DELFT UNIVERSITY OF TECHNOLOGY

REPORT 06-10

Modeling of three-dimensional multi-phase flows with a Mass-Conserving Level-Set method

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

ISSN 1389-6520

Reports of the Delft Institute of Applied Mathematics

Delft 2006

[Modeling of three-dimensional multi-phase flows with a Mass-Conserving Level-Set method]

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

March, 2006

Abstract

In this work incompressible two-phase flows are considered. The aim is to model high density-ratio flows with arbitrary complex interface topologies, such as occur in air/water flows. Between the phases a sharp front exists, where density and viscosity change abruptly.

The computational method used in this paper is the Mass Conserving Level-Set method. It is based on the Level-Set methodology, using a VOF-function to conserve mass. This function is advected without the necessity to reconstruct the interface. The ease of the method is based on an explicit relationship between the Volume-of-Fluid function and the Level-Set function. The method is straightforward to apply to arbitrarily shaped interfaces, which may collide and break up.

In this work special attention is paid on the extension of the MCLS method to three dimensional problems, re-initialization of the Level-Set function, redistribution of mass errors and computation of curvature.

1 Introduction

The Mass-Conserving Level-Set method ([1]) is used to model bubbly flows in three dimensional space. It is a combination of the Level-Set and Volume-of-Fluid methods. The Level-Set method (see e.g. [2–4]) has some advantages over all other methods. Especially where solving the flow-field is concerned, since interface normals, curvature and distance towards the interface can be expressed easily in terms of the Level-Set function Φ and its derivatives. Also, advecting the interface is possible by application of 'of-the-shelf' techniques for hyperbolic conservation laws. For these reasons, the Level-Set method has been chosen as the basis of our work. However, mass-conservation is not an intrinsic property, which is considered the major drawback of the Level-Set method. The Volume-of-Fluid method (see e.g. [5–7]) on the other hand is mass-conserving by construction. A drawback of this method might be that the interface is not as easily advected and the computation of interface normals and curvature is not as straightforward as with the Level-Set method.

This work has a shared foundation with the CLSVOF method ([8, 9]) and to a lesser extent with the combined Level-Set/particle method ([10]) in the sense that it is based on

Level-Set and additional effort is made to conserve mass. Like the CLSVOF method, our method combines the Level-Set method with a Volume-of-Fluid method. The difference is that instead of combining two existing methods, the MCLS approach uses an explicit relationship between the Volume-of-Fluid and Level-Set functions and is based on finding small corrections to the Level-Set function only.

For the modeling of surface tension effects, it is necessary to compute the curvature κ of the interface. The curvature κ is related to the first and second order derivatives of Level-Set function Φ. In the MCLS approach corrections to Φ are made near the interface. Since these corrections are confined to a small region near the interface, errors in the computation of the derivatives of Φ are introduced locally. These errors contribute to inaccuracies in the computation of the curvature and consequently to the surface tension forces. This introduces unphysical, parasitic currents.

The MCLS method has been described and applied to two-dimensional test-cases in Van der Pijl et al. [1, 11]. Some three-dimensional test-cases are presented in [12], albeit without surface tension. This paper focuses on the extension of MCLS to three-dimensional space and surface tension computation (Section 4). In the two-dimensional case flux redistribution has been used to address under and overshoots of Ψ due to flux splitting. This could be extended to three dimensions. However, a more powerful approach, namely mass redistribution, is chosen here. It is discussed in Section 4.4. The MCLS approach is based on finding small corrections to the Level-Set function Φ. Under certain circumstances, the re-initialization procedure of Sussman et al. [13] shifts the interface too much so that the correction becomes too large. A simple modification to the re-initialization procedure is presented in Section 5.

The appearance of parasitic currents when modeling surface tension effects are discussed in Section 6. To reduce parasitic currents, re-initialization is studied again in Section 6.3. Parasitic currents are further reduced by smoothing the curvature in Section 6.5.

The applicability of the method to three-dimensional air-water flows is demonstrated by the simulation of rising bubbles and falling droplets in Section 7. Results with and without surface-tension effects are presented.

2 Governing Equations

Consider two incompressible fluids '0' and '1' in domain $\Omega \in \mathbb{R}^3$. The fluids are separated by an interface S. The flow is governed by the incompressible Navier-Stokes equations:

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

subject to the incompressibility constraint

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

where $\mathbf{u} = (u, v, w)^t$ is the velocity vector and ρ, p, μ and $\mathbf{g}$ are the density, pressure, viscosity and gravity vector respectively. We assume the density and viscosity constant within each fluid.

The coupling of the fluids is achieved by the interface conditions, which express continuity of mass and momentum at the interface:

$$\begin{array}{rcl} [\mathbf{u}] & = & 0, \\ [p\mathbf{n} - \mathbf{n} \cdot \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^t\right)] & = & \sigma \kappa \mathbf{n}, \end{array} \tag{3}$$

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. In [14, 15] it is shown that if the viscosity μ is continuous at the interface, the derivatives of the velocity components are continuous too. In that case Eqn. (3) reduces to $[\mathbf{u}] = \mathbf{0}$ and $[p] = \sigma\kappa$. To achieve that, the viscosity is made continuous by smoothing it over three mesh widths. Instead of exactly taking into account the pressure-jump at the interface due to the surface tension forces, the continuous surface force/stress (CSF, [16]) methodology is adopted.

3 Computational Approach

The Navier-Stokes equations are solved on a Cartesian grid in a rectangular domain by the pressure-correction method ([17]). The unknowns are stored in a Marker-and-Cell (staggered) layout ([18]). For the interface representation the Level-Set methodology is adopted. The interface conditions are satisfied by means of the continuous surface force (CSF) methodology. The discontinuous density field is dealt with similarly to the ghost fluid method for incompressible flow ([14]).

3.1 Pressure Correction

The Navier-Stokes equations are discretized using finite differences on a staggered grid as explained in [1] for the two-dimensional case. First a tentative velocity vector $\mathbf{u}^*$ is computed by:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\nabla \cdot \mathbf{u}^n \mathbf{u}^n + \frac{1}{\rho} \nabla \cdot \mu \left(\nabla \mathbf{u}^* + (\nabla \mathbf{u}^n)^t\right), \tag{4}$$

where ^{n} denotes time-level n. The stress tensor is split into a part on time level * (implicit) and ^{n} (explicit), due to the fact that u, v and w are solved sequentially. Note that since $\nabla \cdot \mathbf{u} = 0$ the term $\nabla \cdot \mu \left((\nabla \mathbf{u}^n)^t\right)$ vanishes away from the interface where $\mu =$ constant. The resulting system of equations is solved by an incomplete Cholesky preconditioned Conjugate Gradient (ICCG) method. The velocities at the new time instant $^{n+1}$ are computed by:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho} G p + \mathbf{g} + \mathbf{f}, \tag{5}$$

under the constraint of Eqn. (2). This gives

$$\left\{ \begin{array}{rcl} \mathbf{u}^{n+1} & = & \mathbf{u}^* + \Delta t \left(-\frac{1}{\rho} G p + \mathbf{g} + \mathbf{f}\right), \\ D\mathbf{u}^{n+1} & = & 0, \end{array} \right. \tag{6}$$

where D represents the discretization of the divergence and G is the discrete gradient operator, which remains to be specified. The term $\mathbf{f}$ in Eqns. (5) and (6) is due to the CSF methodology (see e.g. [19]). It is adopted to incorporate the pressure jump due to interface tension forces (see Eqn. (3)). With the CSF methodology, the interface tension forces are rewritten as volume forces $\mathbf{f}$ and regularized:

$$\mathbf{f} = -\frac{1}{\bar{\rho}}\sigma\kappa\delta_\alpha(\Phi)\nabla\Phi. \tag{7}$$

In this expression Φ is the Level-Set function, $\bar{\rho}$ is the average density, σ is the surface tension coefficient, $\kappa = \nabla \cdot \frac{\nabla\Phi}{|\nabla\Phi|}$ is the curvature of the interface and δ_α is the regularized Dirac delta function. More details can be found in [1].

Discretization of $\frac{1}{\rho}Gp$ requires special care, because p and ρ are discontinuous at the interface. In [20], the pressure derivatives $\frac{1}{\rho}\frac{\partial p}{\partial x}$ at u locations $(i+1, j+\frac{1}{2}, k+\frac{1}{2})$ are approximated by

$$\left(\beta\frac{\partial p}{\partial x}\right)_{i+1,j+\frac{1}{2},k+\frac{1}{2}} = \hat{\beta}_{i+1,j+\frac{1}{2},k+\frac{1}{2}}\frac{p_{i+\frac{3}{2},j+\frac{1}{2},k+\frac{1}{2}} - p_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}}{\Delta x}. \tag{8}$$

The quantity $\hat{\beta}$ is the harmonic average of $\beta = \frac{1}{\rho}$. Similar expression are used for $\beta\frac{\partial p}{\partial y}$ and $\beta\frac{\partial p}{\partial z}$. Here we make use of $[\frac{1}{\rho}\nabla p] = \mathbf{0}$ ([14]). Note that due to the Continuous Surface Force (CSF) methodology no pressure jumps have to be taken into account at the interface. Further details can be found in [20].

Finally Eqn. (6) gives:

$$D\frac{1}{\rho}Gp = D\left(\frac{1}{\Delta t}\mathbf{u}^* + \mathbf{g} + \mathbf{f}\right). \tag{9}$$

Further details on this pressure-correction method can be found in [17] and the application to two-dimensional multi-phase flows in [1].

3.2 Time-step restrictions

The extension of the time-step restriction from the two-dimensional work ([1]) to three dimensions is trivial. Following Kang et al. [14] and Sussman et al. [19], an adaptive time stepping procedure is chosen by considering the time-step restrictions due to convection and surface tension effects. Diffusion is accounted for implicitly, hence no time-step restriction is encountered. The time-step restriction is:

$$\Delta t \leq \mathrm{CFL}\min(\Delta t_c, \Delta t_s), \tag{10}$$

where, following Kang et al. [14] and Sussman et al. [19], $\mathrm{CFL} = \frac{1}{2}$ is used. Here Δt_c is the restriction due to advection:

$$\Delta t_c = \frac{1}{\frac{|u|_{max}}{\Delta x} + \frac{|v|_{max}}{\Delta y} + \frac{|w|_{max}}{\Delta z}} \tag{11}$$

Δt_s is the restriction due to surface tension. The restriction due to surface tension given by Kang et al. [14] is

$$\Delta t_s = \frac{1}{\sqrt{\frac{\sigma|\kappa|_{max}}{\min(\rho_0,\rho_1)\min(\Delta x,\Delta y)^2}}}. \tag{12}$$

Since the surface tension force is regularized, i.e. $\frac{1}{\rho}\sigma\kappa$ is replaced by $\frac{1}{\frac{1}{2}(\rho_0+\rho_1)}\sigma\kappa\delta(\Phi)h$ and $h = \min(\Delta x, \Delta y, \Delta z)$, the restriction becomes

$$\Delta t_s = \frac{1}{\sqrt{\frac{|\sigma\kappa\delta(\Phi)|_{max}}{\frac{1}{2}(\rho_0+\rho_1)\min(\Delta x,\Delta y,\Delta y)}}}. \tag{13}$$

4 Interface advection

For the interface representation and advection the Level-Set approach has been chosen. A motivation for this approach is given in [1]. The interface, say S, is defined as the zero level-set of Φ:

$$S(t) = \left\{\mathbf{x} \in \mathbb{R}^3 | \Phi(\mathbf{x}, t) = 0\right\}. \tag{14}$$

The interface is evolved by advecting the Level-Set function in the flow field as if it were a material property

$$\frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \nabla \Phi = 0. \tag{15}$$

A homogeneous Neumann boundary condition for Φ is imposed at the boundaries. First-order spatial discretization and forward Euler temporal discretization are employed.

The interface advection is decoupled from the flow-field computations. The flow-field is computed with a given interface position and the interface is advected with the new flow-field. Symbolically, if

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} &= \mathcal{A}(\mathbf{u}, \Phi), \\ \frac{\partial \Phi}{\partial t} &= \mathcal{B}(\Phi, \mathbf{u}), \end{aligned} \tag{16}$$

then the decoupled temporal discretization is given by

$$\begin{aligned} \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} &= \mathcal{A}(\mathbf{u}^n, \mathbf{u}^{n+1}, \Phi^{n+\frac{1}{2}}), \\ \frac{\Phi^{n+\frac{3}{2}} - \Phi^{n+\frac{1}{2}}}{\Delta t} &= \mathcal{B}(\Phi^{n+\frac{1}{2}}, \Phi^{n+\frac{3}{2}}, \mathbf{u}^{n+1}). \end{aligned} \tag{17}$$

Re-initialization is applied as described in [13, 21] to keep Φ a distance function. This ensures that the front has finite thickness at all time, which is important when the surface tension force is distributed over a number of grid cells (CSF approach).

The difficulty with the Level-Set method is that conservation of Φ does not imply conservation of mass. In order to conserve mass with the Level-Set method, corrections to the Level-Set function are made by considering the fractional volume Ψ of a certain fluid

within a computational cell. First the usual Level-Set advection is performed: first-order advection and re-initialization as described above. Since the obtained Level-Set function $\Phi^{n+1,*}$ will certainly not conserve mass, 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: $\Psi^n = f(\Phi^n, \nabla\Phi^n)$;

2. the volume-of-fluid function has to be advected conservatively 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 $f(\Phi^{n+1}, \nabla\Phi^{n+1}) = \Psi^{n+1}$ holds.

These three steps have been explained for the two-dimensional case in [1]. The extension to three dimensions is described here.

4.1 Step 1: Volume-of-Fluid function

In the computational domain two functions $\Phi : \mathbb{R}^3 \to \mathbb{R}$ and $\Psi : \mathbb{R}^3 \to \mathbb{R}$ are considered. These are the Level-Set function $\Phi(\mathbf{x})$ and the Volume-of-Fluid function $\Psi(\mathbf{x})$ respectively, where $\mathbf{x} \in \mathbb{R}^3$. The Volume-of-Fluid function Ψ in a computational cell Ω_k is defined by

$$\Psi(\mathbf{x}_k) = \frac{1}{\mathrm{vol}(\Omega_k)} \int_{\Omega_k} H(\Phi)\, \mathrm{d}\Omega, \tag{18}$$

where H is the Heaviside step function and Ω_k is a cube with sizes Δx, Δy and Δz. Assume that $\mathbf{x}_k$ corresponds to the center of the cube Ω_k. The linearization of Φ around $\mathbf{x}_k$ is called φ:

$$\varphi(\mathbf{y}; \Phi, \nabla\Phi, \mathbf{x}_k) = \Phi_k + (\tfrac{\partial\Phi}{\partial x}, \tfrac{\partial\Phi}{\partial y}, \tfrac{\partial\Phi}{\partial z})_k^t \cdot (\mathbf{y} - \mathbf{x}_k), \tag{19}$$

where the derivatives are approximated by central differences. The linearization of Φ can be used to approximate Ψ:

$$\Psi(\mathbf{x}_k) \approx f(\Phi_k, \nabla\Phi_k). \tag{20}$$

The goal is to find a function f. To ease the analysis considerably, map Ω onto a unit cube with coordinates $(\xi, \eta, \zeta) \in (-\frac{1}{2}, \frac{1}{2})^3$. The linearization can be written as

$$\varphi = \Phi_k + D_\xi \xi + D_\eta \eta + D_\zeta \zeta, \tag{21}$$

where the axes are chosen such that

$$D_\xi \geq D_\eta \geq D_\zeta \geq 0. \tag{22}$$

This choice will limit the number of possible interface topologies in the cube. As a result

$$\begin{aligned} D_\xi &= \max(|\Delta x\big(\tfrac{\partial\Phi}{\partial x}\big)_k|, |\Delta y\Big(\tfrac{\partial\Phi}{\partial y}\Big)_k|, |\Delta z\big(\tfrac{\partial\Phi}{\partial z}\big)_k|), \\ D_\zeta &= \min(|\Delta x\big(\tfrac{\partial\Phi}{\partial x}\big)_k|, |\Delta y\Big(\tfrac{\partial\Phi}{\partial y}\Big)_k|, |\Delta z\big(\tfrac{\partial\Phi}{\partial z}\big)_k|), \\ D_\eta &= |\Delta x\big(\tfrac{\partial\Phi}{\partial x}\big)_k| + |\Delta y\Big(\tfrac{\partial\Phi}{\partial y}\Big)_k| + |\Delta z\big(\tfrac{\partial\Phi}{\partial z}\big)_k| - D_\xi - D_\zeta. \end{aligned} \tag{23}$$

Since Φ is linearized, the zero level-set (the interface) of Φ is a plane.

The Volume-of-Fluid function will be derived geometrically by computing the relative volume enclosed by the cut plane within Ω_k. It is then important to take into account the topology of the cut plane. The topology of the cut plane changes when it passes a corner of the cube. Due to symmetry, only the cases $\Phi \leq 0$ will be considered here. Two cases are defined:

$$\begin{aligned} \text{case I} &: D_\xi \leq D_\eta + D_\zeta \\ \text{case II} &: D_\xi > D_\eta + D_\zeta \end{aligned} \tag{24}$$

Note that there are only two cases due to the choice of ξ, η and ζ. In Fig. 1 a geometrical representation is shown which can be used to compute f. This is achieved by using the pyramids at corners A, B, etcetera. These have length, width and height $\frac{\Phi_A}{D_\xi}$, $\frac{\Phi_A}{D_\eta}$, $\frac{\Phi_A}{D_\zeta}$ respectively and similar for the other corners. Here Φ_A is the value $\varphi(\mathbf{x}_A)$ at corner A, etcetera:

$$\begin{aligned} \Phi_A &= \Phi_k + \tfrac{1}{2}D_\xi + \tfrac{1}{2}D_\eta + \tfrac{1}{2}D_\zeta \\ \Phi_B &= \Phi_k + \tfrac{1}{2}D_\xi + \tfrac{1}{2}D_\eta - \tfrac{1}{2}D_\zeta \\ \Phi_C &= \Phi_k + \tfrac{1}{2}D_\xi - \tfrac{1}{2}D_\eta + \tfrac{1}{2}D_\zeta \\ \Phi_D &= \Phi_k - \tfrac{1}{2}D_\xi + \tfrac{1}{2}D_\eta + \tfrac{1}{2}D_\zeta \\ \Phi_E &= \Phi_k + \tfrac{1}{2}D_\xi - \tfrac{1}{2}D_\eta - \tfrac{1}{2}D_\zeta. \end{aligned} \tag{25}$$

Function f then follows as:

$$f = \frac{\mathcal{A}}{6D_\xi D_\eta D_\zeta} \qquad \Phi \leq 0 \tag{26}$$

and

$$f = 1 - f(-\Phi, \nabla\Phi) \qquad \Phi > 0, \tag{27}$$

where

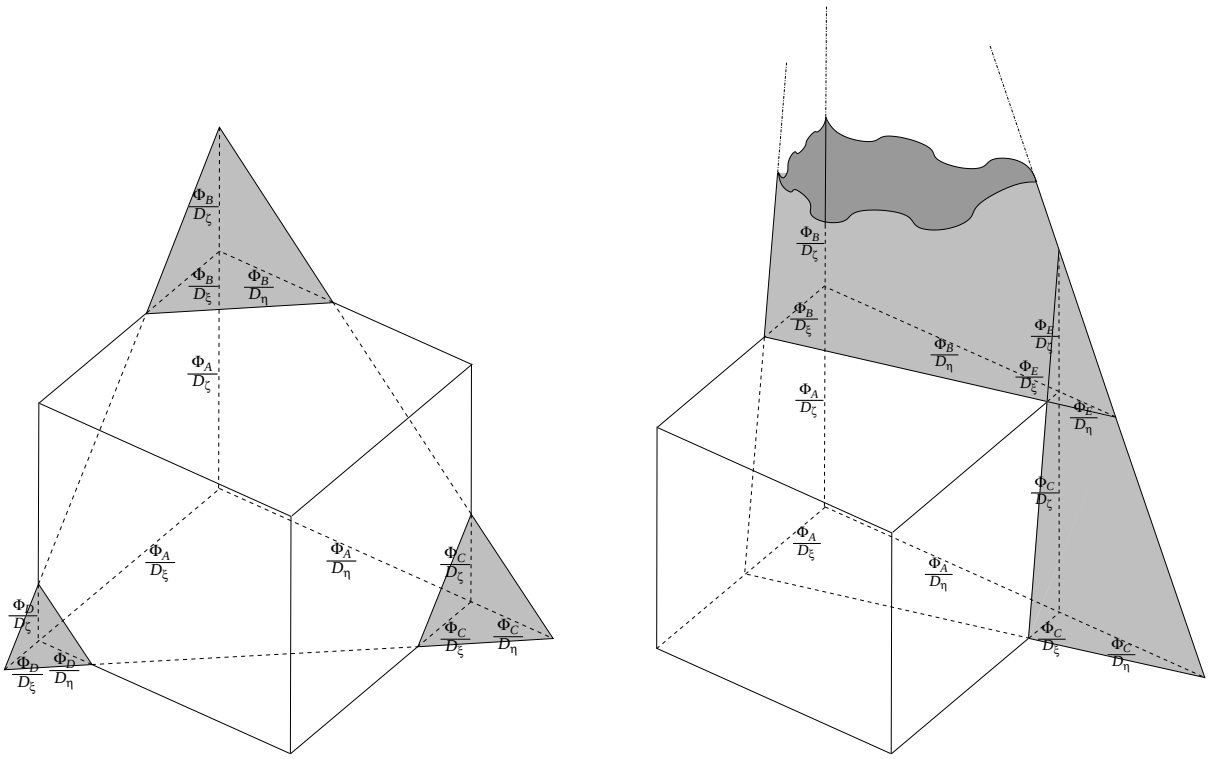
$$\begin{aligned} \mathcal{A} = \; & \max(\Phi_A, 0)^3 - \max(\Phi_B, 0)^3 - \\ & \max(\Phi_C, 0)^3 - \max(\Phi_D, 0)^3 + \\ & \max(\Phi_E, 0)^3. \end{aligned} \tag{28}$$

Examples for the cases in (24) are shown in Figs. 2(a) and 2(b).

4.2 Step 2: Volume-of-Fluid advection

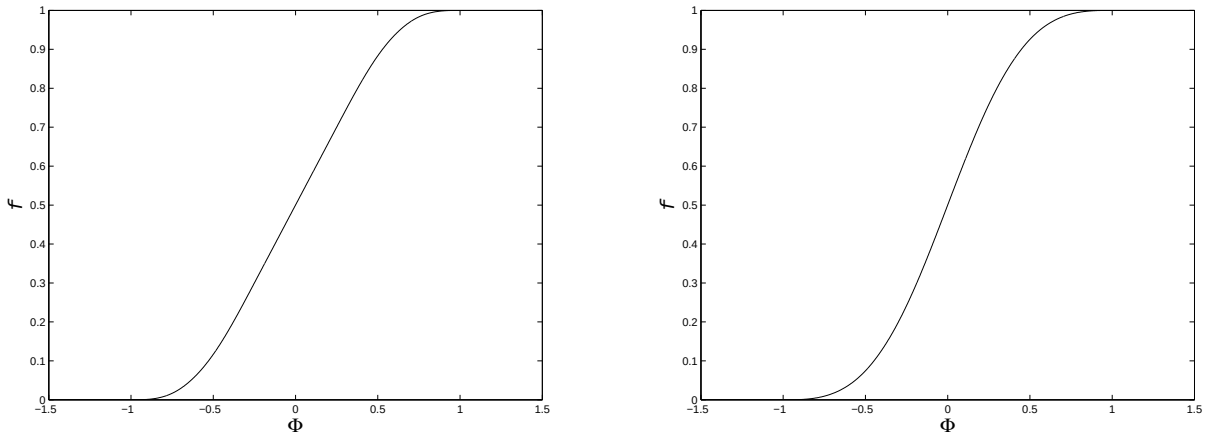
The Volume-of-Fluid advection is a straightforward extension from the two-dimensional case as described in [1]. It is in terms of the Level-Set methodology:

$$\frac{\mathrm{d}\Psi(\mathbf{x}_k)}{\mathrm{d}t} + \frac{1}{\mathrm{vol}(\Omega_k)} \int_{\partial\Omega_k} H(\Phi)\mathbf{u}\cdot\mathbf{n}\,\mathrm{d}S = 0. \tag{29}$$



(a) case I (b) case II

Figure 1: Volumes for $\Phi \leq 0$



(a) $D_\xi = 1.25$; $D_\eta = 0.5$; $D_\zeta = 0.25$ (case I) (b) $D_\xi = 0.8$; $D_\eta = 0.7$; $D_\zeta = 0.5$ (case II)

Figure 2: Examples of Volume-of-Fluid function

The Volume-of-Fluid function after a time step is found by considering the flux of fluid that flows through the boundaries of a computational cell during time-step Δt. Consider a face Γ of the control volume Ω_k. The flux F through the boundary Γ is

$$F = \int_{\Delta t} \int_{\partial\Omega_k} H(\Phi(\mathbf{x}, t+\tau))\ \mathbf{u}(\mathbf{x}, t+\tau) \cdot \mathbf{n}\ \mathrm{d}S\, \mathrm{d}\tau. \tag{30}$$

This flux can be rewritten as

$$F = \int_{\Omega_D} H(\Phi(\mathbf{x}, t))\, \mathrm{d}\Omega, \tag{31}$$

where Ω_D is the *donating region* of face Γ, which initially contains all fluid that will flow through face Γ during time-step Δt (see Fig. 3). Summation over all boundary faces leads for computational cell (i, j, k) to:

$$\begin{aligned} \Psi^{n+\frac{3}{2}}_{i,j,k} = \Psi^{n+\frac{1}{2}}_{i,j,k} - \frac{1}{\Delta x \Delta y \Delta z} \Big(& F_{x i+\frac{1}{2},j,k} - F_{x i-\frac{1}{2},j,k} + \\ & F_{y i,j+\frac{1}{2},k} - F_{y i,j-\frac{1}{2},k} + \\ & F_{z i,j,k+\frac{1}{2}} - F_{z i,j,k-\frac{1}{2}} \Big), \end{aligned} \tag{32}$$

where the subscripts indicate the corresponding boundary face. Depending on the sign of the velocity at the face, the donating region can either be on the left-hand or at the right-hand side neighboring cell. Formally, the flux can therefore be split into a contribution from both neighbors, called F^+ and F^- respectively (see Fig. 3 for the two-dimensional case). Of course if $F^+_{\ldots} \neq 0$ then $F^-_{\ldots} = 0$ and vice versa. In this way the fluxes in x, y and z direction at a face can be written as (omitting the subscripts):

$$F_x = F^+_x + F^-_x, \qquad F_y = F^+_y + F^-_y, \qquad F_z = F^+_z + F^-_z. \tag{33}$$

Since the fluxes measure the fractional volume in the denoting region, Eqn. (20) can be applied with some scaling of variables. For the fluxes in z direction for example:

$$\begin{aligned} F^+_{z\ i+\frac{1}{2},j+\frac{1}{2},k} &= \Delta x \Delta y \Delta z \nu^+\ f\left(\widehat{\Phi}_L, (\partial_x \Phi_L, \partial_y \Phi_L, \nu^+ \partial_z \Phi_L)^t\right) \\ F^-_{z\ i+\frac{1}{2},j+\frac{1}{2},k} &= \Delta x \Delta y \Delta z \nu^-\ f\left(\widehat{\Phi}_R, (\partial_x \Phi_R, \partial_y \Phi_R, -\nu^- \partial_z \Phi_R)^t\right). \end{aligned} \tag{34}$$

where

$$\nu^+ = \tfrac{\max(u,0)\Delta t}{\Delta x}, \quad \nu^- = \tfrac{\min(u,0)\Delta t}{\Delta x}, \tag{35}$$

and

$$\begin{aligned} \Phi_L &= \Phi_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}, & \Phi_R &= \Phi_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}, \\ \partial_x \Phi_L &= \tfrac{\partial \Phi}{\partial x}\big|_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}, & \partial_x \Phi_R &= \tfrac{\partial \Phi}{\partial x}\big|_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}, \\ \partial_y \Phi_L &= \tfrac{\partial \Phi}{\partial y}\big|_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}, & \partial_y \Phi_R &= \tfrac{\partial \Phi}{\partial y}\big|_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}, \\ \partial_z \Phi_L &= \tfrac{\partial \Phi}{\partial z}\big|_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}, & \partial_z \Phi_R &= \tfrac{\partial \Phi}{\partial z}\big|_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}. \end{aligned} \tag{36}$$

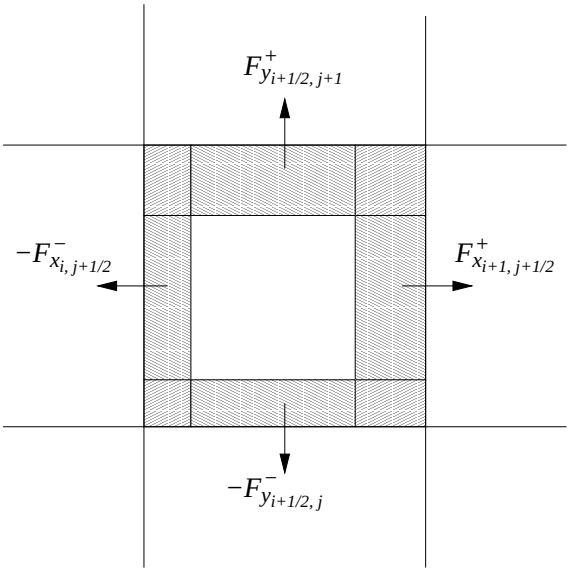


Figure 3: Donating regions for fluxes F_x and F_y. These are the shaded areas that will flow through the boundaries during a time step. Doubly fluxed areas exist near the corners of the cell.

and

$$\widehat{\Phi}_L \;=\; \Phi_L + \tfrac{1}{2}(1-\nu^+)\Delta z\partial_z\Phi_L, \quad \widehat{\Phi}_R \;=\; \Phi_R - \tfrac{1}{2}(1+\nu^-)\Delta z\partial_z\Phi_R. \tag{37}$$

Time-step Δt and velocity u are included in ν^+ and ν^-. This scaling of variables makes the advection of Ψ rather straightforward, since an analytic expression for the function f is given in Eqns. (26) and (27). The fluxes in the other direction are obtained in the same way.

Fig. 3 illustrates 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. This can be remedied by employing either a multidimensional scheme or flux-splitting. For simplicity we have chosen the second approach. The order of fluxing is: first in x-direction, then in y-direction and then in z-direction. The flux-splitting of Sussman and Puckett [8]

is adopted:

$$
\begin{aligned}
\Psi^{(x)}_{i,j,k} &= \frac{\Psi^{n}_{i,j,k} - \frac{1}{\Delta x \Delta y \Delta z}\left({F_x}^{n}_{i+\frac{1}{2},j,k} - {F_x}^{n}_{i-\frac{1}{2},j,k} \right)}{1 - \frac{\Delta t}{\Delta x}(u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k})}, \\
\Psi^{(y)}_{i,j,k} &= \frac{\Psi^{(x)}_{i,j,k} - \frac{1}{\Delta x \Delta y \Delta z}\left({F_y}^{(x)}_{i,j+\frac{1}{2},k} - {F_y}^{(x)}_{i,j-\frac{1}{2},k} \right)}{1 - \frac{\Delta t}{\Delta y}(v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k})}, \\
\Psi^{(z)}_{i,j,k} &= \frac{\Psi^{(y)}_{i,j,k} - \frac{1}{\Delta x \Delta y \Delta z}\left({F_z}^{(y)}_{i,j,k+\frac{1}{2}} - {F_z}^{(y)}_{i,j,k-\frac{1}{2}} \right)}{1 - \frac{\Delta t}{\Delta z}(w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}})}, \\
\Psi^{n+1}_{i,j,k} &= \Psi^{(z)}_{i,j,k} - \Delta t \Big(\Psi^{(x)}_{i,j,k} \frac{u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}}{\Delta x} + \\
&\qquad \Psi^{(y)}_{i,j,k} \frac{v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}}{\Delta y} + \\
&\qquad \Psi^{(z)}_{i,j,k} \frac{w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}}{\Delta z} \Big).
\end{aligned}
\tag{38}
$$

The fluxes ${F_y}^{(x)}$ are computed with a corrected Level-Set function $\Phi^{(x)} : f(\Phi^{(x)}, \nabla\Phi^{(x)}) = \Psi^{(x)}$ and similar for ${F_z}^{(y)}$. Note that any other flux (or operator)-splitting technique could be adopted. Note also that due to the construction of Eqn. (32) the quantity Ψ is conserved, which is necessary for mass conservation.

4.3 Step 3: Inverse function

The Volume-of-Fluid function $\Psi^{n+\frac{3}{2}}$ was computed in the previous step. The Level-Set function after pure Level-Set advection is called Φ^*. Note that Φ^* will in general not conserve mass, i.e.

$$f(\Phi^*_k, \nabla\Phi^*_k) \neq \Psi^{n+\frac{3}{2}}_k. \tag{39}$$

In order to conserve mass within each computational cell, a correction $\Delta\Phi$ to Φ^* is sought, such that mass is conserved within each computational cell:

$$\left| f({\Phi_k}^{n+\frac{3}{2}}, \nabla{\Phi_k}^{n+\frac{3}{2}}) - {\Psi_k}^{n+\frac{3}{2}} \right| \leq \epsilon \quad \forall k, \tag{40}$$

where $\Phi^{n+\frac{3}{2}} = \Phi^* + \Delta\Phi$ is the Level-Set function at the new time-level, f is the Volume-of-Fluid function (step 1) and ϵ is some tolerance. It will be clear that due to the behavior of Ψ no unique solution Φ exists. On the other hand, it can be expected that the corrections $\Delta\Phi$ are small, since the mass errors in Φ^* are due to the truncation error of the discretization of Eqn. (15). Therefore a simple Picard method is used. Using the gradients of Φ from the previous iteration, a new update for Φ is found by computing

$$\Phi^{(l+1)}_k(\mathbf{x}) = \begin{cases} g(\Psi^{n+\frac{3}{2}}_k, \nabla\Phi^{(l)}_k) & |\Psi^{(l)}_k - \Psi^{n+\frac{3}{2}}_k| > \epsilon \\ \Phi^{(k)}_k & |\Psi^{(l)}_k - \Psi^{n+\frac{3}{2}}_k| \leq \epsilon \end{cases} \quad \forall k, \tag{41}$$

where $\Psi^{(l)}_k = f(\Phi^{(l)}_k, \nabla\Phi^{(l)}_k)$, typically $\epsilon = 10^{-8}$ and g is the inverse function of f with respect to its first argument:

$$f(g(\Psi, \nabla\Phi), \nabla\Phi) = \Psi. \tag{42}$$

An analytic expression for g was available in the two-dimensional case ([1]). However, expressing g analytically for the three-dimensional case is a complicated task. Instead Newton iterations are used to compute $\Phi^{(k+1)}$:

$$\Phi^{(k+1),(l+1)} = \Phi^{(k+1),(l)} + \frac{\Psi^0 - f(\Phi^{(k),(l)}, \nabla\Phi^{(k)})}{\frac{\partial f}{\partial \Phi}(\Phi^{(k),(l)}, \nabla\Phi^{(k)})}. \tag{43}$$

The derivative $\frac{\partial f}{\partial \Phi}$ is computed analytically. The iterations are stopped if$|\Psi_k^{n+1} - f(\dots)| \leq \epsilon$.

4.4 Mass redistribution

For the advection of the Volume-of-Fluid function Ψ (step 2, Section 4.2) operator splitting is applied. However, undershoots and/or overshoots can still occur as reported by Sussman et al.[8]. This leads to unphysical values of Ψ, namely < 0 and > 1. This is due to the non-vanishing source term in Eqn. (38):

$$\Delta t \left(\Psi^{(x)}_{i,j,k} \left(\frac{\partial u}{\partial x} \right)_{i,j,k} + \Psi^{(y)}_{i,j,k} \left(\frac{\partial v}{\partial y} \right)_{i,j,k} + \Psi^{(z)}_{i,j,k} \left(\frac{\partial w}{\partial z} \right)_{i,j,k} \right).$$

If the unphysical values are replaced by 0 or 1, mass errors arise which are in general of the order 10^{-4} (Sussman and Puckett [8]). This is also observed with the present method. Redistribution of Ψ avoids these mass errors. The idea is to flux mass out of cells with $\Psi > 1$ and flux mass into cells with $\Psi < 0$.

Besides the unphysical values of Ψ, mass can also be distributed without the presence of an interface. This is the case when the Volume-of-Fluid function Ψ is between and not equal to 0 and 1, but no interface is present within the corresponding computational cell. In that case an entity exists with a size smaller than a computational cell. Although formally not a mass error, it is not taken into account by the Ghost-Fluid method when the flow-field is computed and will be called 'numerical vapor'. However, if numerical vapor would be bluntly removed, it would cause mass errors.

There will not be any interface in a control volume $\Omega_{i,j,k}$ around $x_{i,j,k}$ $\Phi_{i,j,k}\Phi_{i\pm1,j,k} > 0$ and $\Phi_{i,j,k}\Phi_{i,j\pm1,k} > 0$ and $\Phi_{i,j,k}\Phi_{i,j,k\pm1} > 0$.

Summarizing, mass errors arise when

$$\begin{array}{lll}
\text{operator splitting:} & \Psi_{i,j,k} < 0 \quad \vee \quad \Psi_{i,j,k} > 1 & \\
\text{numerical vapor:} & \Psi_{i,j,k} \in (0,1) \quad \wedge & \\
& \Phi_{i,j,k}\Phi_{i+1,j,k} > 0 \quad \wedge \quad \Phi_{i,j,k}\Phi_{i-1,j,k} > 0 \quad \wedge & \\
& \Phi_{i,j,k}\Phi_{i,j+1,k} > 0 \quad \wedge \quad \Phi_{i,j,k}\Phi_{i,j-1,k} > 0 \quad \wedge & \\
& \Phi_{i,j,k}\Phi_{i,j,k+1} > 0 \quad \wedge \quad \Phi_{i,j,k}\Phi_{i,j,k-1} > 0. &
\end{array} \tag{44}$$

Assume that after the interface advection the Level-Set function Φ and the Volume-of-Fluid function Ψ are known. The Volume-of-Fluid function is mass-conserving, but contains unwanted values, i.e. $\Psi < 0$ and $\Psi > 1$ and 'numerical vapor'.

If Ψ contains unwanted values then modified functions $\hat{\Psi}$ and $\hat{\Phi}$ are sought such that:

- $\hat{\Psi}$ does not contain unwanted values according to $\hat{\Phi}$,
- $\hat{\Psi}$ conserves mass,
- $\hat{\Phi}$ is close to Φ in order not to destroy local accuracy.

The first constraint would suggest to just remove the unwanted values from Ψ, i.e. set $\hat{\Psi}$ to 0 or 1 if no interface is present and limit to $0 \leq \hat{\Psi} \leq 1$. However, this would violate the second constraint since mass errors $\epsilon(\mathbf{x})$ arise:

$$\epsilon = \hat{\Psi} - \Psi. \tag{45}$$

The total mass lost is then, of course, $\int_\Omega \epsilon \, \mathrm{d}\Omega$. The idea is now to first *redistribute* ϵ to $\hat{\epsilon}$ conservatively and then to add it back to Ψ to obtain the redistributed $\hat{\Psi}$ (which automatically conserves mass):

$$\hat{\Psi} = \Psi + \hat{\epsilon}. \tag{46}$$

The new $\hat{\Psi}$ has to satisfy all three constraints.

From the foregoing constraints it follows that a procedure is needed to *redistribute* ϵ to $\hat{\epsilon}$, such that:

- $\hat{\Psi}$ does not contain unwanted values according to $\hat{\Phi}$,
- the total amount of ϵ is conserved (in order to conserve mass),
- $\|\hat{\Phi} - \Phi\|$ is small, so that $\hat{\Phi}$ will be close to Φ.

The last remark would suggest to formulate a constrained minimization problem. We choose not to proceed in this direction. Instead, ϵ will be redistributed in the direction of the nearest interface, to keep the redistributing effect on Φ local and thus minimizing $\|\hat{\Phi} - \Phi\|$.

Note that $\hat{\Psi}$ and $\hat{\Phi}$ are coupled, which means that changing Ψ will cause changes in Φ, complicating matters even further. A decoupled approach (by means of Picard iterations) will therefore be chosen:

1. keep Φ fixed, i.e. fix the interface position
2. redistribute Ψ to $\hat{\Psi}$ by:
   (a) define a vector field $\mathbf{q}$ which points towards the interface,
   (b) with $\mathbf{q}$, use a PDE to transport ϵ towards the interface,
3. with $\hat{\Psi}$ compute Φ in the usual fashion by the 'inverse function' (see previous section),
4. repeat the procedure (with the updated interface position) when still unwanted values of $\hat{\Psi}$ according to $\hat{\Phi}$ exist.

2a velocity field q:

A velocity field $\mathbf{q}(\mathbf{x})$ is needed, which transports ('fluxes') the errors $\epsilon(\mathbf{x})$ towards the interface. In other words, it has to point towards the interface. Such a velocity field can be defined by taking $\mathbf{q} = (u, v, w)^t$:

$$u_{i+\frac{1}{2}} = \begin{cases} 1, & \theta_{i+\frac{1}{2}} < \frac{1}{2} \\ 0, & \theta_{i+\frac{1}{2}} = \frac{1}{2} \\ -1, & \theta_{i+\frac{1}{2}} > \frac{1}{2}, \end{cases} \tag{47}$$

where $\theta_{i+\frac{1}{2}}$ is the relative distance of the interface between nodes i and $i+1$, measured from node i:

$$\theta_{i+\frac{1}{2}} = \frac{-\Phi_i}{\Phi_{i+1} - \Phi_i}. \tag{48}$$

The magnitude of the components of $\mathbf{q}$ are 1 for reasons of computational efficiency. After some mathematical manipulations, u is found as:

$$u_{i+\frac{1}{2}} = -\text{sign}(|\Phi_{i+1}| - |\Phi_i|). \tag{49}$$

and similar in the other coordinate directions.

Problems arise when $|\Phi_i| \approx |\Phi_{i+1}|$, since numerical errors in Φ will cause a flip-flop effect. Therefore:

$$u_{i+\frac{1}{2}} = 0, \qquad ||\Phi_{i+1}| - |\Phi_i|| < \text{tol } \max(|\Phi_{i+\frac{1}{2}}|, h), \tag{50}$$

where $\Phi_{i+\frac{1}{2}}$ is approximated by $\frac{1}{2}(\Phi_i + \Phi_{i+1})$, h is the mesh size and tol is a tolerance, typically 0.1.

2b transport of ϵ:

With velocity field $\mathbf{q}(\mathbf{x})$, the errors are transported by (for some artificial time τ):

$$\frac{\partial \epsilon}{\partial \tau} + \nabla \cdot (\mathbf{q}\epsilon) = 0, \tag{51}$$

until steady state ($\frac{\partial \epsilon}{\partial \tau} = 0$). Eqn. (51) is discretized by an upwind approximation in the usual fashion. Instead of taking $\tau \to \infty$, which is rather impractical, after each iteration of Eqn. (51) the number of cells (num) with $|\epsilon| >$ tol is computed. The iterations are stopped if

$$\left\| \frac{\partial \epsilon}{\partial \tau} \right\|_\infty \leq \text{tol } \vee \text{ num} = 0 \ \vee \ \text{iter} > \text{maxiter}, \tag{52}$$

where tol is a tolerance, typically 10^{-2}, iter is the iteration-number and maxiter is the maximum number of iterations. The following observations are made:

- the number of cells containing unwanted values of Ψ is small, typically less than 10 for a 64^3 grid
- the number of iterations of Eqn. (51) is small, typically less than 5
- the number of Picard iterations is small, typically 1.

5 Re-initialization

Since the interface forces are regularized near the interface, it is necessary that the Level-Set function is a distance function at all time instances. This ensures that the regularization width has finite thickness at all time. This is achieved by the re-initialization procedure of Sussman et al. [13]. The Level-Set function Φ is re-initialized to make it a distance function by solving until steady state for artificial time t'

$$\frac{\partial \Phi}{\partial t'} = N(\Phi, \Phi^0), \tag{53}$$

with initial conditions

$$\Phi(\mathbf{x}, 0) = \Phi^0(\mathbf{x}). \tag{54}$$

Here Φ^0 defines the interface position after the application of Eqn. (15), and $N(\Phi, \Phi^0)$ is defined by $\text{sign}(\Phi^0)(1 - \|\nabla\Phi\|)$. The discretization of Eqn. (53) is:

$$\frac{\Phi^{k+1} - \Phi^k}{\Delta t'} = N_h(\Phi^k, \Phi^0). \tag{55}$$

Re-initialization has been applied in the two-dimensional work (Van der Pijl et al. [1, 11]). The extension to three dimensions is straightforward.

Due to re-initialization, the interface position can shift enormously (see Van der Pijl et al. [1]: 2D advection tests). This is unwanted, since the MCLS method is based on finding *small corrections* to Φ in order to conserve mass. Especially small droplets or bubbles (about one mesh width) could completely disappear due to re-initialization.

Example Consider a two-dimensional square domain $(x, y) \in [-\frac{1}{2}, \frac{1}{2}]^2$. There are two interfaces: a straight line at $x = -\frac{1}{4}$ and a circle with the origin at $(x, y) = (0, 0)$ and a diameter of 4 mesh widths. The Level-Set function is initialized such that it is a distance function (see Fig. 5(a)). The corresponding Volume-of-Fluid function Ψ is depicted in Fig. 5(b). The Level-Set function after re-initialization is shown in Fig. 5(c). Although the initial Level-Set function is a distance function, large differences exist between the initial and re-initialized Level-Set functions near the circular interface. In fact, the circle has completely disappeared. Consequently, when correction are made to the Level-Set function in order to satisfy mass (prescribed by the original Volume-of-Fluid function) the corrections are large (see Fig. 5(d)). The resulting Level-Set function is highly non-smooth, which is an unwanted effect in our approach.

5.1 Modified re-initialization

As the previous example has shown, situations exist where the re-initialization procedure of Sussman et al. [13] does not satisfy the needs of the MCLS approach. Various improvements of the original re-initialization procedure exist, often combined with higher order discretization. Since we make a correction after re-initialization in order to conserve mass,

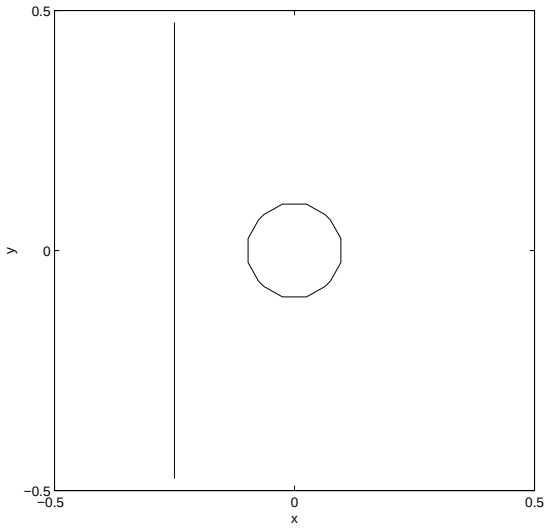


Figure 4: Prescribed interfaces on a 20×20 mesh

it is not required that the interface maintains the initial position exactly. Besides that, we use the fact that the initial data already resembles a distance function. This leads to a simple modification of the original procedure that satisfies the needs of the MCLS approach.

The difficulty with re-initialization is that although it is wanted that that interface position remains intact, no ($\Phi = 0$) boundary conditions are imposed at the interface. The interface is only defined in the initial conditions. The idea is now to leave the values of Φ near the interface ($\Phi = 0$) unchanged if $-b \leq \Phi \leq b$, where $b \geq 0$ is a prescribed band-width, proportional to the mesh width. This is achieved by modification of Eqn. (55):

$$\frac{\Phi^{k+1} - \Phi^k}{\Delta t'} = N_h(\Phi^k, \Phi^0)(1 - q(\Phi^0)) + \frac{\Phi^0 - \Phi^k}{\Delta t'} q(\Phi^0), \tag{56}$$

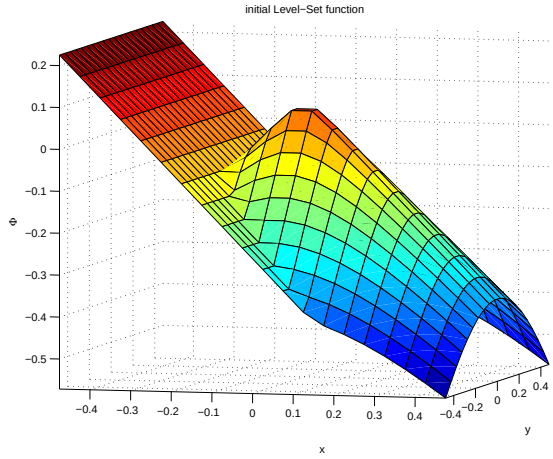
where q is a function, which has value 1 near the interface and value 0 elsewhere:

$$q(\Phi) = \begin{cases} 1 & |\Phi| \leq b \\ 0 & |\Phi| > b. \end{cases} \tag{57}$$

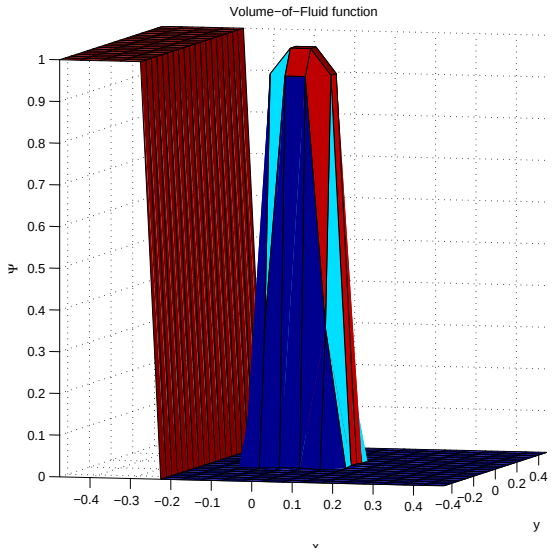
Note that with this approach effectively *boundary conditions* are applied at the ($\Phi^0(\mathbf{x}) = -b$) and ($\Phi^0(\mathbf{x}) = b$) contours respectively. The disadvantage of this approach is that Φ can become non-smooth at $|\Phi| = b$. Therefore, a smooth function f is chosen which mimics the step-wise behavior:

$$q(\Phi^0) = \exp(-(\frac{\Phi^0}{\alpha})^2), \tag{58}$$

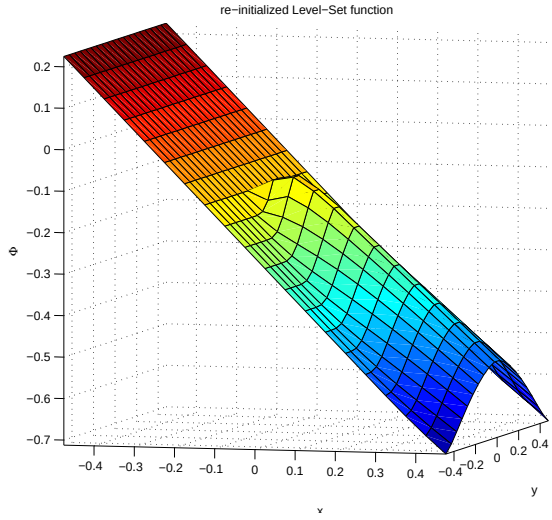
where $\alpha = \sqrt{\frac{2}{3}(\Delta x^2 + \Delta y^2 + \Delta z^2)}$ is a constant.



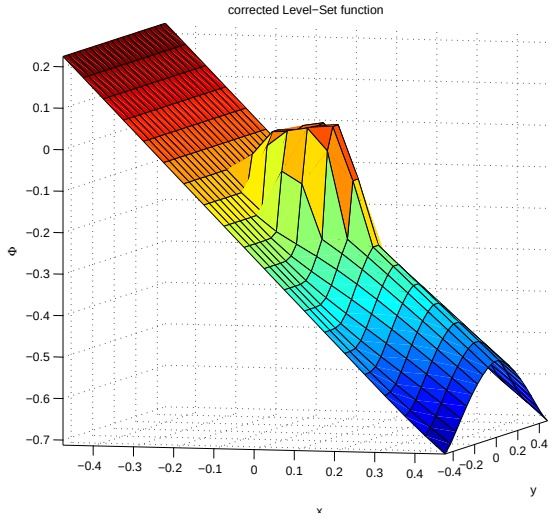
(a) Initial Level-Set function



(b) corresponding Volume-of-Fluid function

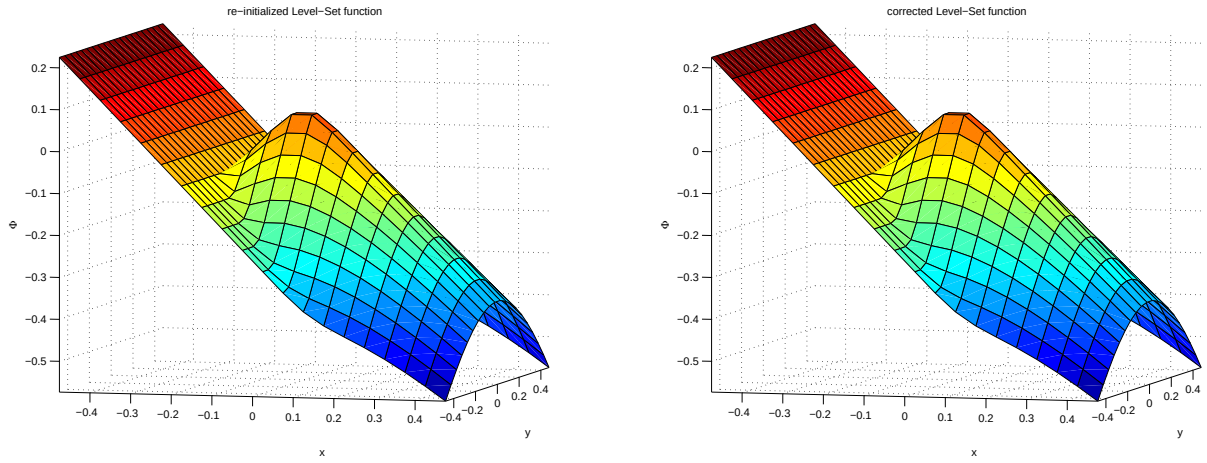


(c) Level-Set function after re-initialization



(d) corrected Level-Set function

Figure 5: Effect of re-initialization on small entities



(a) Level-Set function after modified re-initialization

(b) corrected Level-Set function

Figure 6: Effect of modified re-initialization on small entities

Stop criterion The iterations of Eqn. (56) are stopped if

$$\left\| \frac{\Phi^{k+1} - \Phi^k}{\Delta t'} \right\|_\infty \leq \text{tol}, \tag{59}$$

where tol is a tolerance. Note that this quantity acts as the residual of

$$N_h(\Phi, \Phi^0)(1 - f(\Phi^0)) + \frac{\Phi^0 - \Phi}{\Delta t'} f(\Phi^0) = 0, \tag{60}$$

so

$$\begin{cases} \|N_h(\Phi, \Phi^0)\|_\infty \leq \text{tol}, & f(\Phi^0) = 0 \\ \|\Phi^0 - \Phi\|_\infty \leq \text{tol}\ \Delta t', & f(\Phi^0) = 1. \end{cases} \tag{61}$$

We take tol $= 0.1$.

Example The effects of the modification of the re-initialization procedure on the previous described test-case is shown in Fig. 6. It can be seen that the re-initialized Level-Set function resembles the initial Level-Set function. The same holds for the corrected Level-Set function, which was the objective of the modification of the re-initialization.

6 Surface tension

For the modeling of surface tension effects on the fluid motion, the CSF method of Brackbill et al. [16] is adopted. With this method the interface forces are transformed into volume forces and added to the right-hand side of Eqn. (1). Because of their Dirac delta function character, these surface tension forces are regularized, such that they are distributed over

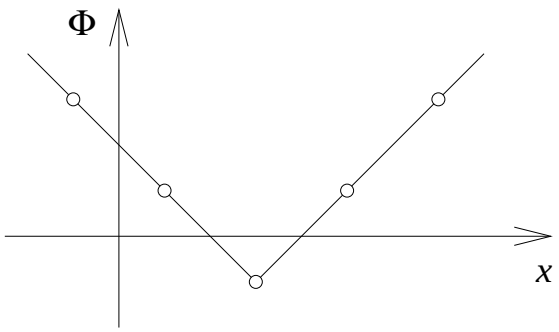


Figure 7: local extremum of the Level-Set function Φ between two approaching interfaces

a small region around the interface. Sussman et al.[13] showed that these forces in case of the Level-Set methodology are

$$\frac{1}{\rho}\sigma\kappa\delta_\alpha(\Phi)\nabla\Phi,$$

where σ is the surface tension coefficient, κ is the interface curvature and δ_α is the smoothed delta function. More details on surface tension in our two-dimensional work can be found in [1]. The extension to three-dimensional space is straightforward.

6.1 Discretization of curvature

The curvature κ is defined by

$$\kappa = \nabla \cdot \frac{\nabla\Phi}{|\nabla\Phi|}. \tag{62}$$

For the ease of implementation, this is rewritten as (see e.g. Brackbill et al. [16], Rudman [22], Kang et al. [14]):

$$\kappa = \frac{\nabla\cdot\nabla\Phi}{|\nabla\Phi|} - \frac{\nabla\Phi\cdot[(\nabla\Phi\cdot\nabla)\nabla\Phi]}{|\nabla\Phi|^3}, \tag{63}$$

which is a symbolic notation for:

$$\kappa = \frac{\nabla\cdot\nabla\Phi}{|\nabla\Phi|} - \frac{1}{|\nabla\Phi|^3}\sum_i\sum_j \frac{\partial\Phi}{\partial x_i}\frac{\partial\Phi}{\partial x_j}\frac{\partial^2\Phi}{\partial x_i\partial x_j}. \tag{64}$$

The second order derivatives are discretized by straightforward central differencing. The first order derivatives however need special care. It is immediately visible that problems arise in Eqn. (64) when $|\nabla\Phi|$ vanishes. Such a situation occurs when the Level-Set function has a local extremum. This happens when two interfaces are approaching each other, see Fig. 7 for an example. To overcome this, instead of using central differences, the first order derivatives are discretized by taking the maximum (in modulus) of the left-hand-side, right-hand-side and central scheme. This is a first order approximation. It is important when droplets or bubbles become small and when droplets or bubbles merge. It also increases the smoothness of the curvature distribution.

6.2 Parasitic currents

The difficulty with the CSF method is that so-called parasitic currents can occur. These are small unphysical currents that according to Lafaurie et al. [23] are due to 'slight unbalance of stresses at the sites in the interfacial region'. Parasitic currents have been studied for a Volume-of-Fluid method by Scardovelli and Zaleski in [24] and by Lafauri et al. in [23]. Tryggvason et al. have analyzed parasitic currents in [25] for a front-tracking method and Enright et al. in [26] for a Level-Set method.

The parasitic currents are caused by the distribution of interface forces and by inaccuracies in the computation of the curvature. In case of the Level-Set methodology, the curvature is given by Eqn. (64). Computing the curvature with pure Level-Set methods faces no problems because of the smoothness of the Level-Set function. However, with the MCLS method additional difficulties are encountered. With the MCLS approach corrections to the Level-Set function are made *locally* near the interface in order to conserve mass. This results in irregular data near the interface. From Eqn. (64) follows that it will cause local errors in κ of the order $\frac{\epsilon}{h^2}$, where ϵ is the magnitude of the corrections and h is the mesh width. Although the corrections are small in general, situations might occur where they are $\mathcal{O}(h)$ locally. See Fig. 5(d) for an example with unmodified re-initialization. Note that in this case the errors are completely due to re-initialization, since no advection is performed in this example. The consequences for the curvature are rather dramatic in the sense that parasitic currents will grow when the mesh is refined. The order of the errors in κ is $\frac{1}{h}$. A study of this phenomenon is performed by Coyajee et al. in [27].

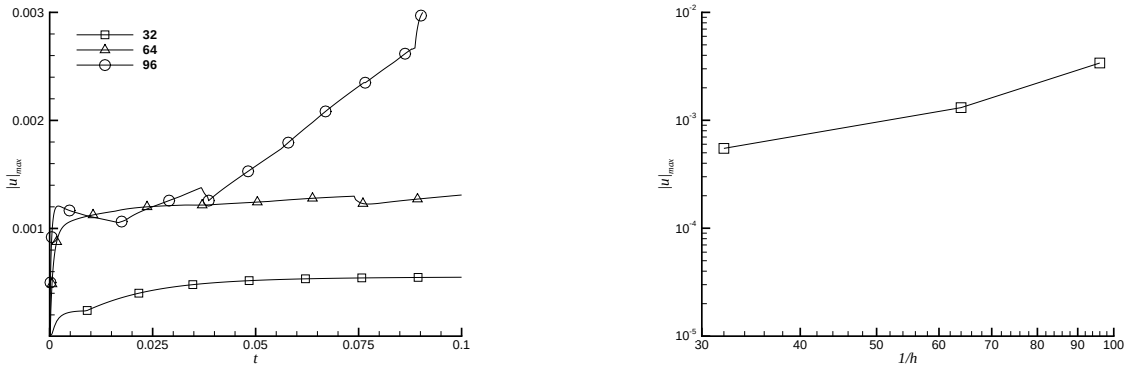
The Laplace test-case is frequently used to study the parasitic currents, see e.g. Rudman [22], Lafaurie et al. [23], Tryggvason et al. [25]. A sphere with radius $\frac{1}{4}m$ is placed in the center of a cubic domain with dimensions $1 \times 1 \times 1m$. The flow is initially at rest. Since the initial conditions satisfy the steady-state problem, all velocities are parasitic currents. We take the following gravity and material constants: $g = 0\ \frac{m}{s^2}$, $\sigma = 0.01\ \frac{kg}{s^2}$, $\rho_0 = 1\ \frac{kg}{m^3}$, $\frac{\rho_1}{\rho_0} = 1$, $\mu_0 = 0.1\ \frac{kg}{ms}$ and $\frac{\mu_1}{\mu_0} = 1$. In Fig. 8 the time evolution of parasitic currents are shown for three different grid sizes: 32^3, 64^3 and 96^3. The maximum values (in modulus) of the parasitic currents grow in time and become larger for decreasing step size. The grid dependency is $\mathcal{O}\left(\frac{1}{h}\right)$. The currents are shown in Fig. 9

Our aim is now to reduce the parasitic currents to a small value that is negligible compared to the physical currents and does not increase when the mesh is refined. This is achieved by:

- reduction of the magnitude of the corrections caused by re-initialization
- smoothing of the curvature

6.3 Re-initialization start criterion

The re-initialization procedure introduces errors in the Level-Set function Φ. It is unwanted that these errors are larger than the changes per time-step due to advection of Φ (by means



(a) time evolution

(b) grid dependency at $t = 0.1$

Figure 8: Parasitic currents for the Laplace test-case with three different grids

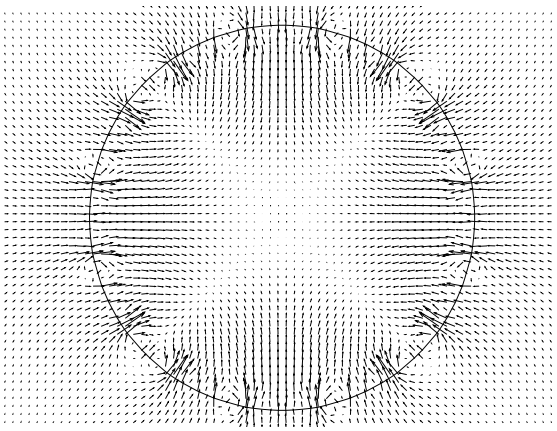
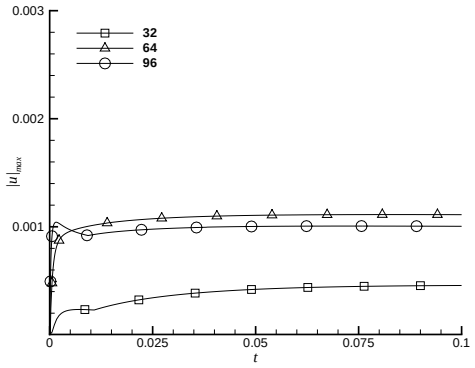
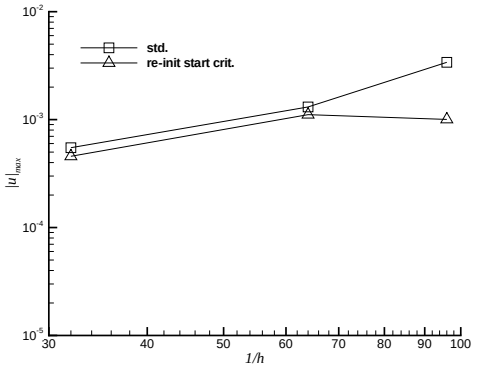


Figure 9: Parasitic currents for the Laplace test-case in the symmetry plane; 96^3 grid



(a) time evolution; The maximum modulus of the parasitic currents converge to a constant value for all grids



(b) grid dependency at $t = 0.1$; Note that the case without the start criterion ('std') is also include in this figure; The parasitic currents decrease due to the start criterion, most notably for the 96^3 grid

Figure 10: Parasitic currents for the Laplace test-case with three different grids; start criterion in re-initialization

of Eqn. (15)). Therefore the Level-Set function is not re-initialized each time-step but only when the interface has traveled some distance. This is measured by a time-step Δt_{reinit}:

$$\Delta t_{\text{reinit}} = \frac{\sigma}{\frac{||u||_\infty}{\Delta x} + \frac{||v||_\infty}{\Delta y} + \frac{||w||_\infty}{\Delta z}}, \tag{65}$$

where $\sigma = 0.1$ is chosen. Re-initialization is only performed if $t^n - \Delta t_{\text{reinit}}$ is larger than or equal to the last re-initialization time. The effect of the time-step criterion on the parasitic currents is shown in Fig. 10. The maximum modulus of the parasitic currents converge to a constant value for all grids. The positive effect on the 96^3 grid is most significant.

6.4 Effect of mass redistribution

The previous figures showed that the velocities (parasitic currents) are very small for the Laplace test case. This means that the displacement of the interface is equally small. It can therefore be expected that the Level-Set function Φ^* after pure Level-Set advection (Eqn. (15)) and without re-initialization does not loose a significant amount of mass. At first sight, there would be no need for any corrections on Φ^* in order to conserve mass. However Fig. 10 shows that the parasitic currents do not decrease when the grid is refined, which indicates that still corrections are made that influence the computation of curvature by means of Eqn. (64). Further inspection of the numerical output reveals that the corrections are due to mass redistribution. This means that 'numerical vapor' exists in the initial conditions according to the mass-distribution criterion as specified by Eqn. (44). In other words, a situation exists where the Level-Set function Φ in a grid point $\mathbf{x}_{i,j,k}$ has the

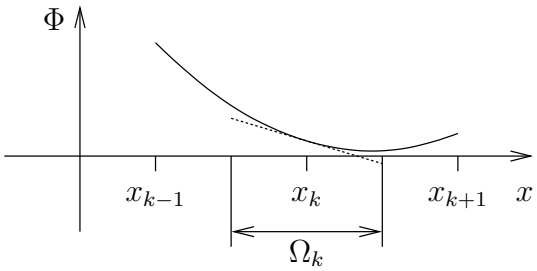


Figure 11: One-dimensional example of a Level-Set function Φ that does not change sign in Ω_k (hence no interface); but the linearization of Φ around $\mathbf{x}_k$ does, hence $0 < \Psi_k < 1$ and the classification 'numerical vapor'.

same sign as Φ in its neighbors $\mathbf{x}_{i\pm1,j,k}$, $\mathbf{x}_{i,j\pm1,k}$ and $\mathbf{x}_{i,j,k\pm1}$, but still an interface exists in the computational cell Ω_k according to the linearization of Φ around $\mathbf{x}_k$. A one-dimensional example is shown in Fig. 11. The mass redistribution will redistribute all mass in $\Omega_{i,j,k}$ in the direction of the nearest interface and a new Volume-of-Fluid function Ψ is obtained. With the new Volume-of-Fluid function Ψ, corrections are made to the Level-Set function Φ such that it satisfies $f(\Phi, \nabla\Phi) = \Psi$, as explained in Section 4.3.

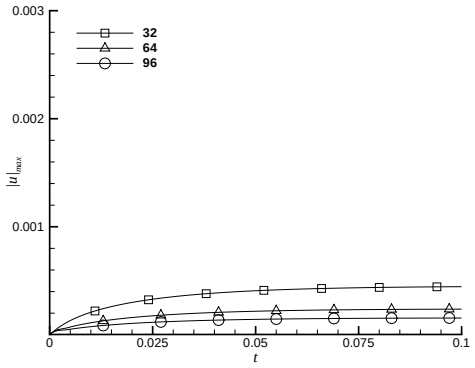
Fig. 12 shows the parasitic currents without mass redistribution. No corrections to Φ^* are made and the pure Level-Set approach is obtained. The parasitic currents are completely due to the regularization of the interface forces (CSF approach) and converge with $\mathcal{O}(h)$. This is the pure Level-Set behavior and shows that the parasitic currents are due to mass redistribution.

The Laplace test case is believed to be the worst-case scenario for the 'Mass Conserving Level-Set' approach. On the one hand corrections are made due to mass redistribution and the initial Level-Set function becomes perturbed. On the other hand the interface does not move and the perturbations remain intact since $\frac{\partial\Phi}{\partial t} = -\mathbf{u}\cdot\nabla\Phi \approx 0$. This means that the perturbations will never disappear. When the interface *does* move, the perturbations will (numerically) diffuse after the convection step (Eqn. (15)). It is then advantageous to compute the curvature before the corrections are made. Once again, this makes no difference for the Laplace test case since the interface is stationary.

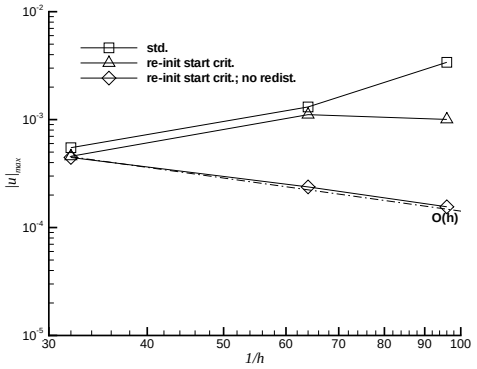
6.5 Curvature smoothing

Volume-of-Fluid methods encounter similar, but more severe difficulties when computing the curvature, since the Volume-of-Fluid function has a step-like behavior. With these methods it is common fashion to effectively compute the curvature based on a smoothed function, see e.g. Scardovelli and Zaleski [24] or Rudman [22]. With front-tracking methods Sousa et al. [28] remove 'undulations' from the interface before computing the curvature. Doing so, they effectively smooth the interface grid. Both approaches have been an inspiration for this work.

The first step that comes to mind would be to smooth the Level-Set function before



(a) time evolution; The maximum modulus of the parasitic currents converge to a constant value for all grids



(b) grid dependency at $t = 0.1$; Note that the two previous cases ('std' and 're-init start crit.') are also included in this figure

Figure 12: Parasitic currents for the Laplace test-case with three different grids; start criterion in re-initialization and no mass redistribution

computing the curvature. However, a smooth distribution of the *curvature* is sought, not of the Level-Set function itself. Therefore, it is chosen to smooth the curvature instead. This should not be performed carelessly, since some properties of the curvature should be maintained. These are:

1. the curvature is continuous and sufficiently smooth,
2. the curvature κ is the divergence of $\dfrac{\nabla\Phi}{|\nabla\Phi|}$, so κ has to be modified conservatively,
3. the curvature has to correspond to the interface at the interface ($\Phi = 0$) when the curvature is already sufficiently smooth.

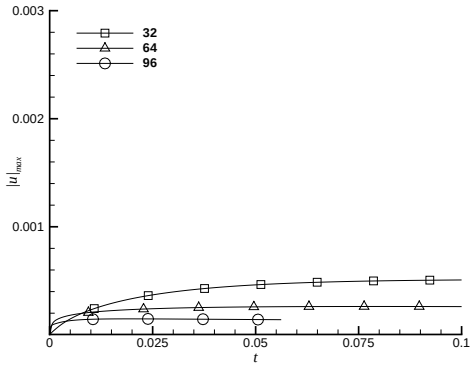
To meet these demands, an approach based on a diffusion equation in pseudo-time τ is chosen:

$$\frac{\partial\kappa}{\partial\tau} = \nabla \cdot d\nabla\kappa. \tag{66}$$

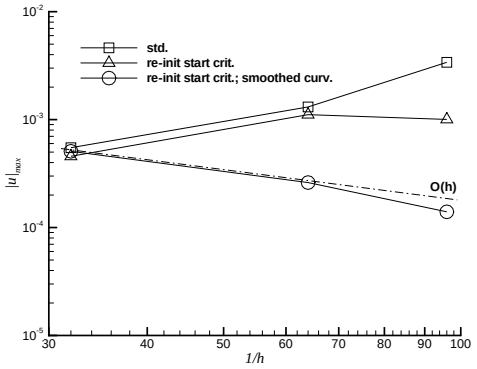
The first two requirements are satisfied if Eqn. (66) is used provided d is smooth. Here d is a diffusion coefficient that ensures that the last demand is also met. It is constructed in the following way:

$$d = 1 - \exp\left(\frac{-\Phi^2}{\Delta x^2 + \Delta y^2 + \Delta z^2}\right). \tag{67}$$

Note that d is smooth. Since the diffusion coefficient is very small near the interface, significant modifications to κ are only made when the derivatives $\nabla\kappa$ are large and κ is left intact when it is smooth. An explicit Euler temporal discretization is used. The maximum



(a) time evolution; The maximum modulus of the parasitic currents converge to a constant value for all grids



(b) grid dependency at $t = 0.1$; Note that the two previous cases ('std' and 're-init start crit.') are also included in this figure; The parasitic currents decrease further due to smoothing of the curvature

Figure 13: Parasitic currents for the Laplace test-case with three different grids; time-step criterion in re-initialization and smoothed curvature

allowed pseudo-time step (for $d \leq 1$) of $\frac{h^2}{6}$ is used. For the time-step of the explicit Euler method $\frac{N}{16}$ is used, where N is the number of grid cells in one direction. Note that the number of iterations is proportional to the mesh size.

The results for the Laplace test-case are shown in Fig. 13. The maximum value (in modulus) of the parasitic currents converges with $\mathcal{O}(h)$ as shown in Fig. 13(a).

7 Applications

The behavior of the MCLS approach is shown by an advection test with a prescribed velocity field. Thereafter, the method is applied to the complete set of equations by considering a falling drop and a rising bubble in two and three dimensions, respectively. Merging of rising bubbles is studied for two aligned and two misaligned bubbles.

7.1 Advection test

Consider a sphere which is placed in the center of a cube. The width of the cube is twice the diameter of the sphere. There are 30 grid cells in x, y and z-direction. Periodic boundary conditions are applied at the boundaries. The CFL number is 0.35. Note that this CFL number is also used in the full model simulations. The sphere is advected with uniform velocity field $(u, v, w) = (2, 3, -1)$. Relative mass errors are plotted in Fig. 14. The results are consistent with the work in [1].

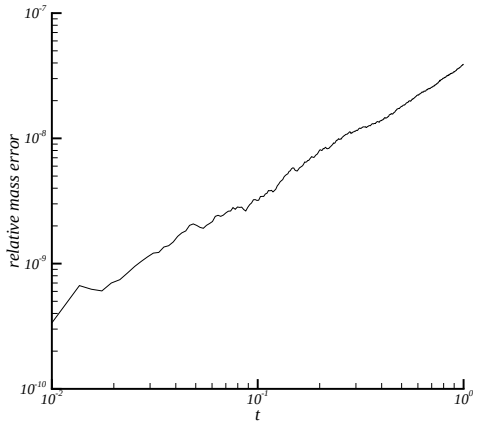


Figure 14: Relative mass errors for advected sphere; $\epsilon = 10^{-8}$

7.2 Rising air bubble in water without surface tension

The behavior of the MCLS approach is shown by a rising bubble in three dimensions. The gravity and material constants are:

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

where subscripts $_w$ and $_a$ indicate water and air respectively. An air bubble is initially at rest and aligned on the center line of the computational domain. The dimensions of the domain are: $L_x = L_y = L_z = 0.01\ m$. The radius of the bubble is $R = 0.00125\ m$. The distance from the center of the bubble to the floor of the domain is $z_0 = 0.0025\ m$. A free surface is located at $z_1 = 0.0075\ m$. Results on a $96 \times 96 \times 96$ grid are shown in Fig. 15. The snapshots are taken at equal time differences of 0.005 *sec*. For the ease of visualization, only $y < \frac{1}{2}\ L_y$ is plotted. Also, the interface position in the plane $y = \frac{1}{2}\ L_y$ is plotted. It can be seen that the bubble deforms and breaks up to form a bell- and ring-like structure, just before it breaks through the free surface. In [3] similar simulations are made for a rising bubble with low surface tension. Good resemblance is observed with these results and the snapshots presented in Fig. 16.

In Fig. 17 the rise speed w_c of the droplet before collision with the free surface is plotted as function of time t. For the rise speed w_c the velocity of the center of gravity of the bubble

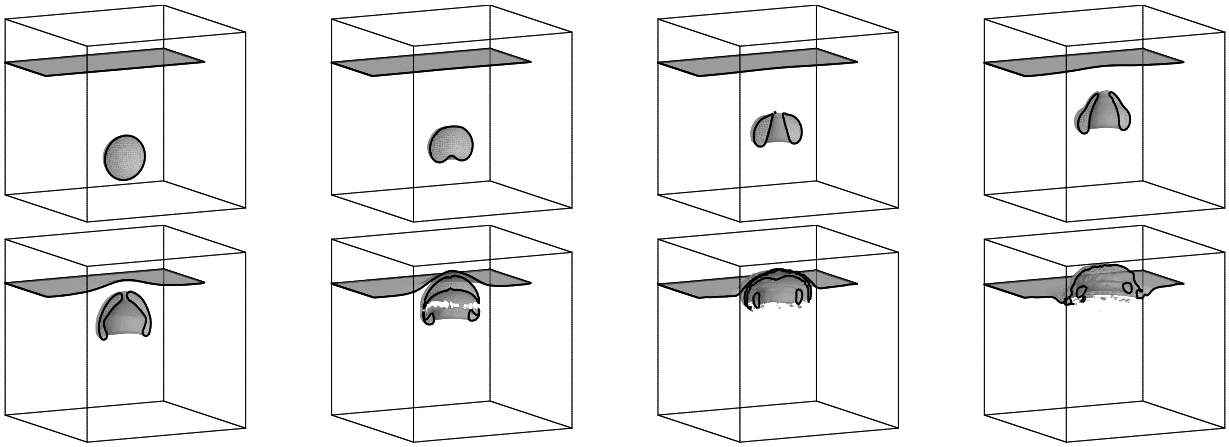


Figure 15: Rising bubble without surface tension; 96^3 grid

is taken:

$$w_c = \frac{\int\limits_{\overline{\Omega}} (1 - H(\Phi)) w \, \mathrm{d}\Omega}{\int\limits_{\overline{\Omega}} (1 - H(\Phi)) \, \mathrm{d}\Omega}, \tag{68}$$

where H is the Heaviside step function and the integration is over $\overline{\Omega}$, which is the part of the domain Ω that does not include the air above the free surface. Note that by definition

$$\Phi = \begin{cases} < 0, & \text{air}, \\ > 0, & \text{water}, \end{cases} \tag{69}$$

so that $1 - H(\Phi)$ is 1 in air and 0 in water. Using the Volume-of-Fluid function Eqn. (18) in computational cell Ω_k

$$\Psi_k = \frac{1}{\mathrm{vol}(\Omega_k)} \int\limits_{\Omega_k} H(\Phi) \, \mathrm{d}\Omega, , \tag{70}$$

w_c is approximated as

$$w_c = \frac{\sum\limits_k (1 - \Psi_k) w_k}{\sum\limits_k 1 - \Psi_k}, \tag{71}$$

where the summation is over all computational cells in $\overline{\Omega}$. The rise speed is in good agreement for both grids. The rise speed reaches a maximum just after the bubble starts moving. A local maximum is reached again before impact with the free surface in top of the domain.

The shape of the bubble just before it breaks through the free surface is depicted in Fig. 18. The bubble is transformed into a spherical cap with radius $R_s = 0.0018\ m$. In [29] an expression for the rise speed of a spherical cap, based on irrotational flow, is derived:

$$w_s = \frac{2}{3}\sqrt{gR_s} = 0.0885 m/s. \tag{72}$$

(a) $t = 0.009\ s$ (b) $t = 0.01\ s$ (c) $t = 0.013\ s$ (d) $t = 0.015\ s$

Figure 16: Rising bubble without surface tension; 96^3 grid; zoomed in

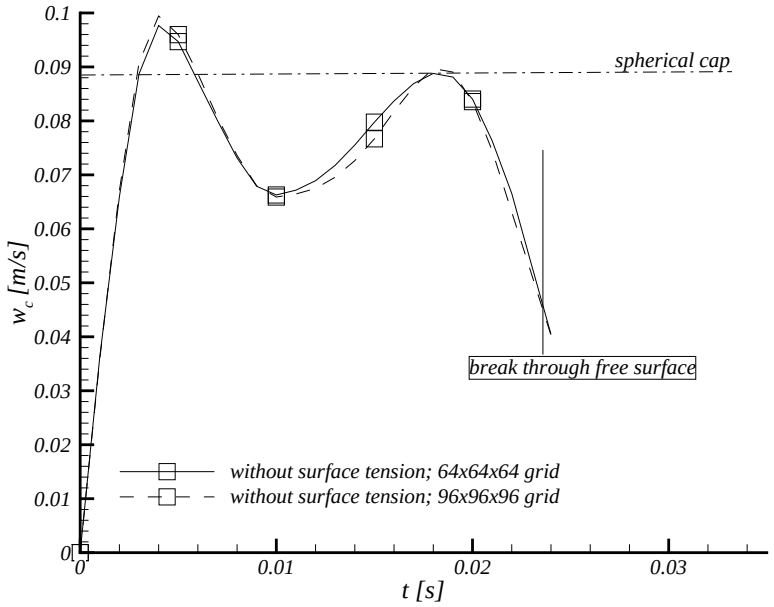


Figure 17: Rise speed of the bubble without surface tension; the dotted line is the analytic rise speed of the spherical cap

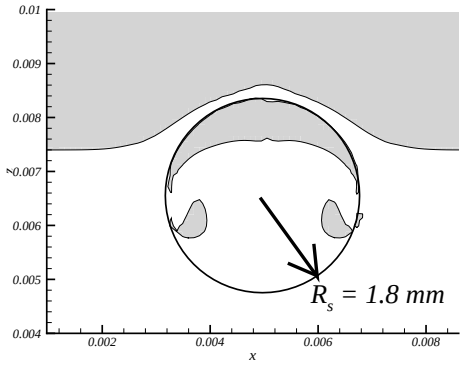


Figure 18: Bubble without surface tension just before it breaks through the free surface; symmetry plane $y = \frac{1}{2}L_y$; R_s is the radius of the spherical cap.

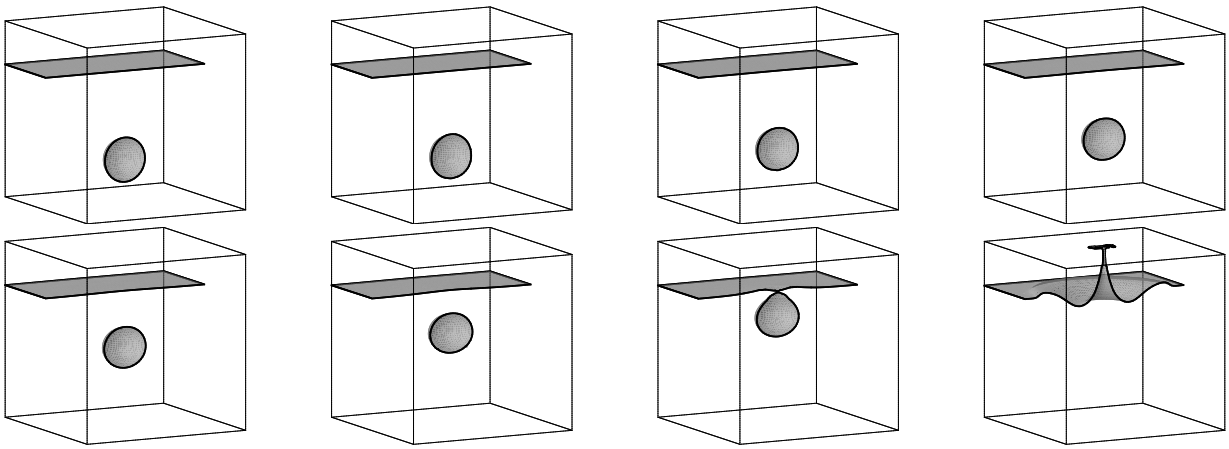


Figure 19: Rising bubble with surface tension; 96^3 grid

This speed is indicated in Fig. 17. The numerical results compare well with this speed until the bubble breaks through the free surface.

7.3 Rising air bubble in water with surface tension

In Fig. 19 results with surface tension are presented. The deformation of the bubbles is significantly reduced due to the surface tension effects. In Fig. 20 the rise speed w_c of the bubble with surface tension is plotted for both grids. As a reference the rise speed of a bubble without surface tension (previous test case) is also included in this figure. The agreement between both grids is good. When the bubble is just released, the acceleration of the bubble with surface tension is lower than the bubble without surface tension. But instead of reaching a maximum rise speed, the bubble with surface tension continuous to

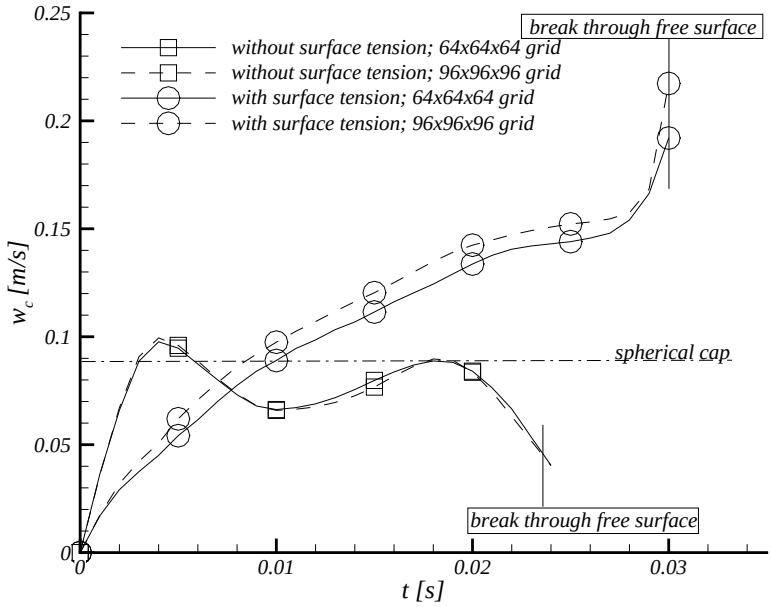


Figure 20: Rise speed of the bubble with surface tension; marks are at equally spaced time intervals of 0.005 s.

accelerate further. Eventually higher speeds are reached for the bubble with surface tension than without surface tension. A peak is observed just before collision with the free surface in top of the domain. This indicates that the bubble is sucked towards the free-surface before merging with it.

7.4 Falling water droplet in air without surface tension

A falling water droplet is considered next. The gravity and material constants are the same as for the bubble. The dimensions of the domain are also the same as for the rising-bubble test case. The droplet is released at half the height of the domain, i.e. $z_0 = 0.005\ m$ and the free surface is initially located at $1/4^{th}$ height: $z_1 = 0.0025\ m$.The radius of the droplet is $R = 0.00125\ m$. Results are shown in Fig. 21. The snapshots are taken at intervals of 0.01 *sec*. The droplet accelerates after it is released and hits the free surface. A jet appears after collision that reaches up to the ceiling of the domain.

In Fig. 22 the fall speed w_c of the droplet before impact with the free surface is plotted as function of time t. For the fall speed w_c the velocity of the center of gravity of the droplet is taken, which is computed in a similar fashion as the rise speed of a bubble; Eqn.

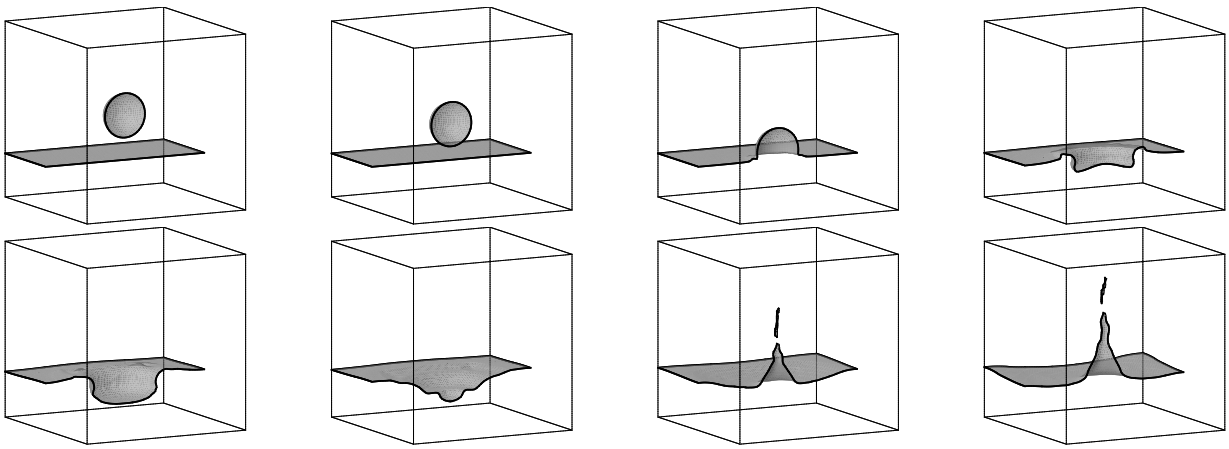


Figure 21: Falling droplet without surface tension; 96^3 grid

(71) becomes

$$w_c = \frac{\sum_k \Psi_k \mathbf{x}_k}{\sum_k \Psi_k}. \tag{73}$$

Good agreement of the fall speed with the free-fall velocity $w_c = -gt$ is observed.

7.5 Falling water droplet in air with surface tension

The release height of a droplet with surface tension is increased to $z_0 = 0.0075\ m$. The fall height of the droplet without surface tension is smaller because otherwise the droplet becomes too deformed when it hits the free surface. The deformation of the droplet with surface tension is much less and allows a higher fall height. In Fig. 23 results are presented. The droplet indeed does not change shape before it collides with the free surface. Although the release height of the droplet is larger than without surface tension, the jet after collision is considerably smaller. This is the effect of surface tension that tries to minimize the interfacial area. In Fig. 24 the falling speed is compared with the free fall velocity $w_c = -gt$. The computed fall speeds for both grids are in agreement and larger than the free-fall velocity. Since the droplet without surface tension accelerated with $\frac{\mathrm{d}w_c}{\mathrm{d}t} = -g$ (see previous test case) the acceleration with surface tension is higher than without surface tension. This is believed to be caused by the CSF approach. The total force acting on a bubble or droplet due to surface tension should vanish, since the surface tension forces are internal forces and the interface is closed. However with the CSF approach a non-conservative expression is obtained. This means that numerically the forces do not necessarily add up to zero any longer. Inspection of the numerical data shows that

$$\sum_k \frac{1}{\rho} \mathbf{f}_s \approx -10^{-8}. \tag{74}$$

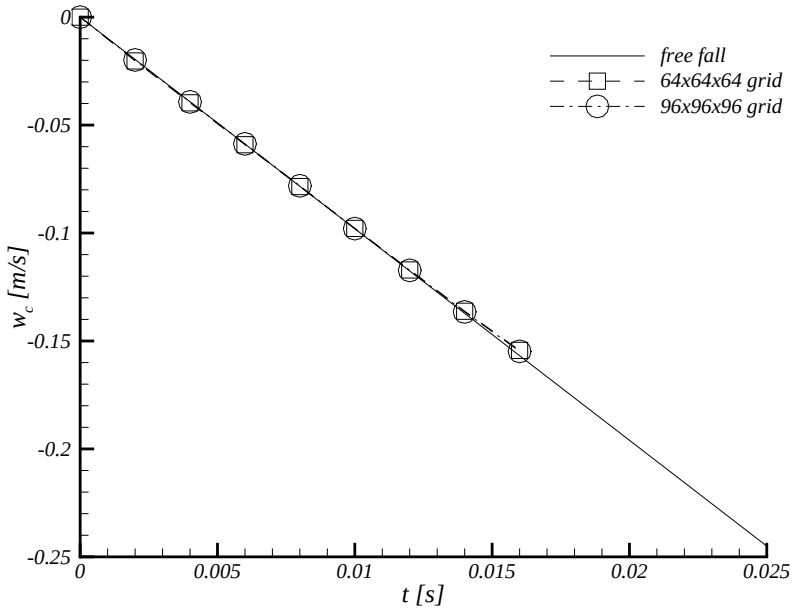


Figure 22: Fall speed of the droplet without surface tension compared with the free-fall velocity

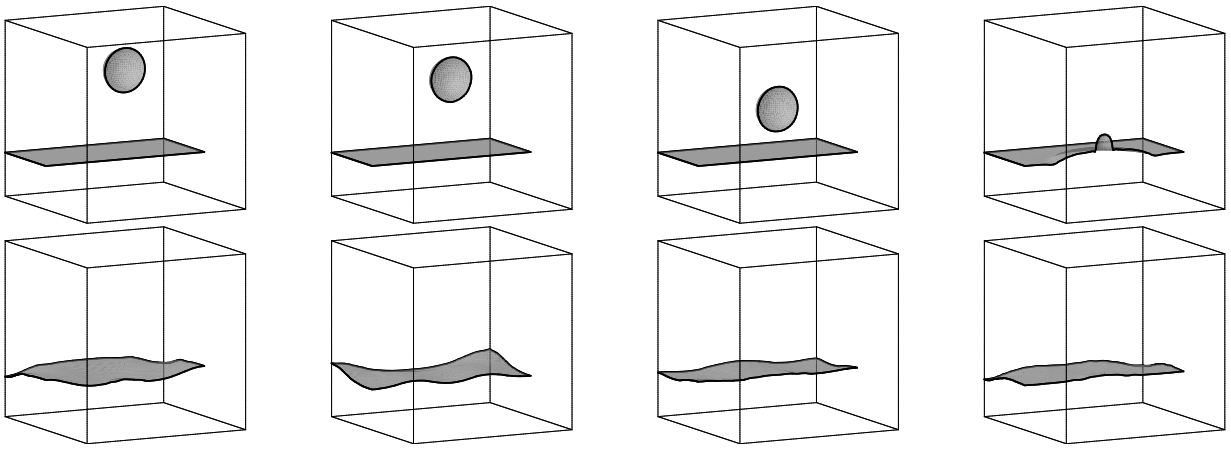


Figure 23: Falling droplet with surface tension; 96^3 grid

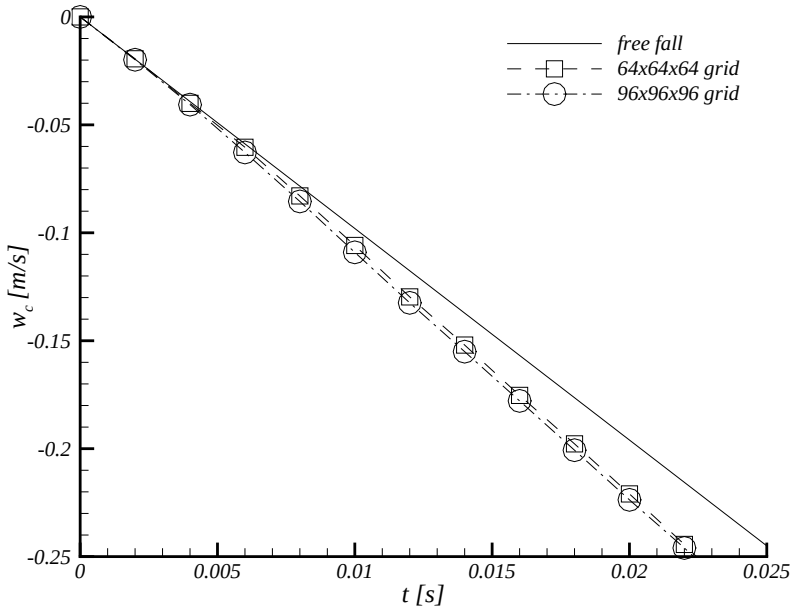


Figure 24: Fall speed of the droplet with surface tension compared with the free-fall velocity

The acceleration of the center of gravity of the droplet will therefore be

$$g_z + \frac{1}{\frac{4}{3}\pi R^3} \sum_k \frac{1}{\rho} \mathbf{f}_s \approx -11 m/s^2, \tag{75}$$

where $\frac{4}{3}\pi R^3$ is the volume of the droplet. Note that g_z is the z-component of the gravity vector $\gg= (g_x, g_y, g_z)^t$. This is in agreement with the results in Fig. 24, where the droplet indeed accelerates with approximately $-11\ m/s^2$.

7.6 Comparison of rising bubble with experimental data

In [30] bubbly flows are simulated with a front tracking and a Volume-of-Fluid method. The rise speed is compared with experimental closure relations. Assume that '1' corresponds to the heavier fluid outside the bubble and '0' to the light fluid inside the bubble. The flow is characterized by the following dimensionless parameters:

$$\begin{aligned} \text{Eö} &= \frac{g(\rho_1 - \rho_0)(2R)^2}{\sigma} \\ \text{Mo} &= \frac{g\mu_1^4(\rho_1 - \rho_0)}{\rho_1^2 \sigma^3}. \end{aligned} \tag{76}$$

where Eö is the Eötvos number and Mo is the Morton number. The final rise speed w_c is then measured in the Reynolds number Re:

$$\mathrm{Re} = \frac{\rho_l w_c 2R}{\mu_1}. \tag{77}$$

In [30] the following parameters are chosen:

$$\begin{aligned} \text{Eö} &= 1.0 \\ \log(\mathrm{Mo}) &= -3.8. \end{aligned} \tag{78}$$

This corresponds to:

$$\begin{aligned} g &= 9.8\ \frac{m}{s^2}, & \sigma &= 0.0792\ \frac{kg}{s^2}, \\ \rho_0 &= 1.293\ \frac{kg}{m^3}, & \rho_1 &= 262\ \frac{kg}{m^3}, \\ \mu_0 &= 1.71 \times 10^{-6}\ \frac{kg}{ms}, & \mu_1 &= 5.50 \times 10^{-2}\ \frac{kg}{ms}. \end{aligned}$$

The bubble radius is $R = 0.0013\ m$. In this work, the dimensions of the domain are: $L_x = 0.01\ m$, $L_y = L_x$, $L_z = 2L_x$. The initial distance from the bubble to the floor of the domain is: $z_0 = 2R$. The grid size is $64 \times 64 \times 192$. The rise speed of the bubble w_c is presented in Fig. 25.

When the flow around the bubble is fully developed, the drag force F_d is balanced by the buoyancy force, i.e.:

$$F_d = \frac{4}{3}\pi R^3 g(\rho_1 - \rho_0). \tag{79}$$

Experimental data is usually fitted with the drag force drag coefficient c_d

$$c_d = \frac{F_d}{\frac{1}{2}\rho_1 w_c^2 \pi R^2}. \tag{80}$$

In [30] three different experimental closures are used to compare with the numerical results. These are:

$$(I)\ c_d = 14.9\mathrm{Re}^{-0.78}, \tag{81}$$

$$(II)\ c_d = \frac{3.05(783\gamma^2 + 2142\gamma + 1080)}{(60 + 29\gamma)(4 + 3\gamma)}\mathrm{Re}^{-0.74}, \tag{82}$$

$$(III)\ c_d = \max(\min(\frac{A}{\mathrm{Re}_b(1 + 0.15\mathrm{Re}_b^{0.6882})}, \frac{3A}{\mathrm{Re}_b}), \frac{8}{3}\frac{\text{Eö}}{\text{Eö} + 4}), \tag{83}$$

where $\gamma = \frac{\rho_0}{\rho_1}$. For more detail on these experimental closures see [30]. The resulting Reynolds numbers are shown in Table 1. There is agreement in the order of magnitude of the rise speed. When compared to the experimental data, the results of Gunsing shows a Reynolds number that is about 40% larger for the VOF method and 10% for the front tracking method. Our work under-predicts the rise speed with about 10%.

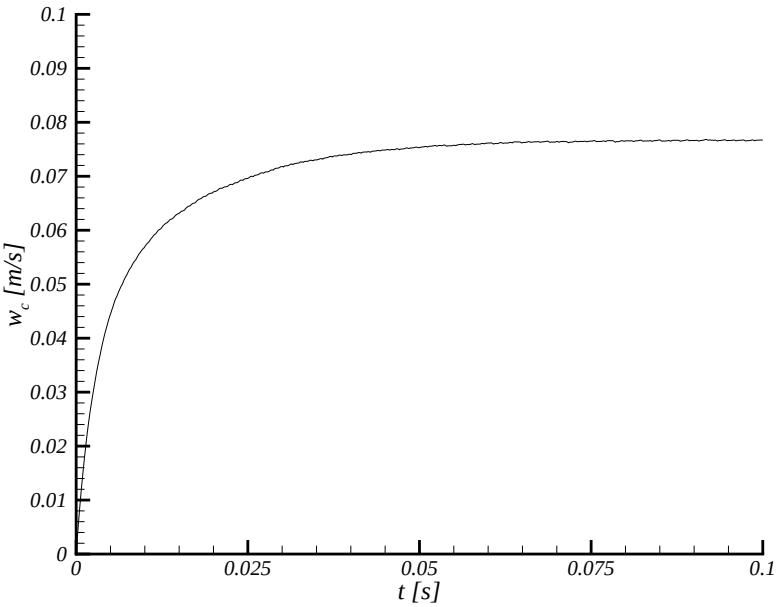
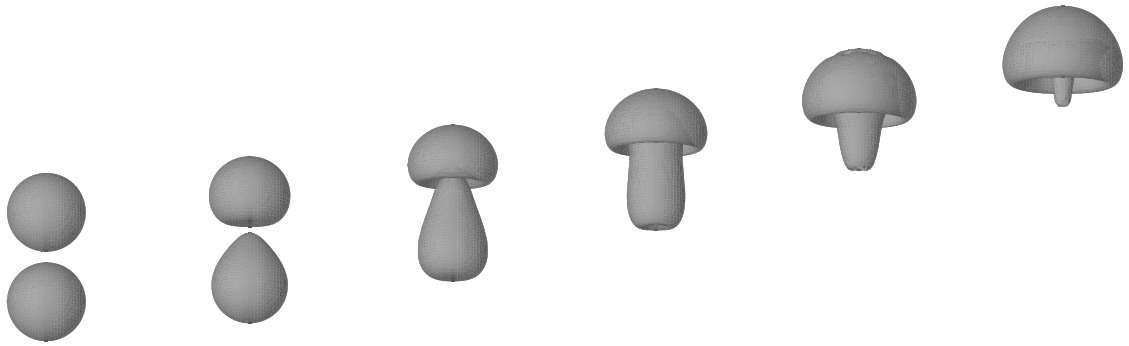


Figure 25: Rise speed of the bubble; Eö = 1.0; $\log(M) = -3.8$

			Re
Gunsing	[30]	VOF	8.2
Gunsing	[30]	front tracking	6.4
experimental	(I)	Eqn. (81)	5.2
experimental	(II)	Eqn. (82)	5.3
experimental	(III)	Eqn. (83)	5.8
this work			4.6

Table 1: Reynolds numbers of the rising bubble compared with the numerical results from Gunsing [30] and experimental data; Eö = 1.0; $\log(M) = -3.8$



$t = 0\ s$ $t = 0.02\ s$ $t = 0.04\ s$ $t = 0.06\ s$ $t = 0.08\ s$ $t = 0.1\ s$

Figure 26: Coalescence of two aligned rising bubbles; $96 \times 96 \times 192$ grid

7.7 Coalescence of two aligned rising bubbles

The coalescence of two rising bubbles is studied by De Sousa et al. in [28]. Two bubbles are initially at rest and aligned on the center line of the computational domain. The dimensions of the domain are: $L_x = 0.01\ m$, $L_y = 0.01\ m$, $L_z = 0.02\ m$. The radii of the bubbles are $R = 0.0013\ m$. The distance from the bottom bubble to the floor of the domain is: $z_0 = 0.0025\ m$. The distance between the bubbles is $\Delta z = 0.0055\ m$. Note that this makes the gap between the bubbles $\Delta z - 2R = 0.0004\ m$ (as prescribed by De Sousa et al. [28]). The gravity and material constants are:

$$\begin{aligned} g &= 9.8\ \frac{m}{s^2}, & \sigma &= 5.8 \times 10^{-4}\ \frac{kg}{s^2}, \\ \rho_0 &= 440\ \frac{kg}{m^3}, & \rho_1 &= 880\ \frac{kg}{m^3}, \\ \mu_0 &= 0.00625\ \frac{kg}{ms}, & \mu_1 &= 0.00125\ \frac{kg}{ms}, \end{aligned}$$

where subscripts $_0$ and $_1$ indicate the lighter and heavier fluid respectively. The coalescence of two rising bubbles is illustrated in Fig. 26. Snapshots are taken at equally spaced time intervals of $0.02\ s$, starting from the initial condition. Good qualitative agreement is found with the results presented by De Sousa et al. [28]. The snap-shots presented in [28] are taken at intervals of $0.03\ s$, which means that the bubbles rise speed in this research is approximately 35% larger than in [28]. Note that velocities predicted by our model have been validated in the previous sections.

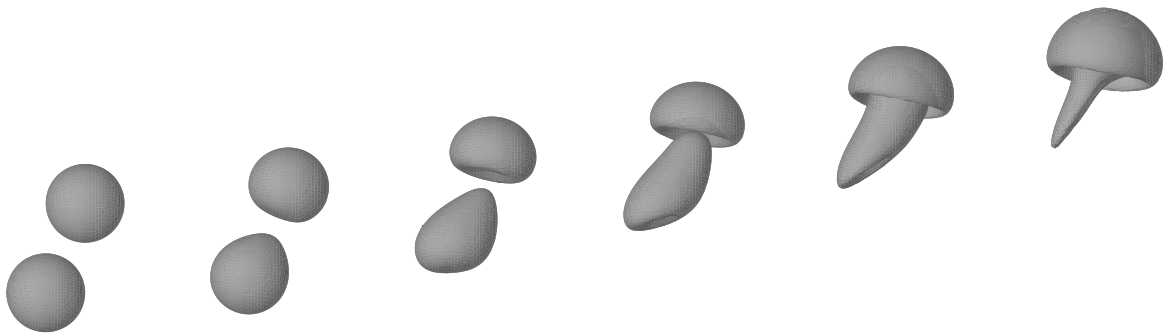


Figure 27: Coalescence of two misaligned rising bubbles; $96 \times 96 \times 192$ grid

7.8 Coalescence of two misaligned rising bubbles

The bubbles are initially misaligned to study asymmetric merging. The set-up is the same as for the aligned rising bubbles, except for a horizontal distance Δx between the bubbles. This distance is equal to the bubble radius, i.e. $\Delta x = R$. The results are plotted in Fig. 27. It can be seen that the bottom bubble is deformed considerable and sucked upward towards the top bubble. A thin tail is observed after merging. The tail becomes smaller, but never completely disappears. These simulations are also performed by De Sousa et al. [28]. The qualitative agreement is good, albeit that the rise speed is different. This was also the case for the aligned bubbles. Furthermore, the tail after merging of the two bubbles suddenly disappears in [28], whereas according to Fig. 27 the tail does not disappear. The tail remains but becomes thinner as time passes. The interface does also seems to be more irregular in [28], despite of the so-called 'undulation removal procedure' carried out by De Sousa et al. [28] to smooth the interface. Note that in this work the interface is not smoothed. Still a smooth interface is observed.

7.9 Computational Costs

The computational costs of the MCLS method in three dimensional space is presented in Table 2. The numbers correspond to CPU seconds spent per time-step for the misaligned rising bubble test case on a $96 \times 96 \times 192$ mesh. The interface advection takes approximately 13% of the total time-step. About 70% of the time-step is spent on solving the Poisson equation (Eqn. (6)).

	MCLS	
Level-Set advection	0.4	
re-initialization	2.3	
VOF advection		
flux x	0.7	
correct	3.9	
flux y	0.8	
correct	3.9	
flux z	0.8	
correct	3.9	
redist	6.1	
total	20.1	
Total advection		23
curv. smooth.		25
Poisson solver		126
Total time-step		180

Table 2: Computational costs measured in CPU seconds per time-step for the rising bubble test case on a $96 \times 96 \times 192$ mesh.

8 Conclusion

The extension of the mass conserving Level-Set (MCLS) method to three dimensions has been presented. The method is based on a Level-Set approach, with enhanced mass conservation by considering the fractional volume of fluid within a computational cell. Mass is conserved up to a specified tolerance. The MCLS method combines the appealing features of the Level-Set method with the mass-conservation properties of Volume-of-Fluid methods. Special attention is paid on the re-initialization procedure and the computation of surface tension. The applicability of the MCLS method was illustrated by the application to air-water flows.

References

[1] S.P. van der Pijl, A. Segal, C. Vuik, and P. Wesseling. A mass-conserving Level-Set method for modelling of multi-phase flows. *International Journal for Numerical Methods in Fluids*, 47:339–361, 2005.

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

[3] 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.

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

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

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

[7] 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.

[8] 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.

[9] M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187:110–136, 2003.

[10] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183:83–116, 2002.

[11] S.P. van der Pijl, A. Segal, and C. Vuik. Modelling of multi-phase flows with a Level-Set method. In M. Feistauer, V. Dolejší, P. Knobloch, and K. Najzar, editors, *Numerical Mathematics and Advanced Applications*, pages 698–707, Berlin, 2004. ENUMATH 2003, Springer.

[12] S.P. van der Pijl, A. Segal, and C. Vuik. Modelling of three-dimensional bubbly flows with a mass-conserving Level-Set method. In P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer, editors, *Proceedings European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004, Jyväskylä, Finland, July 24-28, 2004*, Jyväskylä, 2004. University of Jyväskylä, Department of Mathematical Information Technology.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] Scardovelli R and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31:567–603, 1999.

[25] 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.

[26] D. Enright, D. Nguyen, F. Gibou, and R. Fedkiw. Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In *Proceedings of the FEDSM03 3rd ASME/JSME Joint Fluids Engineering Conference*, 2003.

[27] E.R.A. Coyajee, M. Herrmann, and B.J. Boersma. Simulation of dispersed two-phase flow with a coupled Volume-of-Fluid/Level-Set method. In *Proceedings of the Summer Program 2004*. Center of Turblence Research, 2004.

[28] F.S. de Sousa, N. Mangiavacchi, L.G. Nonato, A. Castelo, M.F. Tomé, V.G. Ferreira, J.A. Cumanito, and S. McKee. A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces. *Journal of Computational Physics*, 198:469–499, 2004.

[29] G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, UK, 1967.

[30] M. Gunsing. *Modelling bubbly flows using volume of fluid, front tracking and discrete bubble models*. PhD thesis, University of Twenty, The Netherlands, 2004.