European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS 2004
P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer (eds.)
Jyväskylä, 24–28 July 2004

MODELING OF THREE-DIMENSIONAL BUBBLY FLOWS WITH A MASS-CONSERVING LEVEL-SET METHOD

S.P. van der Pijl[⋆1], **A. Segal and C. Vuik**

[⋆]Delft University of Technology, Delft, The Netherlands
e-mail: S.P.vanderPijl@EWI.TUDelft.nl, web page: http://ta.twi.tudelft.nl/users/vdpijl

Key words: Multi-Phase, Incompressible, Level-Set, Volume-of-Fluid

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.

[1]Supported by the Netherlands Organization for Scientific Research (NWO)

1 Introduction

Various methods have been put forward to treat bubbly flows. The two methods that are most suitable for the current research are the *Volume-of-Fluid* (VOF) method and the *Level-Set* method. For both methods a marker function is used to define the interface. In the Volume-of-Fluid method, a marker function, say Ψ, indicates the fractional volume of a certain fluid, say fluid '1', in a computational cell.

An alternative for the Volume-of-Fluid method is the Level-Set method ([1, 2]). 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 was a material constant:

$$\frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \nabla \Phi = 0. \tag{1}$$

The Level-Set method has some advantages over the Volume-of-Fluid method. Especially when solving the flow-field is concerned, since interface normals, curvature and distance towards the interface can be expressed easily in terms of Φ and its derivatives. Also, advecting the interface is possible by the application of 'of-the-shelf' techniques available from 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 Mass-Conserving Level-Set method (MCLS, [3]).

This work has a shared foundation with the CLSVOF method ([4, 5]) and to a lesser extend with the combined Level-Set/particle method ([6]) 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 Volume-of-Fluid/PLIC. In fact we use the Volume-of-Fluid function Ψ as a help variable to conserve mass, without applying the difficult convection (namely interface reconstruction) which makes the VOF so elaborate. We propose a simple relationship between the Level-Set function Φ and Volume-of-Fluid function Ψ. This relation is obtained by assuming piecewise linear interfaces within a computational cell:

$$\Psi = f(\Phi, \nabla \Phi). \tag{2}$$

It makes the advection of the Volume-of-Fluid function Ψ easy (i.e. without interface reconstruction) and finding Φ from Ψ a straightforward task. The PLIC method is not adopted (unlike CLSVOF), yet mass is conserved in the same manner. Note that the CLSVOF method might not be easily extendible to 3D space. Yet the extension of MCLS to three-dimensional space can be done in a straightforward way. 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 '0', and '1' in domain $\Omega \in \mathbb{R}^3$ which are separated by an interface S. Both fluids are assumed to be incompressible, i.e.:

$$\nabla \cdot \mathbf{u} = 0, \tag{3}$$

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

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

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

$$\mu = \mu_0 + (\mu_1 - \mu_0)H(\Phi) \tag{5}$$

and similar for ρ, where 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:

$$\begin{aligned} [\mathbf{u}] &= 0 \\ [p\mathbf{n} - \mathbf{n} \cdot \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^t\right)] &= \sigma \kappa \mathbf{n}, \end{aligned} \tag{6}$$

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 the viscosity μ is continuous at the interface, it can be shown that the derivatives of the velocity components are continuous too ([7, 8]). In that case Eqn. (6) reduces to $[\mathbf{u}] = \mathbf{0}$ and $[p] = \sigma\kappa$. To achieve that, the viscosity is made continuous by smoothing Expression (5):

$$\mu = \mu_0 + (\mu_1 - \mu_0)H_\alpha(\Phi), \tag{7}$$

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

$$H_\alpha(x) = \begin{cases} 0 & x < -\alpha \\ \frac{1}{2}\left(1 + \sin(\frac{x}{\alpha}\frac{1}{2}\pi)\right) & |x| \leq \alpha \\ 1 & x > \alpha \end{cases} \tag{8}$$

and α is a parameter proportional to the mesh width. Here α is chosen as (following [9]) $\alpha = \frac{3}{2}h$, where h is the mesh width. According to [10], 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 taking into account the pressure-jump at the interface due to the surface tension forces, the continuous surface force/stress (CSF, [11]) 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 ([12]). The unknowns are stored in a Marker-and-Cell (staggered) layout ([13]). 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 GhostFluid method for incompressible flow ([7]). Further information about the flow-field computations can be found in [3].

3.1 Interface advection

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

3.1.1 Level-Set

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

$$S(t) = \left\{ \mathbf{x} \in \mathrm{I\!R}^2 | \Phi(\mathbf{x}, t) = 0 \right\}. \tag{9}$$

The interface is evolved by advecting the Level-Set function in the flow field as if it were a material constant (Eqn. (1)):

$$\frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \nabla \Phi = 0. \tag{10}$$

A homogeneous Neumann boundary condition for Φ is imposed at the boundaries. It will be clear that accuracy of the approximation of Eqn. (10) determines the accuracy of the interface representation. The accuracy will also determine the mass errors. We use a first-order spatial and a forward Euler temporal discretization.

3.1.2 MCLS

The difficulty with the Level-Set method is conservation of Φ does not imply conservation of mass. On the other hand, with the Volume-of-Fluid 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 unmodified re-initialization. Low order advection and re-initialization will 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 Eqn. (10) 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 'volume-of-fluid' function Ψ) is to be computed from the Level-Set function Φ^n: $\Psi = f(\Phi, \nabla\Phi)$;
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 will be explained subsequently.

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

$$\Psi_k = \frac{1}{\mathrm{vol}(\Omega_k)} \int\limits_{\Omega_k} H(\Phi)\,\mathrm{d}\Omega. \tag{11}$$

where H is the Heaviside step function. The Level-Set function Φ is *linearized* around Φ_k, which leads to

$$\Psi_k = f(\Phi_k, \nabla\Phi_k). \tag{12}$$

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

$$f = \frac{\mathcal{A}}{6D_\xi D_\eta D_\zeta} \qquad \Phi_k \leq 0 \tag{13}$$

and

$$f = 1 - f(-\Phi_k, \nabla\Phi_k) \qquad \Phi_k > 0, \tag{14}$$

where

$$\begin{array}{rcl} \mathcal{A} & = & \max(\Phi_A,0)^3 - \max(\Phi_B,0)^3 - \\ & & \max(\Phi_C,0)^3 - \max(\Phi_D,0)^3 + \\ & & \max(\Phi_E,0)^3 \end{array} \tag{15}$$

and

$$\begin{array}{rcl} \Phi_A & = & \Phi_k + \frac{1}{2}D_\xi + \frac{1}{2}D_\eta + \frac{1}{2}D_\zeta \\ \Phi_B & = & \Phi_k + \frac{1}{2}D_\xi + \frac{1}{2}D_\eta - \frac{1}{2}D_\zeta \\ \Phi_C & = & \Phi_k + \frac{1}{2}D_\xi - \frac{1}{2}D_\eta + \frac{1}{2}D_\zeta \\ \Phi_D & = & \Phi_k - \frac{1}{2}D_\xi + \frac{1}{2}D_\eta + \frac{1}{2}D_\zeta \\ \Phi_E & = & \Phi_k + \frac{1}{2}D_\xi - \frac{1}{2}D_\eta - \frac{1}{2}D_\zeta \end{array} \tag{16}$$

and

$$\begin{array}{rcl} D_\xi & = & \max(|D_x|,|D_y|,|D_z|) \\ D_\zeta & = & \min(|D_x|,|D_y|,|D_z|) \\ D_\eta & = & |D_x| + |D_y| + |D_z| - D_\xi - D_\zeta \end{array} \tag{17}$$

and

$$\begin{array}{rcl} \Phi_k & = & \Phi(\mathbf{x}_k) \\ D_x & = & \Delta x \left.\frac{\partial\Phi}{\partial x}\right|_k \\ D_y & = & \Delta y \left.\frac{\partial\Phi}{\partial y}\right|_k \\ D_z & = & \Delta z \left.\frac{\partial\Phi}{\partial z}\right|_k, \end{array} \tag{18}$$

which are approximated by central differences.

Step 2: Volume-of-Fluid advection At a certain time instant the volume-of-fluid function can be computed from Φ by means of Eqn. (12). The volume-of-fluid 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:

$$\Psi^{n+1}_{i,j,k} = \Psi^n_{i,j,k} - \frac{1}{\Delta x \Delta y \Delta z}\Big(\begin{array}{lcl} F_{x\,i+\frac{1}{2},j,k} & - & F_{x\,i-\frac{1}{2},j,k} + \\ F_{y\,i,j+\frac{1}{2},k} & - & F_{y\,i,j-\frac{1}{2},k} + \\ F_{z\,i,j,k+\frac{1}{2}} & - & F_{z\,i,j,k-\frac{1}{2}}\Big). \end{array} \tag{19}$$

The fluxes are again computed by linearizing Φ. In fact, the fluxes are computed by the straightforward application of f (Eqn. (13)).

It is possible that fluid is fluxed more than once through different faces, which would cause unphysical values of Ψ. As reported in e.g. [14], this can be solved by employing either a multidimensional scheme or flux-splitting. For reasons of simplicity we have chosen for the second approach. The order of fluxing is: first in x-, then y- and then in z-direction. Currently the flux-splitting of [4] is adopted. As reported in [4], undershoots and/or overshoots can still occur, which leads to unphysical values of Ψ, namely < 0 and > 1. If these values are replaced by 0 and 1 respectively, mass errors arise which are of order 10^{-4}. This is also experienced in the current research. Mass errors are completely avoided by redistributing Ψ.

Step 3: Inverse function Having found a new Volume-of-Fluid 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 $(\Phi_1, \Phi_2, \dots)$, such that

$$|f(\Phi_k^{n+1}, \nabla\Phi_k^{n+1}) - \Psi_k^{n+1}| < \epsilon \qquad \forall k = 1, 2, \dots, \tag{20}$$

where ϵ is some tolerance. It will be clear that due to the behavior of Ψ multiple solutions Φ exist. 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 Volume-of-Fluid 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 Eqn. (13) with respect to argument Φ_k:

$$f(g(\Psi, \nabla\Phi), \nabla\Phi) = \Psi. \tag{21}$$

This equation is solved by means of Newton iterations.

4 Applications

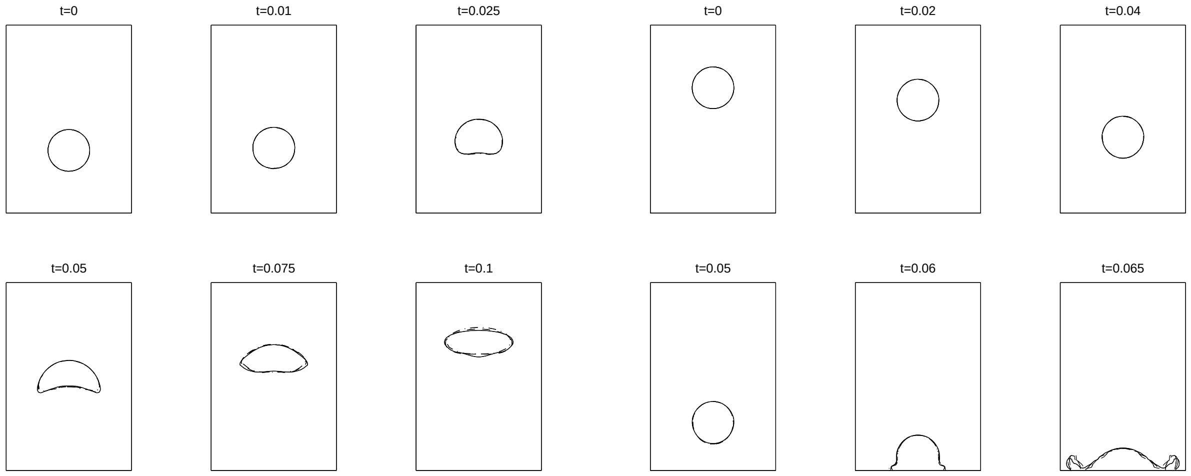
The behavior of the MCLS approach is shown by a falling drop and a rising bubble in two and three dimensions.

4.1 Two dimensions

In [7] a two-dimensional rising air bubble in water is considered. The dimensions and sizes are: $L_x = 0.02\ m$, $L_y = 1\frac{1}{2}L_x$, $R_0 = \frac{1}{6}L_x$, $x_0 = y_0 = \frac{1}{2}L_x$. The gravity and material constants are: $g = 9.8\ \frac{m}{s^2}$, $\sigma = 0.0728\ \frac{kg}{s^2}$, $\rho_w = 10^3\ \frac{kg}{m^3}$, $\rho_a = 1.226\ \frac{kg}{m^3}$, $\mu_w = 1.137\ 10^{-3}\ \frac{kg}{ms}$ and $\mu_a = 1.78\ 10^{-5}\ \frac{kg}{ms}$. where subscripts $_w$ and $_a$ indicate water and air respectively.

Results are shown in Fig. 1(a) for three different mesh sizes. We take $\epsilon = 10^{-8}$. 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 [7]. The results are the same for $t \leq 0.025$ for all mesh sizes. Thereafter small differences occur. The results compare well with [7]. The MCLS method seems to result in a more coherent structure at the highly curved part of the interface at $t = 0.05$. This is thought to be caused by the low resolution of the grids used here.

In Fig. 1(b) 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 $y_0 = L_x$. Mass conservation properties are the same as before. The result are the same until the droplet hits the



(a) Rising bubble

(b) Falling droplet

Figure 1: Air/water flows; $- \cdot -$: 30×45; $- -$: 40×60; — : 60×90 mesh

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 [7]. Note that the results in [7] span $t \leq 0.05$; no results after collision are presented.

4.2 Three dimensions

In Figs. 2 and 3 a three dimensional rising bubble bursting through a free surface is is shown for a $64 \times 64 \times 64$ and a $96 \times 96 \times 96$ grid respectively. Note that surface tension is not modeled. The material properties are the same is in the two dimensional case, except for the surface tension coefficient (which is zero). The domain is a cube with a width, length and height of 0.01 *m*. The bubble is initially placed at $1/4^{th}$ height. 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-like and ring-like structure, just before it breaks through the free surface.

In Figs. 4 and 5 a falling droplet is shown with the same settings as for the bubble. The droplet is released at half the height of the domain and the free surface is initially located at $1/4^{th}$ height. The snapshots are taken at intervals of 0.01 *sec*.

5 Conclusion

The extension of the mass Conserving Level-Set (MCLS) to three dimensions has been presented. The method is based on the Level-Set methodology, where mass is conserved by considering the fractional volume of a certain fluid within a computational cell. Mass

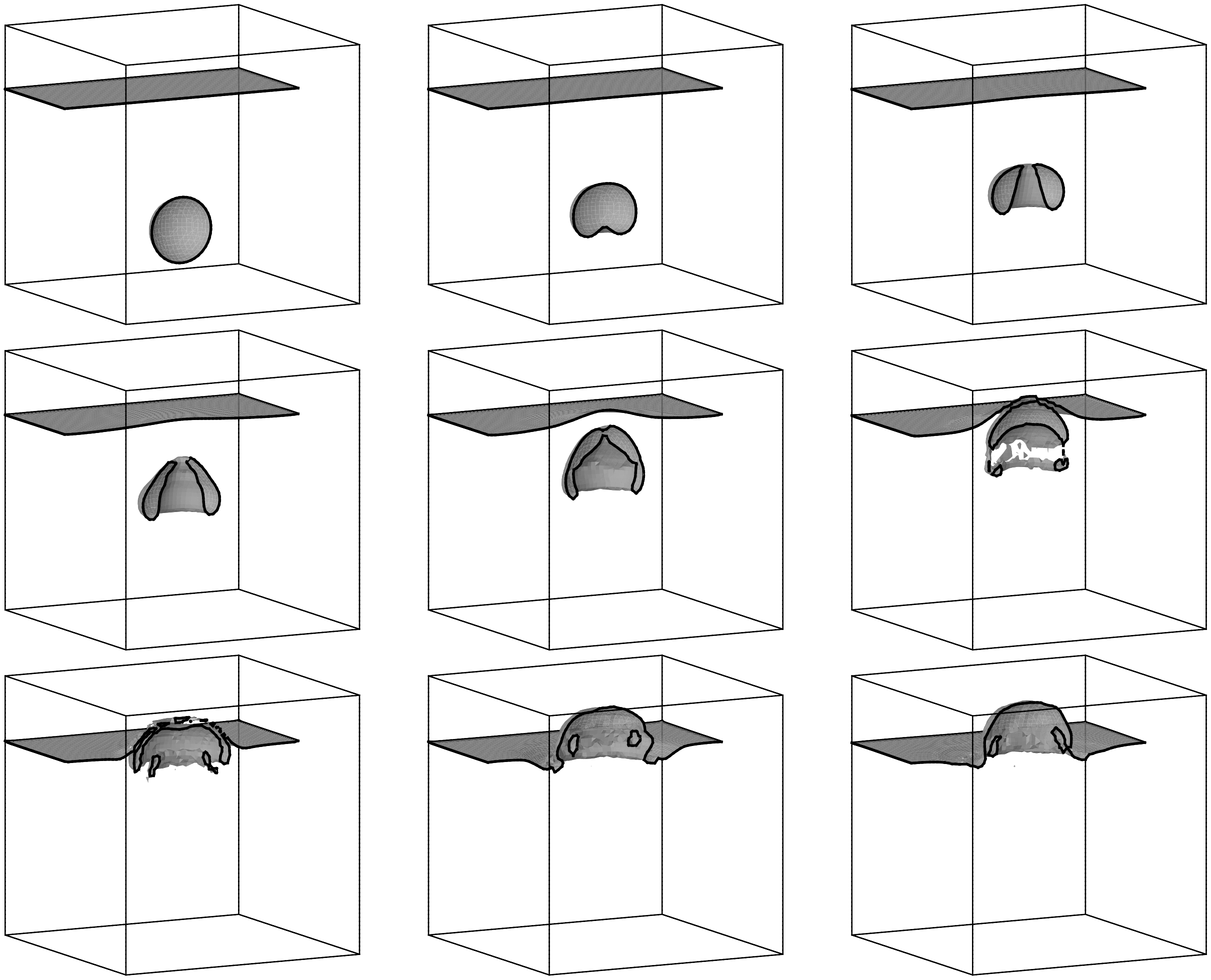


Figure 2: Rising bubble; 64^3 grid

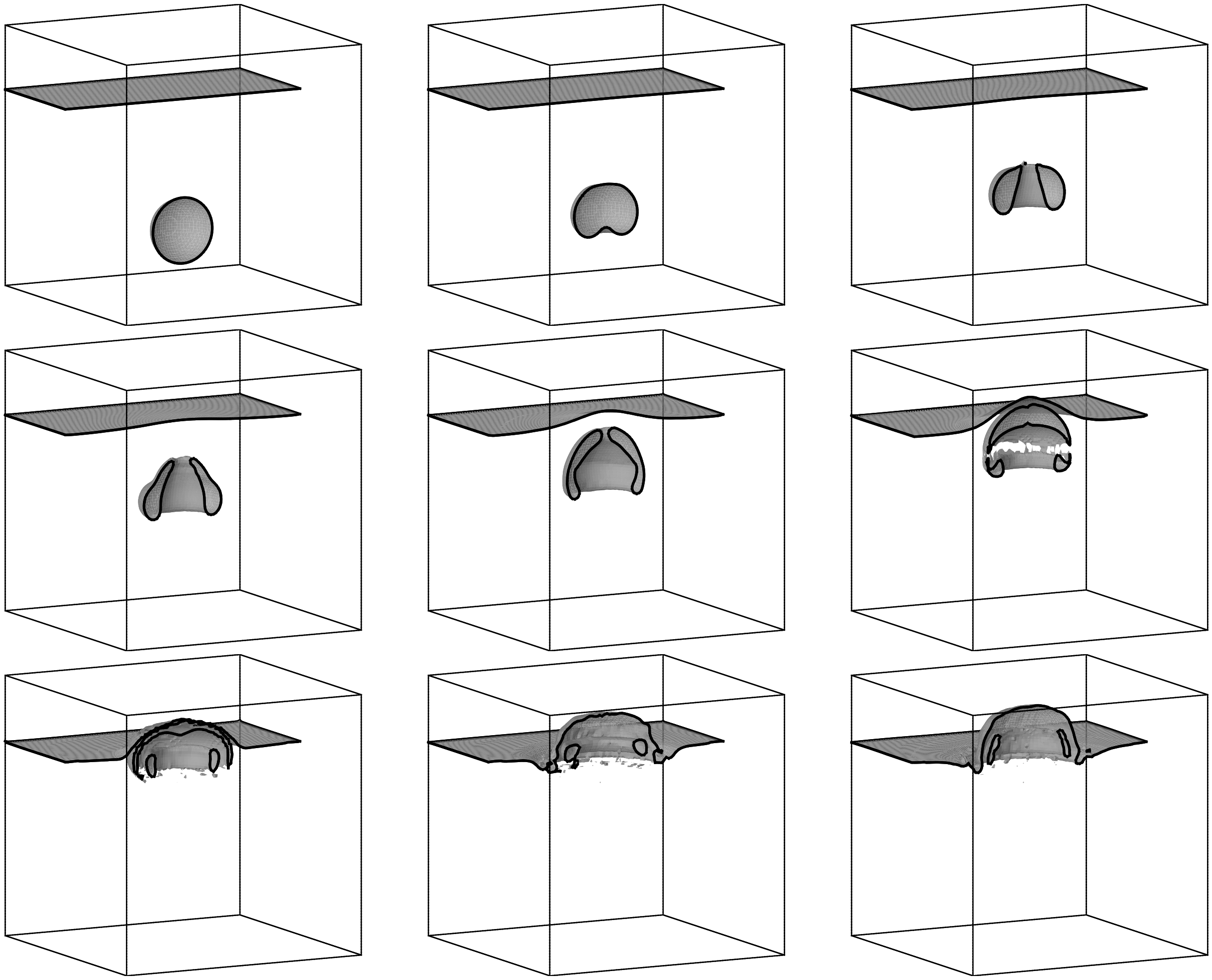


Figure 3: Rising bubble; 96^3 grid

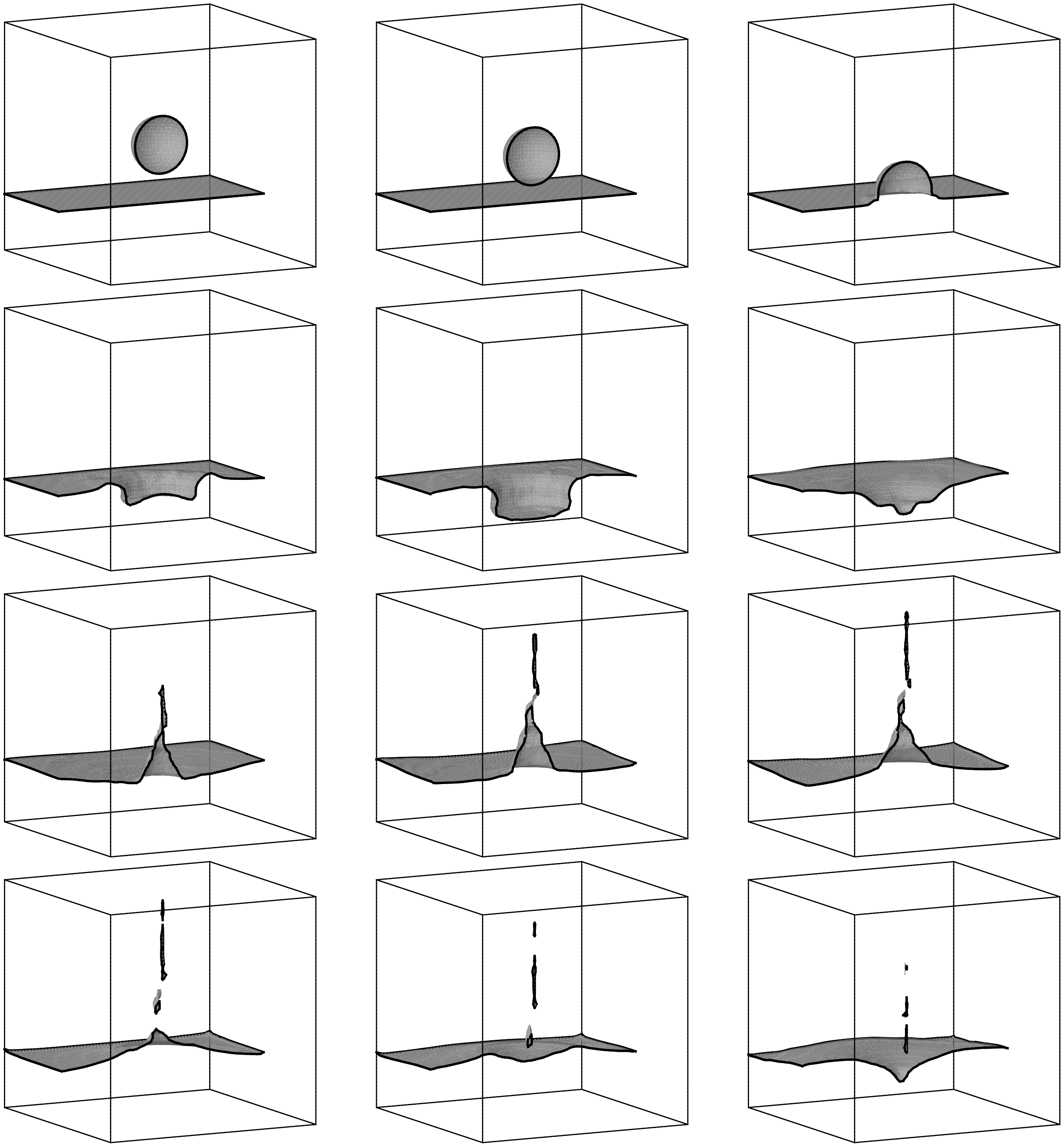


Figure 4: Falling droplet; 64^3 grid

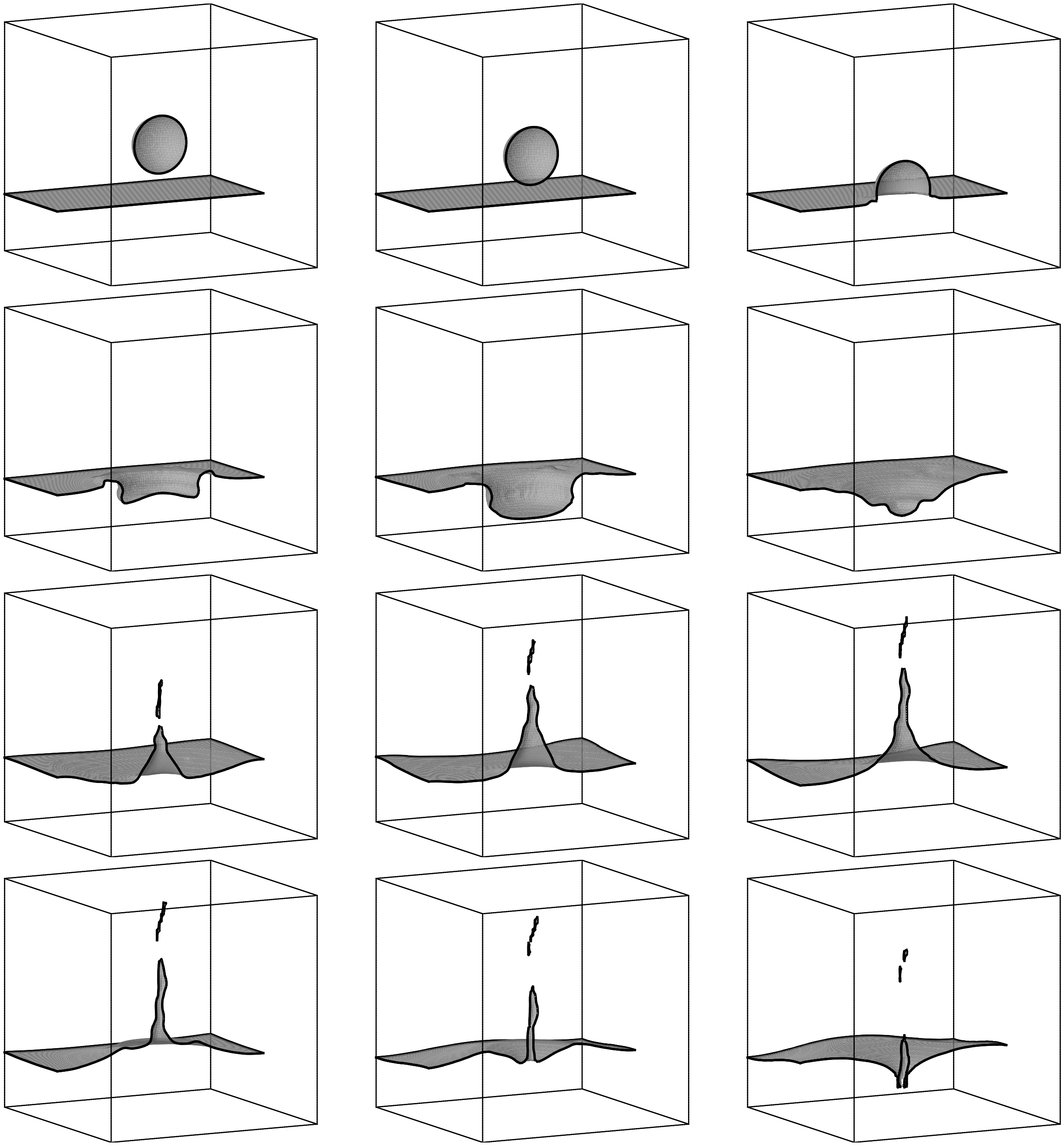


Figure 5: Falling droplet; 96^3 grid

is conserved up to a specified (vanishing) tolerance. The MCLS method combines the attractiveness of the Level-Set method with the mass-conserving properties of the Volume-of-Fluid methods, without adopting the latter. This makes the implementation 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 in two and three dimensions.

REFERENCES

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

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

[3] S.P. van der Pijl, A. Segal, and C. Vuik. A mass-conserving level-set method for modeling of multi-phase flows. AMA report 03-03, Delft University of Technology, 2003. http://ta.twi.tudelft.nl.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

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