A boundary condition capturing method for Poisson's equation on irregular domains. *Journal of Computational Physics*, 160:151–178, 2000.

[52] R.J. Leveque and Z. Li. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM Journal on Scientific Computing*, 18(3):709–735, 1997.

[53] A. Wiegmann and K.P. Bube. The explixit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions. *SIAM Journal on Numerical Analysis*, 37(3):827–862, 2000.

[54] T.Y. Hou, Z. Li, S. Osher, and H. Zhao. A hybrid method for moving interface problems with application to the Hele-Shaw flow. *Journal of Computational Physics*, 134:236–252, 1997.

[55] Z. Li and S.R. Lubkin. Numerical analysis of interfacial two-dimensional Stokes flow with discontinuous viscosity and variable surface tension. *International Journal for Numerical Methods in Fluids*, 37:525–540, 2001.

[56] D.J.E. Harvie and D.F. Fletcher. A new volume of fluid advection algorithm: the stream scheme. *Journal of Computational Physics*, 162:1–32, 2000.