INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **47**:339–361
Published online 3 December 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.817

A mass-conserving Level-Set method for modelling of multi-phase flows

S. P. van der Pijl*,†, A. Segal, C. Vuik and P. Wesseling

Department of Applied Mathematical Analysis, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

SUMMARY

A mass-conserving Level-Set method to model bubbly flows is presented. The method can handle high density-ratio flows with complex interface topologies, such as flows with simultaneous occurrence of bubbles and droplets. Aspects taken into account are: a sharp front (density changes abruptly), arbitrarily shaped interfaces, surface tension, buoyancy and coalescence of droplets/bubbles. Attention is paid to mass-conservation and integrity of the interface.

The proposed computational method is a Level-Set method, where a Volume-of-Fluid function is used to conserve mass when the interface is advected. The aim of the method is to combine the advantages of the Level-Set and Volume-of-Fluid methods without the disadvantages. The flow is computed with a pressure correction method with the Marker-and-Cell scheme. Interface conditions are satisfied by means of the continuous surface force methodology and the jump in the density field is maintained similar to the ghost fluid method for incompressible flows. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: level-set; volume-of-fluid; incompressible; multi-phase; Navier–Stokes

1. INTRODUCTION

In this work incompressible two-phase flows are considered. The aim is to model high density-ratio flows with complex interface topologies, such as occur in air/water flows. Between the phases a sharp front exists, where density and viscosity change abruptly. This front is modelled as an interface. The interface is a moving (internal) boundary. Also, surface tension acts on the interface. A numerical method has to take this into account in order to handle arbitrarily shaped interfaces which may collide and break up.

*Correspondence to: S. P. van der Pijl, Department of Applied Mathematical Analysis, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands.
†E-mail: s.p.vanderpijl@math.tudelft.nl

Contract/grant sponsor: Netherlands Organization for Scientific Research (NWO)

Received 7 March 2004
Revised 27 August 2004
Accepted 14 October 2004

The proposed numerical method is based on the same approach as the Coupled Level-Set Volume-of-Fluid (CLSVOF) [1, 2] method in the sense that the Level-Set method is combined with a VOF method in order to conserve mass. However, instead of combining two existing methods, in this work the coupling between the Level-Set function and VOF functions is more straightforward and easily achieved. Furthermore, advection of the VOF function is carried out without interface reconstruction. Instead, additional information provided by the Level-Set formulation is used.

In this paper two-dimensional problems are considered. However, the goal is to simulate in three dimensions, therefore our numerical method allows easy extension to three dimensions. Furthermore, we aim to simulate large numbers of bubbles and droplets in future.

1.1. Survey of methods available

Various methods have been put forward to treat bubbly flows. In general, the interface representation can be explicit ('moving, boundary conforming mesh') or implicit ('fixed mesh') or a combination of both. Purely moving, boundary-conforming meshes are cumbersome for simulating large numbers of arbitrarily shaped interfaces. This technique is therefore not very suitable for the present work.

The *front-tracking* method [3, 4] and the closely related *immersed boundary* method [5] are combinations of fixed and moving mesh methods. Although the interface is tracked by an interface grid, the flow is solved on a fixed grid. The interface conditions are satisfied by regularizing (smoothing) the interface discontinuities and interpolating interface forces from the interface grid to the fixed grid. For this purpose, the interface forces are transformed into volume forces and distributed over a zone with non-zero width. This is sometimes referred to as the *continuous surface force* (CSF) approach [6]. In the *cut-cell* approach [7, 8] on the other hand, the interface conditions are satisfied without smoothing of the interface.

The interface grid will be difficult to evaluate when the interface has arbitrary shape and topology. Therefore an implicit interface definition by means of the *VOF* and *Level-Set* methodology is preferred for the present research. In the VOF method, a marker function, say Ψ, indicates the fractional volume of a certain fluid, say fluid '1', in a computational cell. In a grid cell Ω, Ψ is defined by

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

where χ is the characteristic function, which is 1 in fluid '1' and 0 elsewhere. The value of Ψ will be 0 or 1 in cells without the interface and $0 < \Psi < 1$ 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 behaviour of Ψ will be smoothed by straightforward application of a numerical scheme. Hence the location of the interface will become ill-defined. Furthermore, VOF methods can suffer from 'flotsam and jetsam', which are small remnants of mixed-fluid zones [9]. Furthermore, the step-like behaviour 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 straightforward. This can either be performed algebraically or geometrically. Algebraic methods try to discretize the advection equation for the marker function Ψ while incorporating the step-like behaviour of Ψ. Examples are: *Constrained Interpolation*

Profile [10] and *Flux-Corrected Transport* (FCT) [11, 12]. Geometric methods 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 co-ordinate direction [9]. In case of the *donor–acceptor* (SOLA–VOF) method the interface is stair-stepped [13, 14]. And with the Piecewise Linear Interface Calculation (PLIC) method the interface is piecewise linear [12, 15, 16]. When the discrete interface representation is more accurate, the interface reconstruction is more difficult. This is a drawback especially of the PLIC method, which on the other hand is the most accurate method.

An alternative for the VOF method is the Level-Set method [17, 18]. The interface is now defined by the zero level-set of a marker function, say Φ: $\Phi = 0$ at the interface, $\Phi > 0$ inside fluid '1' and $\Phi < 0$ elsewhere. 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 property:

$$\frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \nabla \Phi = 0 \tag{2}$$

Although this makes the Level-Set method elegant, its 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 Equation (2) more accurately by higher order schemes or by grid refinement. In Reference [19] higher order ENO discretization of Equation (2) is adopted and in Reference [20] the grid is refined adaptively near the interfaces.

The interface forces and discontinuities are often smeared out over a zone with non-zero width in a 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 Reference [21]. 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 Reference [19].

Mass-conservation is improved due to all the aforementioned measures, but is never exactly satisfied. Nice results are shown in Reference [22], where mass-conservation of the Level-Set method is improved by adding passively advected marker particles. These particles are used near the interface. It is required to keep track of the particles and to redistribute (or *reseed*) the particles. The velocity field has to be interpolated at the particle positions. Furthermore, the interface has to be reconstructed from the particles.

The CLSVOF method [1, 2] is a coupling of the Level-Set method with the VOF PLIC method. The flow-field is computed with a Level-Set method, which is advantageous for computing interface normals and curvature and regularization of discontinuities. In order to conserve mass, the interface is advected by a VOF (PLIC) method. After each interface update coupling of the Level-Set and VOF functions takes place. This coupling is not easily achieved. A drawback might be that the elaborateness of the VOF methods is imported. The advantage of the method is that the mass-conservation properties are shown to be comparable with VOF methods.

1.2. Mass-Conserving Level-Set (MCLS) method

The Level-Set method has some advantages over all other methods mentioned before. 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 and is considered the major drawback of the Level-Set method. Our work focuses on a mass-conserving way to advect the interface, resulting in what we will call the MCLS method.

This work has a shared foundation with the CLSVOF method [1, 2] and to a lesser extent with the combined Level-Set/particle method [22] in the sense that it is based on Level-Set and additional effort is made to conserve mass. The difference with CLSVOF is that here there is no combination of two existing methods. The method takes full advantage from all additional information provided by the Level-Set function Φ, rather than coupling Level-Set with VOF/PLIC. In fact we use the VOF function Ψ to conserve mass, without applying the difficult convection step (namely interface reconstruction) which makes VOF so elaborate.

We propose an explicit relation between the Level-Set function Φ and VOF function Ψ, which is the basis of the MCLS method. This relation is obtained by assuming piecewise linear interfaces within a computational cell, and can be written as:

$$\Psi = f(\Phi, \nabla\Phi) \tag{3}$$

This relationship makes advection of the VOF function Ψ easy (i.e. without interface reconstruction) and finding Φ from Ψ straightforward. The PLIC method is not adopted (unlike CLSVOF), yet mass is conserved in the same manner. Note that the PLIC method (and consequently the CLSVOF method) might not be easily extensible to 3D. Extension of MCLS to 3D is straightforward. Note also that with this approach, it is not necessary to smooth (or regularize) Ψ, which is usually necessary in other methods.

2. GOVERNING EQUATIONS

Consider two fluids labeled '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.:

$$\nabla \cdot \mathbf{u} = 0 \tag{4}$$

where $\mathbf{u} = (u, v)^t$ is the velocity vector. The flow is governed by the incompressible Navier–Stokes equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^t) + \mathbf{g} \tag{5}$$

where ρ, p, μ and $\mathbf{g}$ are the density, pressure, viscosity and gravity vector, respectively. The density and viscosity are constant within each fluid. We have

$$\rho = \rho_0 + (\rho_1 - \rho_0)H(\Phi) \tag{6}$$

and

$$\mu = \mu_0 + (\mu_1 - \mu_0)H(\Phi) \tag{7}$$

where the Φ is the Level-Set function describing the interface S, and H is the Heaviside step function.

2.1. Interface conditions

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

$$\begin{aligned} [\mathbf{u}] &= 0 \\ [p\mathbf{n} - \mathbf{n} \cdot \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{t}})] &= \sigma\kappa\mathbf{n} \end{aligned} \tag{8}$$

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. If $\mathbf{s}$ is parallel to the interface, $u_n = \mathbf{n} \cdot \mathbf{u}$ and $u_s = \mathbf{s} \cdot \mathbf{u}$, it can be shown [23, 24] that in general

$$\begin{aligned} \left[\frac{\partial u_n}{\partial n}\right] &= 0, & \left[\frac{\partial u_n}{\partial s}\right] &= 0 \\ \left[\frac{\partial u_s}{\partial n}\right] &= -[\mu]\frac{\partial u_n}{\partial s}, & \left[\frac{\partial u_s}{\partial s}\right] &= 0 \end{aligned} \tag{9}$$

Note that if the viscosity μ is continuous at the interface, Equation (9) shows that the derivatives of the velocity components are continuous too. In that case Equation (8) reduces to $[\mathbf{u}] = \mathbf{0}$ and $[p] = \sigma\kappa$. To achieve that, the viscosity is made continuous by smoothing expression (7):

$$\mu = \mu_0 + (\mu_1 - \mu_0)H_\alpha(\Phi) \tag{10}$$

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

$$H_\alpha(x) = \begin{cases} 0, & x < -\alpha \\ \frac{1}{2}\left(1 + \sin\left(\frac{x}{\alpha}\frac{1}{2}\pi\right)\right), & |x| \leqslant \alpha \\ 1, & x > \alpha \end{cases} \tag{11}$$

and α is a parameter proportional to the mesh width h. Here α is chosen as (following Reference [21]) $\alpha = \frac{3}{2}h$. According to Chang *et al.* [25], the viscosity is then smoothed over three mesh widths, provided $|\nabla\Phi| = 1$. Note that only the viscosity is smoothed, not the density ρ. Note also that when the density is not regularized, mass is conserved when the *volume* of a certain fluid or phase is conserved. In fact, the MCLS method conserves volumes by construction. Due to the non-regularized density-field, mass is conserved too. Instead of exactly taking into account the pressure-jump at the interface due to the surface tension forces, the CSF/stress [6] 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 [26]. The unknowns are stored in a Marker-and-Cell (staggered) layout [27]. For the interface representation the Level-Set methodology is adopted. The interface conditions are satisfied by means of the CSF methodology. The discontinuous density field is dealt with similarly to the ghost fluid method for incompressible flow [23].

3.1. Pressure correction

The Navier–Stokes equations (5) are discretized using finite differences. The unknown velocity components u and v are solved sequentially. Superscript n denotes time-level n. First a tentative velocity u^* is computed by

$$\frac{u^*-u^n}{\Delta t}=-u^n\left(\frac{\partial u}{\partial x}\right)^*-v^n\left(\frac{\partial u}{\partial y}\right)^* +\frac{1}{\rho}\left(\left(\frac{\partial \mu \partial u/\partial x}{\partial x}+\frac{\partial \mu \partial u/\partial y}{\partial y}\right)^*+\left(-\frac{\partial \mu \partial v/\partial y}{\partial x}+\frac{\partial \mu \partial v/\partial x}{\partial y}\right)^n\right) \tag{12}$$

The resulting system of equations is solved by a direct method. Note that there are no pressure gradient terms at the previous time level included in Equation (12). These are not incorporated in the predictor step 12, 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 $\nabla p^{n+1}/\rho-\nabla p^n/\rho=\nabla \delta p/\rho$, which is the usual formulation for single-phase flows. Furthermore, gravity terms are only included in the predictor if the pressure gradient is present. Since the pressure gradient is absent in Equation (12) no gravity is included. The equation for v is treated similarly. 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 stored and evaluated at different locations, are approximated by averaging [27]. Note that the stress tensor is split into a part on time level $*$ (implicit) and n (explicit), due to the fact that u and v are solved sequentially. Note that all terms containing u in the diffusion part of Equation (12) are implicit. In the stress tensor $u_x+v_y=0$ is used.

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

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^*}{\Delta t}=-\frac{1}{\rho}\nabla p+\mathbf{g} \tag{13}$$

under the constraint of Equation (4). This gives

$$\begin{aligned}\mathbf{u}^{n+1}&=\mathbf{u}^*+\Delta t\left(-\frac{1}{\rho}Gp+\mathbf{g}\right)\\ D\mathbf{u}^{n+1}&=0\end{aligned} \tag{14}$$

where D represents the discretization of the divergence and G is the discrete gradient operator, which remain to be specified. Discretization of $\frac{1}{\rho}Gp$ requires special care, because p and ρ are discontinuous at the interface. Computation of $\frac{1}{\rho}Gp$ is similar to that used in the ghost fluid

method for incompressible flows [23, 28] and will be discussed in the next section. Equation (14) gives:

$$D\frac{1}{\rho}Gp = D\left(\frac{1}{\Delta t}\mathbf{u}^* + \mathbf{g}\right) \tag{15}$$

Further details on this pressure-correction method can be found in Reference [26].

3.1.1. CSF. The surface tension force $\sigma\kappa$ in Equation (8) is approximated as follows. Adopting the CSF methodology, the interface force is transformed in a volume force by writing (see e.g. Reference [29])

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

where δ is the delta function. This results in an additional term in Equation (5) of the form

$$\frac{1}{\rho}\sigma\kappa\delta(\Phi)\nabla\Phi \tag{16}$$

which is discontinuous, due to the behaviour of ρ. The curvature of the interface is given by

$$\kappa = \nabla\cdot\frac{\nabla\Phi}{|\nabla\Phi|} \tag{17}$$

which is approximated by central differences. The average density $\bar{\rho} = \frac{1}{2}(\rho_0 + \rho_1)$ is taken in the force term:

$$\frac{1}{\bar{\rho}}\sigma\kappa\delta(\Phi)\nabla\Phi \tag{18}$$

Note that this is allowed, since integration of $\rho/\bar{\rho}\sigma\kappa\delta(\Phi)\nabla\Phi$ perpendicular to the interface still results in the correct interface force, which is the essence of the CSF approach. By regularizing the delta function in the same manner as the Heaviside step function (Equation (11))

$$\delta_\alpha(\Phi) = \begin{cases} \dfrac{1}{2\alpha}\left(1+\cos\left(\dfrac{\Phi}{\alpha}\pi\right)\right), & |\Phi| \leqslant \alpha \\ 0, & |\Phi| > \alpha \end{cases} \tag{19}$$

Equation (5) becomes

$$\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla\cdot\mu(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{t}}) + \mathbf{g} - \frac{1}{\bar{\rho}}\sigma\kappa\delta_\alpha(\Phi)\nabla\Phi \tag{20}$$

Here α has the same value as in Equation (11), i.e. $\alpha = \frac{3}{2}h$. The additional term $\sigma\kappa\delta_\alpha(\Phi)\nabla\Phi/\bar{\rho}$ is incorporated in Equations (13) and (14) in a straightforward way:

$$D\frac{1}{\rho}Gp = D\left(\frac{1}{\Delta t}\mathbf{u}^* + \mathbf{g} - \frac{1}{\bar{\rho}}\sigma\kappa\delta_\alpha(\Phi)\nabla\Phi\right) \tag{21}$$

and

$$\mathbf{u}^{n+1} = \mathbf{u}^* + \Delta t \left(-\frac{1}{\rho} Gp + \mathbf{g} - \frac{1}{\bar{\rho}} \sigma \kappa \delta_\alpha(\Phi) \nabla \Phi \right) \tag{22}$$

If the interface force was *not* regularized, but included as a pressure jump, the pressure-jump terms would enter the pressure-correction step in a similar manner. Note also that $\sigma\kappa/\rho$ is regularized and not $\sigma\kappa$. This keeps the density ρ (piecewise) constant, which guarantees a straightforward application of the pressure-correction methodology.

3.1.2. Discontinuous density field. In this paper the discontinuous density field is dealt with similarly to the ghost fluid method. In Reference [28], the pressure derivatives $1/\rho \partial p/\partial x$ at u locations $(i+1, j+\frac{1}{2})$ are approximated by

$$\left(\beta \frac{\partial p}{\partial x} \right)_{i+1, j+\frac{1}{2}} = \hat{\beta}_{i+1, j+\frac{1}{2}} \frac{p_{i+\frac{3}{2}, j+\frac{1}{2}} - p_{i+\frac{1}{2}, j+\frac{1}{2}}}{\Delta x} \tag{23}$$

and similarly for $\beta \partial p/\partial y$, where $\beta = 1/\rho$. Here we make use of $[1/\rho \nabla p] = \mathbf{0}$ [23]. Note that due to the CSF methodology no pressure jumps have to be taken into account at the interface. The quantity $\hat{\beta}$ is the harmonic average of β. Further details can be found in Reference [28].

3.2. Interface advection

The strategy of modelling bubbly flows is to compute the flow with a given interface position and to subsequently evolve the interface in the given flow field. In the previous sections the manner by which the flow is computed with a given interface position was described. Next we consider the evolution of the interface.

3.2.1. Level-set. The interface, say S, is the zero level-set of Φ:

$$S(t) = \{ \mathbf{x} \in \mathbb{R}^2 | \Phi(\mathbf{x}, t) = 0 \} \tag{24}$$

The interface is evolved by advecting the Level-Set function in the flow field as if it were a material property, according to Equation (2). A homogeneous Neumann boundary condition for Φ is imposed at the boundaries. The accuracy of the approximation of Equation (2) determines the accuracy of the interface representation. This accuracy will also determine the mass errors. The discretization of the gradient of Φ can be either first-order upwind, or second- or third-order ENO [21, 25, 29]. In case of first-order spatial discretization, a forward Euler temporal discretization is sufficient. In case of the higher order spatial discretization, a Runge–Kutta scheme is applied (see e.g. Reference [19]).

If the interface is smoothed over a zone with non-zero width, keeping Φ a distance function ensures that the front has finite thickness at all time [21, 25]. This is especially important when the surface tension force is distributed over a number of grid cells (CSF approach). However, if an initial signed distance function is advected through a non-uniform flow, it does not necessarily correspond to a distance function any longer. Therefore, we apply re-initialization as described in References [21, 25]. In References [19, 28, 30] re-initialization is improved to prevent additional mass-errors associated with re-initialization. Nevertheless, mass

errors still remain. Therefore a different approach is chosen, which is described in the next section.

3.2.2. MCLS. The difficulty with the Level-Set method is that conservation of Φ does not imply conservation of mass. On the other hand, with the VOF method, mass is conserved when Ψ is conserved. 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. Low-order advection and re-initialization ensure numerical smoothness of Φ. Furthermore, when the flow-field is computed, higher order accuracy might not be expected when the CSF method is applied and viscosity is regularized. In that respect, higher order discretization of Equation (2) will only lead to improved mass conservation for the pure Level-Set methods. 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 'VOF' function Ψ) is to be computed from the Level-Set function Φ^n: $\Psi = f(\Phi, \nabla\Phi)$;
2. the VOF function has to be advected conservatively during a time step towards Ψ^{n+1};
3. with this new VOF 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 will now be described in more detail.

Step 1 (*VOF function*): A relation between the Level-Set function Φ and the VOF function Ψ is found by considering the fractional volume of a certain fluid in a computational cell Ω_k. In this paper, a uniform Cartesian mesh is employed consisting of computational cells Ω_k, $k = 1, 2, \ldots$. The centre of Ω_k is denoted by $\mathbf{x}_k = (x_k, y_k)^t$ and Δx and Δy are the mesh sizes in x and y direction, respectively. In the computational cell Ω_k, the VOF function Ψ_k is defined by Equation (1). Employing the Level-Set function Φ, the characteristic function χ becomes $\chi = H(\Phi)$, where H is the Heaviside step function. The connection between Φ and Ψ is therefore:

$$\Psi_k = \frac{1}{\mathrm{vol}(\Omega_k)} \int_{\Omega_k} H(\Phi)\, \mathrm{d}\Omega \tag{25}$$

Φ is *linearized* around Φ_k, which is the value of Φ in $\mathbf{x}_k$: $\Phi = \Phi_k + \nabla\Phi_k \cdot (\mathbf{x} - \mathbf{x}_k)$. Substituting this into Equation (25) and taking $\xi = (x - x_k)/\Delta x$ and $\eta = (y - y_k)/\Delta y$ yields

$$\Psi_k = f(\Phi_k, \nabla\Phi_k) \tag{26}$$

where

$$f(\Phi_k, \nabla\Phi_k) = \int_{\xi=-\frac{1}{2}}^{\frac{1}{2}} \int_{\eta=-\frac{1}{2}}^{\frac{1}{2}} H\left(\Phi_k + \Delta x \left.\frac{\partial\Phi}{\partial x}\right|_k \xi + \Delta y \left.\frac{\partial\Phi}{\partial y}\right|_k \eta\right) \mathrm{d}\eta\, \mathrm{d}\xi \tag{27}$$

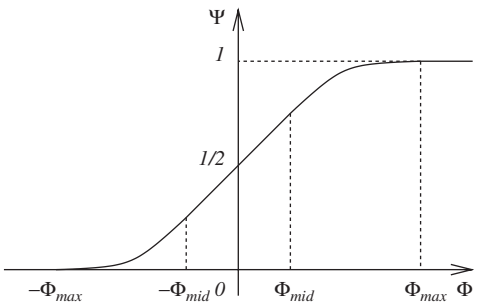

Figure 1. Fractional volume Ψ as function of Level-Set value Φ.
Note that Φ_{mid} and Φ_{max} are determined by $\nabla\Phi$.

Note that in contrast with other approaches, the Heaviside step function is not regularized. After some mathematical manipulations, the function f is obtained as

$$f(\Phi_k, \nabla\Phi_k) = \begin{cases} 0, & \Phi_k \leqslant -\Phi_{\max k} \\ \frac{1}{2}\dfrac{(\Phi_{\max k} + \Phi_k)^2}{\Phi_{\max k}^2 - \Phi_{\text{mid}k}^2}, & -\Phi_{\max k} < \Phi_k < -\Phi_{\text{mid}k} \\ \frac{1}{2} + \dfrac{\Phi_k}{\Phi_{\max k} + \Phi_{\text{mid}k}}, & -\Phi_{\text{mid}k} \leqslant \Phi_k \leqslant \Phi_{\text{mid}k} \\ 1 - \frac{1}{2}\dfrac{(\Phi_{\max k} - \Phi_k)^2}{\Phi_{\max k}^2 - \Phi_{\text{mid}k}^2}, & \Phi_{\text{mid}k} < \Phi_k < \Phi_{\max k} \\ 1, & \Phi_k \geqslant \Phi_{\max k} \end{cases} \tag{28}$$

where

$$\begin{aligned} &\Phi_{\max k} = \tfrac{1}{2}(|D_{xk}| + |D_{yk}|), \quad \Phi_{\text{mid}k} = \tfrac{1}{2}||D_{yk}| - |D_{xk}|| \\ &D_{xk} = \Delta x \left.\frac{\partial\Phi}{\partial x}\right|_k, \qquad D_{yk} = \Delta y \left.\frac{\partial\Phi}{\partial y}\right|_k \end{aligned} \tag{29}$$

which are approximated by central differencing. In Figure 1, f is plotted as a function of Φ. Note that in this figure, the values Φ_{mid} and Φ_{max} are determined by $\nabla\Phi$.

Step 2 (*VOF advection*): At a certain time instant the VOF function can be computed by means of Equation (27). The VOF function after a time step is found by considering the flux of fluid F that flows through a boundary Γ of a computational cell during time-step Δt:

$$F = \int_0^{\Delta t} \int_\Gamma H(\Phi(\mathbf{x}, t+\tau))\,\mathrm{d}S\,\mathrm{d}\tau \tag{30}$$

which is approximated by

$$F \approx \int_0^{\Delta t} \int_\Gamma H(\Phi(\mathbf{x} - \mathbf{u}\tau, t))\,\mathrm{d}S\,\mathrm{d}\tau = \int_{\Omega_D} H(\Phi(\mathbf{x}, t))\,\mathrm{d}\Omega \tag{31}$$

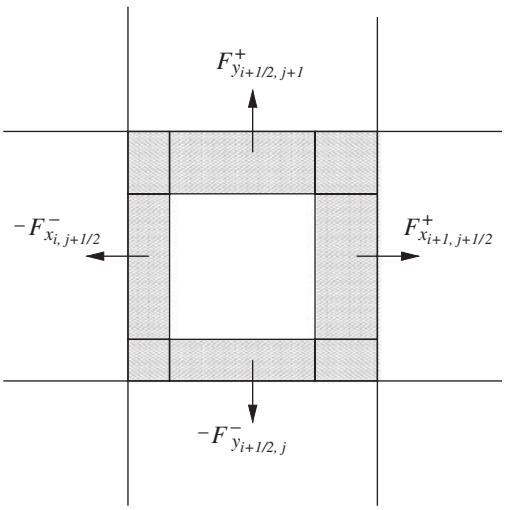

Figure 2. Donating regions for fluxes F_x and F_y. These are the shaded areas that will flow through the boundaries during a time step. Doubly fluxed areas exist near the corners of the cell.

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 Figure 2). Summation over all boundary faces leads to:

$$\Psi^{n+1}_{i+(1/2),j+(1/2)} = \Psi^{n}_{i+(1/2),j+(1/2)} - \frac{1}{\Delta x \Delta y}(F_{xi+1,j+(1/2)} - F_{xi,j+(1/2)} + F_{y_{i+(1/2),j+1}} - F_{y_{i+(1/2),j}}) \tag{32}$$

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 or at the right neighbouring cell. Formally, the flux can therefore be split into a contribution from both neighbours, called F^+ and F^-, respectively (see Figure 2). Of course if $F^+_{\ldots} \neq 0$ then $F^-_{\ldots} = 0$ and vice versa. In this way the fluxes can be written as

$$F_{xi+1,j+(1/2)} = F^+_{x\ i+1,j+(1/2)} + F^-_{x\ i+1,j+(1/2)}, \quad F_{y_{i+(1/2),j}} = F^+_{y\ i+(1/2),j} + F^-_{y\ i+(1/2),j} \tag{33}$$

and similarly for the other fluxes.

The fluxes are computed by linearizing Φ (just like Equation (27)):

$$F^+_{x\ i,j+(1/2)} = \Delta x \Delta y \int_{\xi=(1/2)-v^+}^{(1/2)} \int_{\eta=-(1/2)}^{(1/2)} H(\Phi_L + D_{x_L}\xi + D_{y_L}\eta)\,\mathrm{d}\eta\,\mathrm{d}\xi \tag{34}$$

and

$$F^-_{x\ i,j+(1/2)} = -\Delta x \Delta y \int_{\xi=-(1/2)}^{-(1/2)-v^-} \int_{\eta=-(1/2)}^{(1/2)} H(\Phi_R + D_{x_R}\xi + D_{y_R}\eta)\,\mathrm{d}\eta\,\mathrm{d}\xi \tag{35}$$

and similar for the other fluxes, where

$$v^+ = \frac{\max(u_{i,j+(1/2)}, 0)\Delta t}{\Delta x}, \quad v^- = \frac{\min(u_{i,j+(1/2)}, 0)\Delta t}{\Delta x} \tag{36}$$

and

$$\begin{aligned}
&\Phi_L=\Phi_{i-(1/2),j+(1/2)}, && \Phi_R=\Phi_{i+(1/2),j+(1/2)}\\
&D_{x_L}=\Delta x\left.\frac{\partial\Phi}{\partial x}\right|_{i-(1/2),j+(1/2)}, && D_{y_L}=\Delta y\left.\frac{\partial\Phi}{\partial y}\right|_{i-(1/2),j+(1/2)}\\
&D_{x_R}=\Delta x\left.\frac{\partial\Phi}{\partial x}\right|_{i+(1/2),j+(1/2)}, && D_{y_R}=\Delta y\left.\frac{\partial\Phi}{\partial y}\right|_{i+(1/2),j+(1/2)}
\end{aligned} \tag{37}$$

With some scaling, these fluxes become

$$\begin{aligned}
F^{+}_{x\ i,j+(1/2)}&=v^{+}\Delta x\Delta y f\left(\widehat{\Phi}_L,\left(\frac{1}{\Delta x}\widehat{D}_{x_L},\frac{1}{\Delta y}D_{y_L}\right)^{\mathrm{t}}\right)\\
F^{-}_{x\ i,j+(1/2)}&=v^{-}\Delta x\Delta y f\left(\widehat{\Phi}_R,\left(\frac{1}{\Delta x}\widehat{D}_{x_R},\frac{1}{\Delta y}D_{y_R}\right)^{\mathrm{t}}\right)
\end{aligned} \tag{38}$$

where

$$\begin{aligned}
&\widehat{\Phi}_L=\Phi_L+(1/2)(1-v^{+})D_{x_L}, && \widehat{\Phi}_R=\Phi_R-(1/2)(1+v^{-})D_{x_R}\\
&\widehat{D}_{x_L}=v^{+}D_{x_L}, && \widehat{D}_{x_R}=-v^{-}D_{x_R}
\end{aligned} \tag{39}$$

Time-step Δt and velocity u are included in v^{+} and v^{-}. This scaling of variables makes the advection of Ψ rather straightforward, since an analytic expression for the function f is given in Equations (28) and (29). The fluxes in the y direction are obtained in the same way.

Figure 2 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. The flux-splitting of Sussman and Puckett [1] is adopted:

$$\begin{aligned}
\Psi^{*}_{i+(1/2),j+(1/2)}&=\frac{\Psi^{n}_{i+(1/2),j+(1/2)}-\dfrac{1}{\Delta x\Delta y}(F^{n}_{x\,i+1,j+(1/2)}-F^{n}_{x\,i,j+(1/2)})}{1-\frac{\Delta t}{\Delta x}(u_{i+1,j+(1/2)}-u_{i,j+(1/2)})}\\
\Psi^{**}_{i+(1/2),j+(1/2)}&=\frac{\Psi^{*}_{i+(1/2),j+(1/2)}-\dfrac{1}{\Delta x\Delta y}(F^{*}_{y\,i+(1/2),j+1}-F^{*}_{y\,i+(1/2),j})}{1-\frac{\Delta t}{\Delta y}(v_{i+(1/2),j+1}-v_{i+(1/2),j})}\\
\Psi^{n+1}_{i+(1/2),j+(1/2)}&=\Psi^{**}-\Delta t\left(\Psi^{*}\frac{u_{i+1,j+(1/2)}-u_{i,j+(1/2)}}{\Delta x}+\Psi^{**}\frac{v_{i+(1/2),j+1}-v_{i+(1/2),j}}{\Delta y}\right)
\end{aligned} \tag{40}$$

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

Flux redistribution. As reported in Reference [1], undershoots and/or overshoots can still occur, which leads to unphysical values of Ψ, namely <0 and >1. If these values are replaced

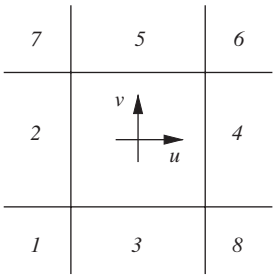

Figure 3. Order of flux redistributing for $u>0$ and $v>0$. The numbers in the graph indicate the order.

by 0 and 1, respectively, mass errors arise which are in general of the order 10^{-4}. This is also observed with the present method and applications. 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 $(i+(1/2)-\text{sign}(u), j+(1/2)-\text{sign}(v))$ neighbouring cells.

Assume that cell k has a value $\Psi_k^{n+1}<0$. Then define a flux F towards neighbouring cell l, so that $\Psi_k=\Psi_k^{n+1}-F=0$: $F=\Psi_k^{n+1}$. In order not to cause unphysical values of Ψ_l, limit F by

$$F=\min(\max(\Psi_k^{n+1},-\Psi_l^{n+1}),1-\Psi_l^{n+1}) \tag{41}$$

so that $0\leqslant\Psi_l=\Psi_l^{n+1}+F\leqslant 1$. Then make corrections to Φ_k and Φ_l by $\Psi_k=\Psi_k^{n+1}-F$ and $\Psi_l=\Psi_k^{n+1}+F$. In case of $\Psi_k^{n+1}>1$, the flux F is

$$F=\min(\max(\Psi_k^{n+1}-1,-\Psi_l^{n+1}),1-\Psi_l^{n+1}) \tag{42}$$

In two dimensions, there exist eight neighbouring cells l which can contribute to cell k. The order in which the neighbouring cells l are chosen is depicted in Figure 3. 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 neighbour first. The order of the other neighbours is arbitrary. The values of u and v at Ψ locations are interpolated by averaging.

Step 3 (*Inverse function*): Having found a new VOF function 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 $(\Phi_1,\Phi_2,\ldots)$, such that

$$|f(\Phi_k^{n+1},\nabla\Phi_k^{n+1})-\Psi_k^{n+1}|\leqslant\varepsilon \quad \forall k=1,2,\ldots \tag{43}$$

where ε is some tolerance. It will be clear that due to the behaviour 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 following iteration (until convergence): leave Φ unmodified in a grid point when the VOF constraint is satisfied and make corrections locally when this constraint is not satisfied. This is achieved by using the inverse function g of f as given in Equation (28) with respect to argument Φ_k:

$$f(g(\Psi,\nabla\Phi),\nabla\Phi)=\Psi \tag{44}$$

and employing Picard-iterations. Starting with $(\Phi_1^{n+1,*}, \Phi_2^{n+1,*}, \ldots)$, if at time step $n+1$ $(\Psi_1, \Psi_2, \ldots)$ has to be equal to $(\Psi_1^{n+1}, \Psi_2^{n+1}, \ldots)$, then the mth iteration is:

$$D_x = \Delta x \left.\frac{\partial \Phi}{\partial x}\right|_k^{n+1,m}, \qquad D_y = \Delta y \left.\frac{\partial \Phi}{\partial y}\right|_k^{n+1,m} \tag{45}$$

$$\Phi_{\max} = (1/2)(|D_x| + |D_y|), \quad \Phi_{\mathrm{mid}} = (1/2)\|D_y| - |D_x\|$$

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

$$\Phi_k^{n+1,m+1} = g(\Psi_k^{n+1}, \nabla \Phi_k^{n+1,m}) \tag{46}$$

where

$$g(\Psi, \nabla\Phi) = \begin{cases} -\Phi_{\max}, & \Psi_k^{n+1} \leqslant 0 \\ \mathscr{B}, & 0 < \Psi_k^{n+1} < 1 - \Psi_{\mathrm{mid}} \\ \mathscr{C}, & 1 - \Psi_{\mathrm{mid}} \leqslant \Psi_k^{n+1} \leqslant \Psi_{\mathrm{mid}} \\ \mathscr{D}, & \Psi_{\mathrm{mid}} < \Psi_k^{n+1} < 1 \\ \Phi_{\max}, & \Psi_k^{n+1} \geqslant 1 \end{cases} \tag{47}$$

and

$$\begin{aligned} \mathscr{B} &= \sqrt{2\Psi_k^{n+1}(\Phi_{\max}^2 - \Phi_{\mathrm{mid}}^2)} - \Phi_{\max} \\ \mathscr{C} &= (\Psi_k^{n+1} - (1/2))(\Phi_{\max} + \Phi_{\mathrm{mid}}) \\ \mathscr{D} &= -\sqrt{2(1 - \Psi_k^{n+1})(\Phi_{\max}^2 - \Phi_{\mathrm{mid}}^2)} + \Phi_{\max} \end{aligned} \tag{48}$$

and

$$\Psi_{\mathrm{mid}} = \frac{1}{2}\frac{\Phi_{\max} + 3\Phi_{\mathrm{mid}}}{\Phi_{\max} + \Phi_{\mathrm{mid}}} \tag{49}$$

These iterations are repeated until $\max_k |\Psi_k^{n+1,m+1} - \Psi_k^{n+1}| \leqslant \varepsilon$. A graphical overview of the method is given in Figure 4.

3.3. Time-step restrictions

Following References [23, 29], 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 VOF function Ψ are advected explicitly, the restriction due to advection is:

$$\Delta t_{\mathrm{c}} = \frac{1}{|u|_{\max}/\Delta x + |v|_{\max}/\Delta y} \tag{50}$$

The restriction due to surface tension given in Reference [23] is

$$\Delta t_{\mathrm{s}} = \frac{1}{\sqrt{\sigma |\kappa|_{\max}/\min(\rho_0, \rho_1)\min(\Delta x, \Delta y)^2}} \tag{51}$$

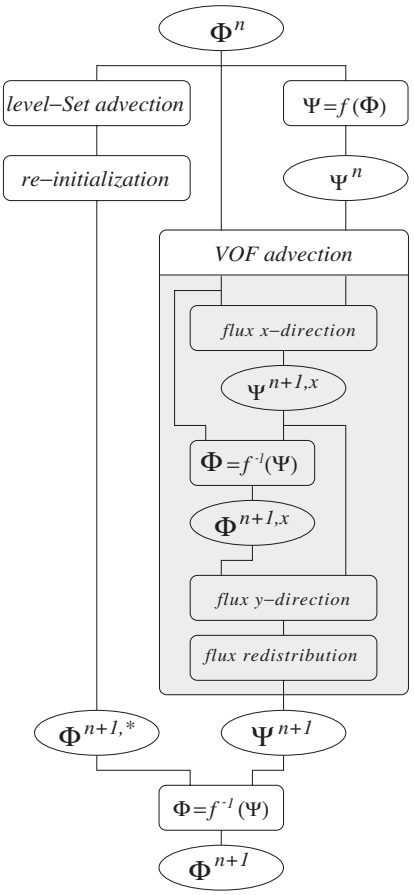

Figure 4. MCLS method: interface advection; Φ: Level-Set function; Ψ: VOF function. The left-hand side branch corresponds to pure Level-Set advection. The right-hand side branch represents the VOF advection.

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

$$\Delta t_s = \frac{1}{\sqrt{|\sigma\kappa\delta(\Phi)|_{\max}/\min(\rho_0, \rho_1)\min(\Delta x, \Delta y)}} \tag{52}$$

Diffusion is accounted for implicitly, hence no time-step restriction is encountered. For the time-step Δt finally holds (see e.g. Reference [29])

$$\Delta t \leqslant \mathrm{CFL}\ \min(\Delta t_c, \Delta t_s) \tag{53}$$

where, again following References [23, 29], $\mathrm{CFL} = (1/2)$ is used.

4. APPLICATIONS

The behaviour of the MCLS approach is shown by a couple of standard advection tests, 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, respectively.

4.1. Advection tests

4.1.1. Linear advection. The first advection test is presented in Figure 5. The velocity field is prescribed by $(u, v) = (0, -1)$. The dimensions of the computational domain are: $L_x = 10$ and $L_y = 100$. We use a 10×100-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 Figure 6 as function of the traversed distance of the circle. First-, second- and third-order pure Level-Set simulations (with and without re-initialization) are compared with the MCLS method. ENO discretization is adopted for the pure Level-Set method (see aforementioned references). The order of re-initialization is in agreement with the order of advection. The tolerance in the VOF advection is taken to be: $\varepsilon = 10^{-8}$. Globally speaking it can be said that mass is always lost for the pure Level-Set advection. Mass losses are smaller for higher accuracy and re-initialization causes much higher mass losses. The MCLS method conserves mass up to the specified tolerance.

4.1.2. Zalesak's rotating disc. The advection test of Zalesak [11] is often used to demonstrate the interface-advection algorithm (see e.g. References [12, 15, 16] for VOF methods and References [1, 19, 22, 29] for Level-Set methods). A slotted disc (Figure 7) is rotated through one revolution around the centre of the computational domain. The velocity-field is prescribed by: $(u, v) = (-(y - (1/2)L_y), x - (1/2)L_x)$. The centre of the slotted disc $(x_0, y_0)^t$ is located at $(x_0, y_0) = ((1/2)L_x, \frac{3}{4}L_y)$. The sizes are: $L_x = L_y$, $R_0 = \frac{3}{20}L_x$ and $W = \frac{1}{3}R_0$.

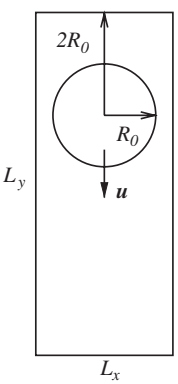

Figure 5. Linear advection test. A disc of radius R_0 is advected in a rectangular domain of L_x by L_y. The centre of the disc is initially placed at $2R_0$ from the top of the domain. The advection velocity is u.

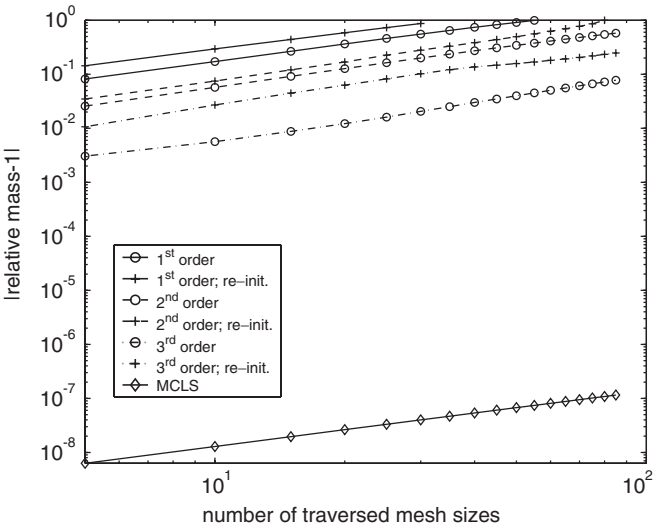

Figure 6. Relative mass errors for the linear advection test; $\varepsilon = 10^{-8}$ (every 10th iteration marked). Pure Level-Set advection with different discretization orders are compared with MCLS.

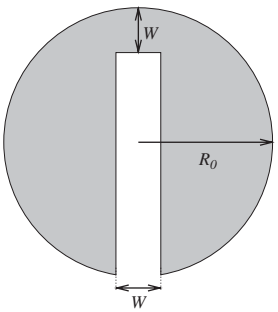

Figure 7. Zalesak's slotted disc advection test (to scale). The dimensions of the slot are depicted in the graph.

In Figures 8 and 9 results are shown for various mesh sizes. As might be expected, given the foregoing linear advection results, mass is still lost with 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 the case of the MCLS method.

The length $l(S)$ of the interface S can be expressed as (see e.g. Reference [25])

$$l(S) = \int_S \mathrm{d}S = \int_\Omega \delta(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega \tag{54}$$

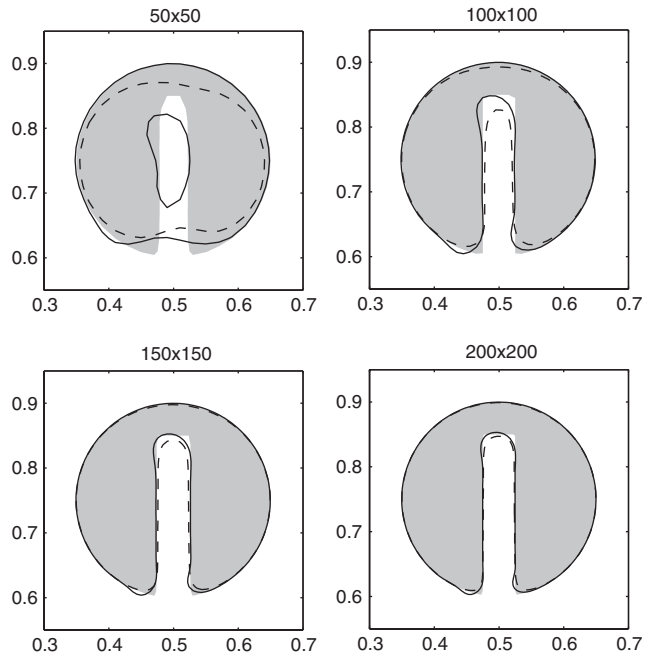

Figure 8. Results for Zalesak's advection test; the shaded area indicates the initial contour. The dashed lines indicate the interface after one revolution with third-order pure Level-Set advection. The solid lines correspond to MCLS advection. Four different mesh sizes have been employed.

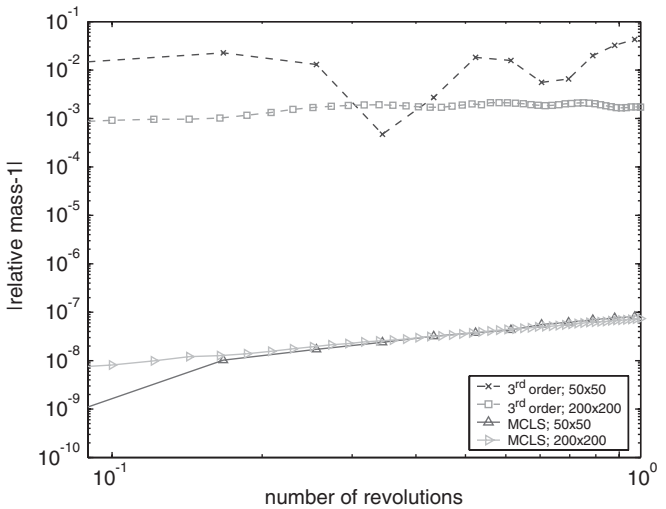

Figure 9. Relative masses for Zalesak's advection test; $\varepsilon = 10^{-8}$ (every 50th iteration marked). Pure Level-Set advection with third-order discretization is compared with MCLS for different mesh sizes.

Table I. Computed interface lengths after one revolution.

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

This is approximated by using central differences and regularization of the Dirac delta function (see Equation (10)):

$$l(S) = \int_\Omega \delta_\alpha(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega \tag{55}$$

where

$$\delta_\alpha(x) = \begin{cases} 0, & |x| > \alpha \\ \dfrac{1 + \cos(\frac{\pi x}{\alpha})}{2\alpha}, & |x| \leqslant \alpha \end{cases} \tag{56}$$

Note that due to Equation (10), the exact value of Φ has no meaning in the Level-Set formulation; only its sign is relevant. The α in Equation (55) therefore equals the α of Equation (10). The exact length of the interface is

$$l(S^0) = \left(4 + 2\pi - 2\arctan\left(\frac{1}{2}\frac{W}{R_0}\right) - \frac{W}{R_0}\right) R_0 \tag{57}$$

In Table I, the computed interface lengths after one revolution are compared with the exact length. 'Initial' means at $t = 0$, when errors are made due to the regularization of the delta function. Furthermore, '3rd 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 be defined as

$$||e||_p = \left(\frac{\int_S |(\Phi - \Phi^0)/L_x|^p\,\mathrm{d}S}{\int_S \mathrm{d}S}\right)^{1/p} = \left(\frac{\int_\Omega |(\Phi - \Phi^0)/L_x|^p \delta_\alpha(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega}{\int_\Omega \delta_\alpha(\Phi)|\nabla\Phi|\,\mathrm{d}\Omega}\right)^{1/p} \tag{58}$$

where L_x is used to non-dimensionalize Φ and Φ^0 is the initial Level-Set function. Results are presented in Figure 10. In all cases the MCLS approach is superior to the third-order ENO, pure Level-Set method.

4.2. *Air/water flow*

In Reference [23], a two-dimensional rising air bubble in water is considered. The dimensions and sizes are: $L_x = 0.02$ m, $L_y = \frac{3}{2}L_x$, $R_0 = \frac{1}{6}L_x$, $x_0 = y_0 = (1/2)L_x$. The gravity and material constants are: $g = 9.8\,\mathrm{m/s^2}$, $\sigma = 0.0728\,\mathrm{kg/s^2}$, $\rho_\mathrm{w} = 10^3\,\mathrm{kg/m^3}$, $\rho_\mathrm{a} = 1.226\,\mathrm{kg/m^3}$, $\mu_\mathrm{w} = 1.137 \times 10^{-3}$ kg/m s and $\mu_\mathrm{a} = 1.78 \times 10^{-5}$ kg/m s, where subscripts w and a indicate water and air, respectively.

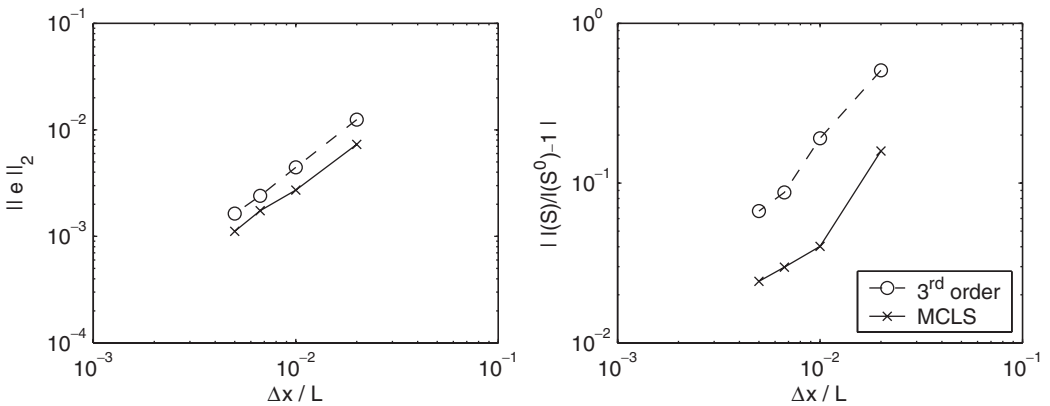

Figure 10. Errors for Zalesak's advection test. The left graph show the errors of the Level-Set function Φ near the interface after one revolution. The right graph shows the errors in the computed interface length.

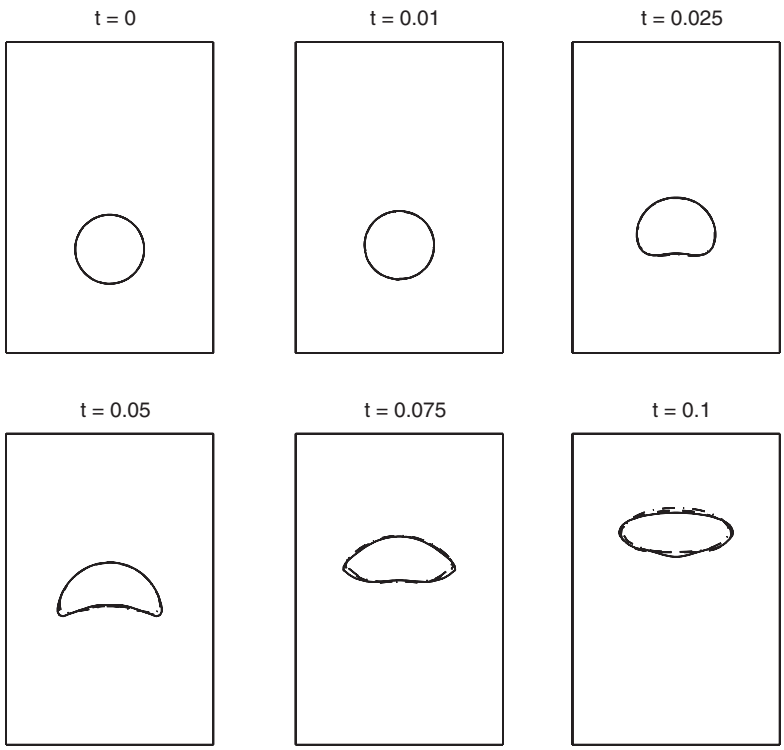

Figure 11. Interface positions for the rising bubble. Three different mesh sizes have been employed: $- \cdot -$: 30×45; $- -$: 40×60; —: 60×90 mesh. Snapshots are presented with equidistant time-steps.

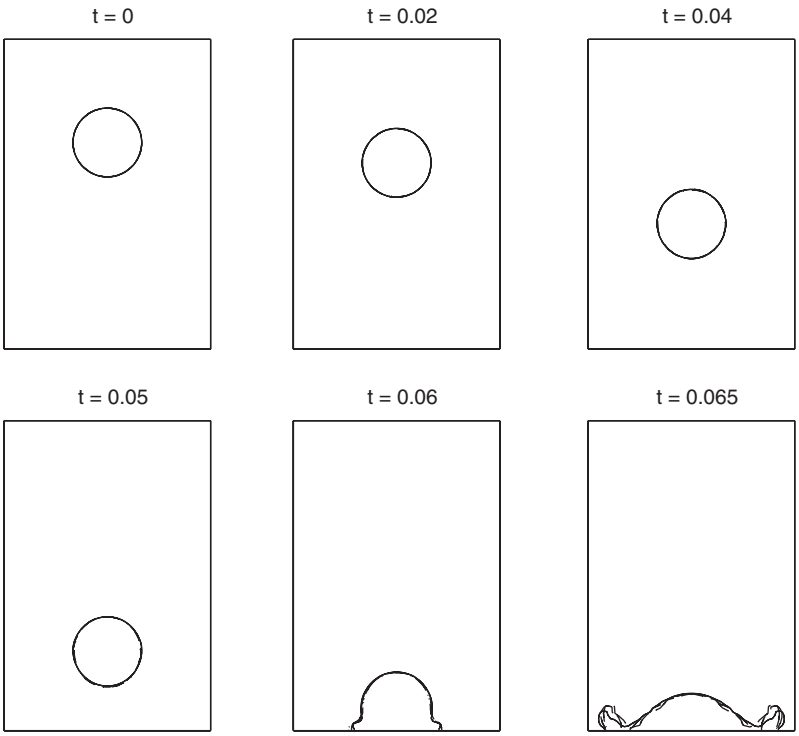

Figure 12. Interface positions for the falling droplet; three different mesh sizes have been employed: $-\cdot-$: 30×45; $--$: 40×60; —: 60×90 mesh. Snapshots are presented with equidistant time-steps.

Results are shown in Figure 11 for three different mesh sizes. We take $\varepsilon=10^{-8}$. Relative mass losses are of the same order and in agreement with the advection tests. Note that the number of grid cells is much smaller than in Reference [23]. The results are the same for $t\leqslant0.025$ for all mesh sizes. Thereafter small differences occur. The results compare well with Reference [23]. The MCLS method seems to result in a more compact 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.

In Figure 12, 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 $y_0=L_x$. Note that we use homogeneous Neumann boundary conditions for the Level-Set function Φ and VOF function Ψ. Mass conservation properties are the same as before. The results 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 Reference [23]. Note that the results in Reference [23] span $t\leqslant0.05$; no results after collision are presented.

4.3. Computational costs

The computational costs of the MCLS method are compared with purely third-order Level-Set advection in Table II. The numbers correspond to CPU seconds spent per time-step for

Table II. Computational costs measured in CPU seconds per time-step.

	MCLS		3rd order
Level-Set advection		0.03	0.07
Re-initialization		0.02	0.07
VOF advection			
Flux x	0.04		
Correct	0.04		
Flux y	0.04		
Correct	0.05		
Redist	0.06		
Total		0.23	
Total advection		0.28	0.14
Total time-step		1.50	1.35

MCLS is compared with third-order Level-Set advection for the rising bubble test case on a 60×90 mesh.

the rising bubble test case on a 60×90 mesh. Note that the Level-Set advection is first-order accurate in case of MCLS. The interface advection for MCLS becomes approximately twice as expensive. This makes the total time-step approximately 10% more expensive. On the other hand, mass is conserved up to the specified tolerance in case of MCLS, whereas mass is lost/gained for the third-order pure Level-Set advection (see Figures 6 and 9 for the advection tests).

5. CONCLUSION

A novel mass conserving Level-Set (MCLS) method 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. Advection tests were used to compare the method with the Level-Set method. Mass is conserved up to a specified (vanishing) tolerance. The MCLS method combines the apealing features of the Level-Set method with the mass-conservation properties of Volume-of-Fluid methods. The implementation is much easier than for a Volume-of-Fluid (based) method, especially in three-dimensional space. The applicability of the MCLS method was illustrated by the application to air–water flows. It is possible to capture bubbles or droplets with a moderate number of grid cells without mass loss up to the prescribed tolerance. This is an important feature for three-dimensional applications, where the number of grid cells available to an individual bubble will be severely limited.

REFERENCES

1. Sussman M, Puckett EG. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics* 2000; **162**:301–337.
2. Sussman M. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics* 2003; **187**:110–136.
3. Unverdi SO, Tryggvason G. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics* 1992; **100**:25–37.

4. Tryggvason G, Bunner B, Esmaeeli A, Juric D, Al-Rawahi N, Tauber W, Han J, Nas S, Jan Y-J. A front-tracking method for the computation of multiphase flow. *Journal of Computational Physics* 2001; **169**:708–759.
5. Lai M-C, Peskin CS. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics* 2000; **160**:705–719.
6. Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. *Journal of Computational Physics* 1992; **100**:335–354.
7. Udaykumar HS, Mittal R, Rampunggoon P, Khanna A. A sharp interface cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics* 2001; **174**:345–380.
8. Ye T, Shyy W, Chung JN. A fixed-grid, sharp-interface method for bubble dynamics and phase change. *Journal of Computational Physics* 2001; **174**:781–815.
9. Noh WF, Woodward P. SLIC (simple line interface calculations). In *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics*, van de Vooren AI, Zandbergen PJ (eds), Lecture Notes in Physics, vol. 59. Springer: New York, 1976; 330–340.
10. Yabe T, Xiao F, Utsumi T. The constrained interpolation profile method for multiphase analysis. *Journal of Computational Physics* 2001; **169**:556–593.
11. Zalesak ST. Fully multidimensional flux-corrected transport algorithm for fluids. *Journal of Computational Physics* 1979; **31**:335–362.
12. Rudman M. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**:671–691.
13. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
14. Lafaurie B, Nardone C, Scardovelli R, Zaleski S, Zanetti G. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics* 1994; **113**:134–147.
15. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
16. Harvie DJE, Fletcher DF. A new volume of fluid advection algorithm: the defined donating region scheme. *International Journal for Numerical Methods in Fluids* 2001; **35**:151–172.
17. Mulder W, Osher S, Sethian JA. Computing interface motion in compressible gas dynamics. *Journal of Computational Physics* 1992; **100**:209–228.
18. Osher S, Fedkiw RP. Level set methods: an overview and some recent results. *Journal of Computational Physics* 2001; **169**:463–502.
19. Sussman M, Fatemi E. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on Scientific Computing* 1999; **20**(4):1165–1191.
20. Sussman M, Almgren AS, Bell JB, Colella P, Howell LH, Welcome ML. An adaptive level set approach for incompressible two-phase flows. *Journal of Computational Physics* 1999; **148**:81–124.
21. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**:146–159.
22. Enright D, Fedkiw R, Ferziger J, Mitchell I. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics* 2002; **183**:83–116.
23. Kang M, Fedkiw RP, Liu X-D. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing* 2000; 323–360.
24. Li Z, Lai M-C. The immersed interface method for the Navier–Stokes equations with singular forces. *Journal of Computational Physics* 2001; **171**:822–842.
25. Chang YC, Hou TY, Merriman B, Osher S. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics* 1996; **124**:449–464.
26. van Kan JJIM. A second-order accurate pressure correction method for viscous incompressible flow. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**:870–891.
27. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surfaces. *Physics of Fluids* 1965; **8**:2182–2189.
28. Liu X-D, Fedkiw RP, Kang M. A boundary condition capturing method for Poisson's equation on irregular domains. *Journal of Computational Physics* 2000; **160**:151–178.
29. Sussman M, Fatemi E, Smereka P, Osher S. An improved level set method for incompressible two-phase flows. *Computers and Fluids* 1998; **27**:663–680.
30. Zheng LL, Zhang H. An adaptive level set method for moving-boundary problems: application to droplet spreading and solidification. *Numerical Heat Transfer*, *Part B* 2000; **37**:437–454.

