Journal of Computational Physics 228 (2009) 4444–4467

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Numerical simulation of drop impact on a liquid–liquid interface with a multiple marker front-capturing method

Emil Coyajee, Bendiks Jan Boersma *

J.M. Burgerscentrum for Fluid Mechanics, Delft University of Technology, Laboratory for Aero and Hydrodynamics, Leeghwaterstraat 21, NL-2628 CA Delft, The Netherlands

ARTICLE INFO

Article history:
Received 17 September 2007
Received in revised form 23 December 2008
Accepted 11 March 2009
Available online 24 March 2009

Keywords:
VOF
Level-Set
Multiphase flow

ABSTRACT

The gravity-driven motion of a droplet impacting on a liquid–liquid interface is studied. The full Navier–Stokes equations are solved on a fixed, uniform grid using a finite difference/front-capturing method. For the representation of fluid–fluid interfaces, a coupled Level-Set/Volume-Of-Fluid method [M. Sussman, E.G. Puckett, A coupled Level-Set and Volume-of-Fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comp. Phys. 162 (2000) 301–337] is used, in which we introduce the novel approach of describing separate interfaces with different marker functions. As a consequence, we prevent numerical coalescence of the droplet and the liquid–liquid interface without excessive (local) grid refinement. To validate our method, numerical simulations of the drop impact event are compared with experiments [Z. Mohamed-Kassim, E.K. Longmire, Drop impact on a liquid–liquid interface, Phys. Fluids 15 (2003) 3263–3273]. Furthermore, a comparison is made with the numerical results of [A. Esmaeeli, G. Tryggvason, Direct numerical simulations of bubbly flows. Part 2. Moderate Reynolds number arrays, J. Fluid Mech. 385 (1999) 325–358] for an array of rising bubbles. The investigation shows that the multiple marker approach successfully prevents numerical coalescence of interfaces and adequately captures the effect of surface tension.

1. Introduction

For the direct numerical solution of multiple droplets in a viscous fluid, some of the most commonly used methods either belong to the class of front-capturing or front-tracking methods. Using either one of these methods to simulate droplet interaction, one is confronted with the wide range of length scales involved.

When two droplets collide, a thin film of the surrounding liquid persists between the droplets, requiring a certain drainage time before coalescence may take place. Depending on the size of the droplets and the material properties of the fluids, the thickness of the thin film can be much smaller than the radius of the droplet (Fig. 1). In a numerical simulation, representation of such thin films requires extensive grid refinement, which is exceptionally expensive when the fluid motion is computed on a fixed, uniform grid. Although local adaptivity of the mesh refinement can be considered in anticipation of the random motion of droplets in a dispersion, its implementation substantially contributes to the complexity of the algorithm. Alternatively, if the thin film is not resolved in a numerical simulation, front-tracking and front-capturing methods may give qualitatively very different results because of different interface representations.

In front-capturing methods, the moving, deformable interface is defined implicitly by a marker function on the fixed grid. As a result, standard front-capturing methods, such as Level-Set (LS) [23,2], Volume-Of-Fluid (VOF) [18,14,11] or combined

* Corresponding author. Tel.: +31 650 604 6420; fax: +31 650 604 0184.
E-mail addresses: b.j.boersma@tudelft.nl, b.j.boersma@wbmt.tudelft.nl (B.J. Boersma).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.03.014

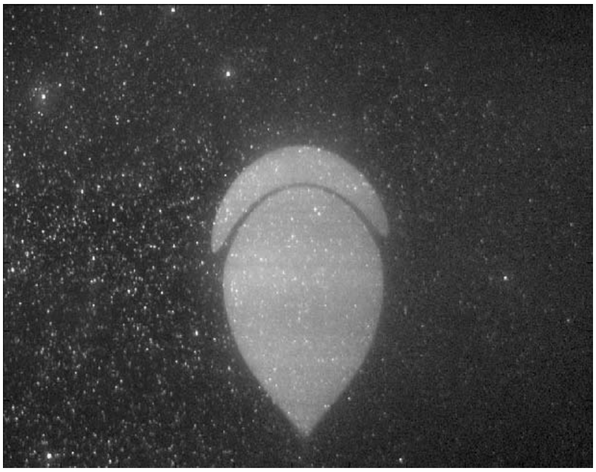


Fig. 1. Collision of two oil drops rising in a glucose–water mixture at low Reynolds and low Weber number conditions [4].

LS/VOF methods [22,25] are not capable of representing multiple interfaces in a single computational stencil. When the interfaces of different droplets collide, they merge automatically, resulting in so-called numerical coalescence of the droplets.

In case of front-tracking, the interface is defined explicitly by means of a set of logically connected particles [24]. Consequently, multiple interfaces can easily be represented in a single cell and droplet collision without coalescence is naturally simulated. To merge interfaces, special effort needs to be made. In the region of merging, some interface particles are deleted while the remaining ones need to be reconnected.

In conclusion, although front-tracking enables simulation of interfaces both with and without coalescence, standard front-capturing methods are limited to automatic merging of interfaces upon collision. Therefore, front-capturing methods have been combined with particle tracking in recent years. Combined particle/VOF [1,13] and particle/LS [5] methods have been proposed to capture interfaces at sub-cell resolution, however at the cost of introducing additional complexity compared to the original VOF or LS methods.

In this work, we propose a different strategy to simulate non-coalescing interfaces with a front-capturing method. The concept is to use separate marker functions for the interfaces of different volumes of the same fluid. It will be shown that only small adjustments to the traditional, single marker front-capturing method are required, adding little complexity to the algorithm.

For the validation of the multiple marker front-capturing method we study two cases. First, it is applied to the problem of the gravity driven impact of a droplet on a liquid–liquid interface. Essentially, this problem is equivalent to the problem of the collision of two droplets, one of finite and one of infinite radius. Our results are compared with the detailed experimental study of drop impact on a liquid–liquid interface by Mohamed-Kassim and Longmire [16]. Second, it is applied to the buoyant rise of an array of bubbles as has been studied before by [6].

The organization of the paper is as following: In Section 2, we present the governing equations for interfacial two-phase flow. Subsequently, our traditional front-capturing method is discussed, which is based on a definition of different bodies of the same fluid with a single marker function. Then, in Section 4 the concept and implementation of the multiple marker approach is introduced. Subsequently in Section 5 the drop impact study is presented. Here, following a motivation of the numerical set-up, a detailed comparison between numerical results and experimental observations is displayed. In Section 6 the buoyant rise of an array of bubbles is discussed. In Section 7, we conclude with a discussion of the possibilities and limitations for future application of the multiple marker method in the simulation of liquid–liquid dispersions.

2. Governing equations for interfacial two-phase flow

Following the front-tracking/front-capturing methodology, two-phase flow is described by a set of equations for fluid and interface motion.

The motion of the fluids is described by the incompressible Navier–Stokes equations:

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\rho(\mathbf{u}_t + \nabla \cdot (\mathbf{u}\mathbf{u})) = -\nabla p + \nabla \cdot (\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) + \rho \mathbf{g}, \tag{2}$$

where $p, \mathbf{u} = (u, v, w)^T$ and $\mathbf{g}$, respectively, denote pressure, the velocity vector and the gravitational acceleration. Subscript 't' denotes differentiation with respect to time. Density ρ and viscosity μ are denoted by ρ_1, μ_1 in fluid phase '1' and by ρ_2, μ_2 in fluid phase '2', see Fig. 2. At the interface between the fluids, the following conditions apply for continuity of velocity and stresses:

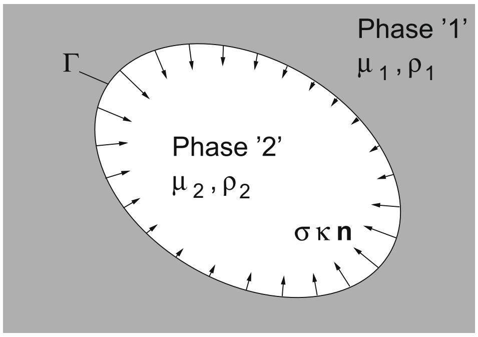


Fig. 2. Continuous fluid phase '1' and dispersed fluid phase '2' with corresponding densities ρ_1, ρ_2 and viscosities μ_1, μ_2. Surface tension acts as a force in the direction of the normal vector $\mathbf{n}$ at interface Γ and its magnitude is proportional to the surface tension coefficient σ and the local curvature κ.

$$[\mathbf{u}]_\Gamma = \mathbf{0}, \tag{3}$$

$$\left[p\mathbf{n} + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)\cdot\mathbf{n}\right]_\Gamma = \sigma\kappa\mathbf{n}, \tag{4}$$

where $[\cdot]\Gamma$ denotes a jump across the interface Γ, $\mathbf{n}$ denotes the interface normal vector, κ represents the value of the interface curvature, and σ is the surface tension coefficient, assumed to be constant.

The interface Γ between the two fluids is represented by the zero level of a marker function, denoted by ϕ, i.e. $\Gamma(t) = \{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0\}$. Away from the interface ϕ, the so-called Level-Set (LS) function, is required to be a distance function to the interface such that $\phi < 0$ in phase '1' and $\phi > 0$ in phase '2'. Consequently, the density and viscosity in 2, 4 are given by:

$$\rho = \rho_1(1 - H(\phi)) + \rho_2 H(\phi), \tag{5}$$

$$\mu = \mu_1(1 - H(\phi)) + \mu_2 H(\phi), \tag{6}$$

where the Heaviside function $H(\phi)$ is defined as follows:

$$H(\phi) = \begin{cases} 1 & \text{if } \phi > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Since the fluids are considered immiscible, the interface is a material property of the flow and its motion is described by:

$$\phi_t + \mathbf{u}\cdot\nabla\phi = 0. \tag{8}$$

3. Computational approach: single marker front-capturing method

The equations are solved numerically on a fixed, uniform Cartesian grid in a three-dimensional rectangular computational domain. Each time step the equations for the fluid velocity and interface position are integrated sequentially: the fluid velocity is advanced for given interface position and the interface is advanced given the updated flow field. The interface is advected using the Coupled Level-Set/Volume-Of-Fluid (CLSVOF) method [22]. The Navier–Stokes equations are integrated in time with a pressure correction method. To facilitate numerical treatment of the interface conditions, the discontinuous variation of viscosity at the interface is smoothed. The remaining jump conditions for the pressure and the discontinuous density are incorporated in the discretization of the Navier–Stokes equations by means of the ghost-fluid method [12].

In the following, we give a short description of the temporal and spatial discretization of the Navier–Stokes solver and the CLSVOF interface advection algorithm in case all interfaces are represented by a single marker function. The extension to a multiple marker front-capturing method is discussed in Section 4.

3.1. Temporal discretization of the Navier–Stokes equations

The Navier–Stokes equations are integrated in time using a pressure correction method [26]. Eq. (2) is split into the following predictor and corrector step:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\frac{3}{2}A(\mathbf{u}^n) + \frac{1}{2}A(\mathbf{u}^{n-1}) + \frac{1}{\rho}\left(D_{imp}(\mathbf{u}^*) + D_{exp}(\mathbf{u}^n) - Gp^{n-\frac{1}{2}}\right) + \mathbf{g}, \tag{9}$$

and

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{Gp^*}{\rho}, \tag{10}$$

where

$$p^* = p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}. \tag{11}$$

In (9), G represents the discrete gradient and A the discrete convective operator. The convective term is integrated explicitly using the second-order Adams–Bashforth method. The diffusive term is split into two parts, D_{imp} and D_{exp} which are respectively integrated in time using the implicit and explicit Euler method. By taking the divergence of (10) and requiring the incompressibility condition for the velocity at the new time level:

$$\mathrm{DIV}(\mathbf{u}^{n+1}) = 0, \tag{12}$$

we obtain a discrete Poisson equation for p^*:

$$\mathrm{DIV}\left(\frac{1}{\rho^{n+\frac{1}{2}}} Gp^*\right) = \frac{1}{\Delta t}\mathrm{DIV}(\mathbf{u}^*), \tag{13}$$

where DIV denotes the discrete divergence operator. Eq. (13) is solved for p^* with the ICCG solver, after which $\mathbf{u}^*$ is corrected toward $\mathbf{u}^{n+1}$ according to (10). Finally, the pressure is updated according to (11).

In (9), the material parameters are always derived from the interface position at $t^{n+\frac{1}{2}}$:

$$\mu = \mu(\phi^{n+\frac{1}{2}}), \quad \rho = \rho(\phi^{n+\frac{1}{2}}), \tag{14}$$

where $\mu(\phi)$ and $\rho(\phi)$ are defined by (5) and (6). Note that in (9), the pressure gradient is evaluated at $t^{n-\frac{1}{2}}$ and incorporation of jump conditions requires the interface representation at $t^{n-\frac{1}{2}}$. To include the correct jump conditions in the discretization of the Poisson problem for p^* (13), the interface representation at both $t^{n+\frac{1}{2}}$ and $t^{n-\frac{1}{2}}$ is required. This is due to the definition of p^* as the difference between $p^{n+\frac{1}{2}}$ and $p^{n-\frac{1}{2}}$ in (11).

The time step is determined from an adaptive criterion based on restrictions due to convection, surface tension an gravity, following [10,22].

3.2. Spatial discretization

The equations are spatially discretized on a fixed Cartesian grid using finite differences. A standard staggered arrangement of variables is used [8], i.e. vector components at the cell faces and scalars in the cell center. Following [23,2], the viscosity is regularized by replacing the Heaviside function $H(\phi)$ in (7) with a continuous representation, denoted $H_\alpha(\phi)$:

$$H_\alpha(\phi) = \begin{cases} 0 & \text{if } \phi < -\alpha, \\ \frac{1}{2}\left(1 + \frac{\phi}{\alpha} + \frac{1}{\pi}\sin\left(\frac{\pi\phi}{\alpha}\right)\right) & \text{if } |\phi| \leqslant \alpha, \\ 1 & \text{if } \phi > \alpha, \end{cases} \tag{15}$$

where we choose $\alpha = 3/2\,h, h$ being the uniform mesh width.

[10] show that when the viscosity is regularized, the gradients of the velocity are also continuous at the interface and condition (4) reduces to:

$$[p] = \sigma\kappa. \tag{16}$$

As a result, no jump conditions for the velocity remain and straightforward central second-order finite differences are used to approximate derivatives of the velocity, anywhere in the computational domain.

The interface condition (16) represents an explicit jump condition for the pressure which is implemented in the discretization of the pressure gradient with the ghost-fluid method [12].

In contrast to viscosity, the density is not regularized. While the density is defined at cell centers, discretization of (13) requires values of $\frac{1}{\rho}$ at cell faces. Following [12], the weighted harmonic average is used to obtain values of $\frac{1}{\rho}$ at a cell face in the vicinity of the interface.

3.3. CLSVOF interface advection

The LS interface representation described in Section 2 provides all information required for the implementation of interface conditions, e.g. by means of the ghost-fluid method: the discrete LS function implicitly defines the position of the interface on the computational grid. However, numerical methods for the advection equation of the LS function (8) do not conserve the volume of each fluid phase, which adversely affects the quality of long time simulations of incompressible fluid flow. On the other hand, interface advection with the Volume-Of-Fluid (VOF) method can be performed such that the volume of each phase is conserved [19]. In the VOF method, fluid phases are represented by the discrete volume fraction or Volume-Of-Fluid (VOF) function. The discrete VOF function, denoted by ψ, can be defined in terms of the LS function by:

$$\psi_{i,j,k}(t) = \frac{1}{\Delta x \Delta y \Delta z}\int_{\Omega_{i,j,k}} H(\phi(\mathbf{x},t))d\mathbf{x}, \tag{17}$$

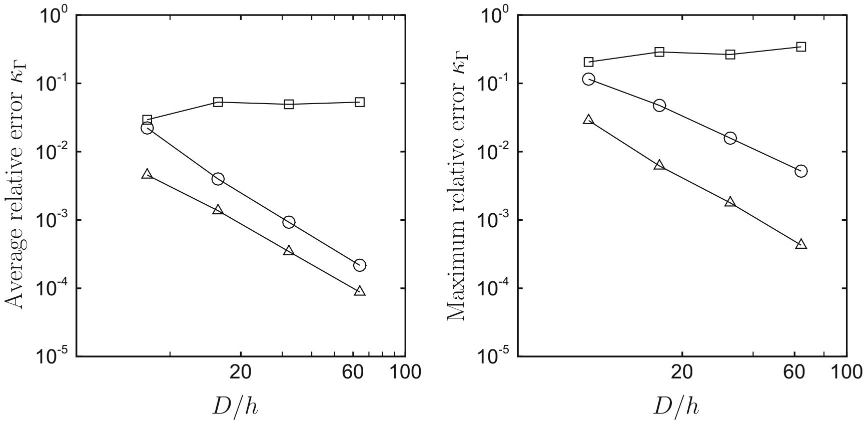


Fig. 3. Average (*left*) and maximum (*right*) relative error for the curvature of a sphere. Curvatures evaluated from the exact LS function (*deltas*), the reconstructed LS function (CLSVOF method) (*squares*) and the exact VOF function with the height-function methodology (*circles*).

where $\Omega_{i,j,k}$ denotes the volume of a computational cell. Consequently, if all fluid in cell (i,j,k) belongs to fluid phase ‘1’, $\psi_{i,j,k} = 0$. If all fluid belongs to phase ‘2’, $\psi_{i,j,k} = 1$. If the cell contains both fluid phase ‘1’ and ‘2’, $0 < \psi_{i,j,k} < 1$.

The LS and VOF approaches have been combined in the CLSVOF interface advection method [22,21]. By applying the CLSVOF method in this work, we benefit from both interface representations: the LS function is used for implementation of the interface conditions and the VOF function is used to ensure volume conservation of the interface advection.

Another approach to combine LS with VOF to obtain volume conservation is the MCLS interface advection method by [25]. The algorithm of the MCLS method is less elaborate than that of the CLSVOF method. However, we found the CLSVOF method more accurate for advection tests such as the slotted disk by [27] and the single reversed vortex test by [17]. Therefore the CLSVOF method is used in this work.

In the CLSVOF method, interface advection is governed by the following conservation equations for the LS and VOF functions, respectively[1]:

$$\phi_t + \nabla \cdot (\mathbf{u}\phi) = 0 \quad \text{and} \quad \psi_t + \nabla \cdot (\mathbf{u}\psi) = 0. \tag{18}$$

To ensure a volume conserving advancement of the fluid phases with corresponding interface position, the coupled LS and VOF advection procedure consists of the following two steps:

1. Given $\mathbf{u}^n, \psi^{n-\frac{1}{2}}$ and $\phi^{n-\frac{1}{2}}$, Eq. (18) are solved using a second-order accurate conservative operator split advection scheme to obtain $\psi^{n+\frac{1}{2}}$ and $\phi^{n+\frac{1}{2}}$. To ensure the volume conserving advection of the volume fraction function, the discrete fluxes of ψ are determined from a geometric reconstruction of the fluid phases in each cell. Assuming a piecewise linear interface in cells where $0 < \psi < 1$, the co-advected LS function is used to obtain the normal vector of the reconstructed interface.
2. The advection procedure for the LS function yields an interface representation which does not conserve the volume of each fluid phase. Therefore, the advected LS function $\phi^{n+\frac{1}{2}}$ is reinitialized to represent a distance function to the advected VOF function $\psi^{n+\frac{1}{2}}$, which does represent a volume conserving distribution of the fluid phases.The LS reinitialization is achieved by first reconstructing a piecewise linear interface in each cell for which $0 < \psi^{n+\frac{1}{2}} < 1$. The normal vector to the piecewise linear interface segments is obtained from the advected LS function, $\phi^{n+\frac{1}{2}}$. The intercept of the linear segment in the cell is obtained from the value of the advected VOF function, $\psi^{n+\frac{1}{2}}$. Finally, in each cell over a width of at least four cells at either side of the interface the LS is assigned to be the signed distance function to the reconstructed piecewise linear interface.

The advection algorithm for the time integration of Eq. (18) was implemented following [22]. The LS reinitialization algorithm was provided by Dr. Sussman, for which we express our gratitude.

3.4. Interface curvature

In the CLSVOF method, both the LS and VOF interface representations are available for the estimation of interface curvature. In principle, the curvature is most easily computed from the LS function:

$$\kappa = \nabla \cdot \mathbf{n} = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}. \tag{19}$$

[1] Note that the LS equation in (18) is equivalent to (8) since $\nabla \cdot \mathbf{u} = 0$.

However, in the CLSVOF method the LS function is reinitialized each timestep as a distance function to piecewise linear interface segments which are reconstructed from the volume fractions. Therefore, the reconstructed LS function itself is continuous but not smooth. The curvature evaluated from the reconstructed LS function with (19) does not converge with grid refinement. This is shown in the grid convergence study of Fig. 3, which compares curvature errors evaluated with (19) using the exact LS function and the reconstructed LS function for the case of a spherical interface.

Fortunately, the interface curvature can also be determined with second order accuracy from the volume fractions using the height-function methodology [9,21,14,7]. In the remainder of the paper we will base the surface tension calculation on this approach.

4. Extension to the multiple marker front-capturing method

To simulate multiple domains of the same fluid without automatic (numerical) coalescence, we extend the single marker front-capturing method to deal with multiple marker functions for separate interfaces. I.e. we still simulate two-phase flow, however using multiple LS functions to represent separate volumes of the same fluid.

Simulating L volumes of fluid phase '2', we introduce the same number of LS functions, $\phi^1, \ldots, \phi^L$, such that each volume l has a boundary defined by $\Gamma^l(t) = \{\mathbf{x}|\phi^l(\mathbf{x},t) = 0\}$ and an interior domain $\Omega^l(t) = \{\mathbf{x}|\phi^l(\mathbf{x},t) > 0\}$ (Fig. 4). In the CLSVOF method both LS and VOF functions are used to represent fluid phases. For volume l of fluid phase '2', which is represented by LS function ϕ^l, we define a corresponding VOF function, denoted by ψ^l:

$$\psi^l_{i,j,k}(t) = \frac{1}{\Delta x \Delta y \Delta z}\int_{\Omega_{i,j,k}} H(\phi^l(\mathbf{x},t))\,d\mathbf{x}, \tag{20}$$

To calculate material properties, a common LS function to all interfaces, denoted by ϕ^c, is determined:

$$\phi^c(\mathbf{x},t) = \max\left(\phi^1(\mathbf{x},t), \ldots, \phi^L(\mathbf{x},t)\right). \tag{21}$$

giving $\rho = \rho(\phi^c)$ and $\mu = \mu(\phi^c)$ in (5) and (6).

In Section 3.2 we mentioned that we use the ghost-fluid method for the surface tension forces. The extension toward multiple interfaces is straightforward. When two interfaces are present between grid points (Fig. 5), contributions from both interfaces are taken into account by adding individual jump conditions in the discretization of the pressure gradient, e.g. at $i+\frac{1}{2}$:

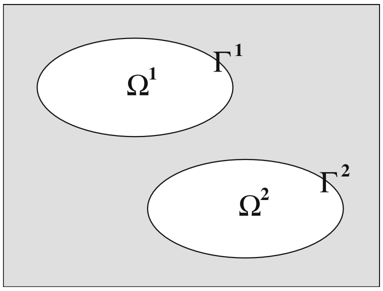


Fig. 4. Different fluid volumes, Ω^1 and Ω^2, and their boundaries, Γ^1 and Γ^2, which is defined by separate LS functions ϕ^1 and ϕ^2: $\Omega^1 = \{\mathbf{x}|\phi^1(\mathbf{x}) > 0\}, \Gamma^1 = \{\mathbf{x}|\phi^1(\mathbf{x}) = 0\}, \Omega^2 = \{\mathbf{x}|\phi^2(\mathbf{x}) > 0\}$ and $\Gamma^2 = \{\mathbf{x}|\phi^2(\mathbf{x}) = 0\}$.

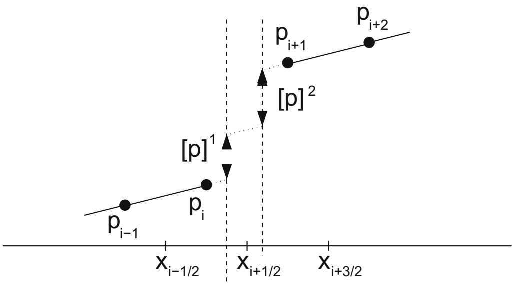


Fig. 5. One-dimensional example of the discontinuous pressure distribution in the presence of two nearby interfaces (*dashed lines*) with corresponding pressure jumps $[p]^1$ and $[p]^2$.

$$\left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2}} = \frac{p_{i+1} - p_i - [p]^1 - [p]^2}{\Delta x}, \tag{22}$$

where $[p]^1$ and $[p]^2$ denote the pressure jumps at interface 1 and 2. A general formulation for the pressure gradient at $x_{i+\frac{1}{2}}$ in case of L interfaces is given by:

$$\left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2}} = \left(p_{i+1} - p_i - \sum_{l=1}^{L}[p]^l\right)/\Delta x. \tag{23}$$

Since the pressure jump is equivalent to the effect of surface tension forces at the interface (16), $[p]^l$ is defined as follows:

$$[p]^l = \begin{cases} \sigma\kappa_\Gamma^l & \text{if } \phi_i^l \leqslant 0 \text{ and } \phi_{i+1}^l > 0 \\ -\sigma\kappa_\Gamma^l & \text{if } \phi_i^l > 0 \text{ and } \phi_{i+1}^l \leqslant 0 \\ 0 & \text{otherwise}. \end{cases} \tag{24}$$

Here κ_Γ^l corresponds to the curvature of interface l, which is evaluated using ψ^l.

Every time step, each interface is advected separately. In other words, $\phi^{l,n-\frac{1}{2}}$ and $\psi^{l,n-\frac{1}{2}}$ are advanced sequentially toward $\phi^{l,n+\frac{1}{2}}$ and $\psi^{l,n+\frac{1}{2}}$ using the same velocity field $\mathbf{u}^n$ within the CLSVOF advection algorithm. No explicit condition is imposed to avoid overlap of interfaces at the end of a time step. In the next section, our results show that overlap of interfaces is very small and converges with mesh refinement.

5. Gravity-driven drop impact on a liquid–liquid interface

In this section, the multiple marker front-capturing method is applied to the case of gravity/buoyancy-driven impact of a drop on a liquid–liquid interface. A droplet is considered, settling (under influence of gravity) or rising (due to buoyancy) in a continuous liquid until it approaches the quiescent interface between the continuous liquid and another layer of the droplet fluid.

Fig. 6 shows images of buoyancy-driven drop impact from the study by [4]. Upon impact, the droplet deforms the interface (Fig. 6(a)). The interface is deformed until the capillary pressure exerted by the deformed interface is sufficiently large to rebound the droplet (Fig. 6(b)). Meanwhile, a thin film of the continuous liquid prevents coalescence of the droplet with the secondary layer of droplet fluid. Depending on conditions, the droplet oscillates one or more times during rebound Fig. 6(c)) until it eventually settles to a quasi steady state (Fig. 6(d)). This 'rest position' is maintained for a *significantly longer time* than the time required for the dynamic rebound behavior, until the thin film of the continuous liquid is sufficiently drained, break-up of the film occurs and coalescence of the droplet with the second layer takes place.

Mohamed-Kassim and Longmire [16] have observed similar behavior for the gravity-driven impact of a drop on a liquid–liquid interface. In fact, they find that for a wide range of conditions $(0.3 < Re < 300,\ 0.005 < We < 8.4)$, film drainage time is significantly longer than the time for which the dynamic behavior of the droplet takes place. This implies that gravity-driven drop impact is a relevant problem to study the dynamics of droplet collision and rebound without film rupture. Note that drop impact on a planar liquid–liquid interface corresponds to the collision of two drops, respectively of finite and infinite radius.

Additional previous work concerning the gravity or buoyancy-driven impact of a drop on a liquid–liquid interface include the investigations by [3,20,15]. The numerical studies by [3,20] predominantly focus on characterization of drainage regimes for the thin film between the droplet and the interface, respectively for low and moderate Reynolds number conditions. In the work by [15], a combined numerical and experimental study is performed, studying the motion of bubbles, drops and spheres through liquid–liquid interfaces at low Reynolds number conditions. [28] recently introduced an adaptive finite element/Level-Set method to study break-up and coalescence of drops liquid flow. Here, two-dimensional simulations of gravity-driven drop impact are performed at low and moderate Reynolds number conditions.

The current investigation directly compares the results of three-dimensional numerical simulations with experimental results by Mohamed-Kassim and Longmire [16] for gravity-driven drop impact at moderate Reynolds numbers. The primary objective is validation of the multiple marker approach for the simulation of interface dynamics during collision.

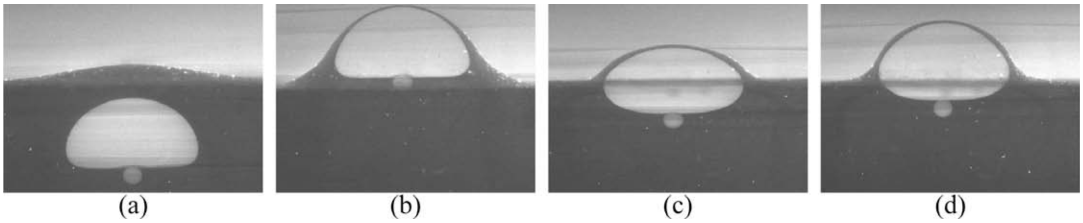


Fig. 6. Buoyancy-driven impact of an oil droplet with an oil–water interface [4]. Visualization of the lighter, colored droplet phase is performed with laser induced fluorescence.

In the following, first the experimental study for the validation of numerical results is presented. Second, some essential information on the numerical set-up and its requirements are discussed. Subsequently, numerical and experimental results are compared for the droplet trajectory and interface deformation. Finally, the evolution of the flow field is presented and discussed.

5.1. Experimental set-up and conditions

Simulations with the multiple marker method are performed for two sets of parameters, corresponding to conditions of the experimental study by Mohamed-Kassim and Longmire [16].

In the experiment a container of 40 cm^2 cross-section and 30 cm height is filled with two layers of liquid. The bottom layer is a mixture of distilled water and glycerin, 13 cm in height, the top layer is silicon oil, 10 cm in height. At the top of the silicon oil layer an amount of the water/glycerin mixture is injected, creating a droplet with a sphere equivalent diameter of 1.03 cm. Material properties are listed in Table 1. The water/glycerin mixture is dyed to distinguish between the two liquids. Both liquids are seeded with small particles to enable fluid velocity measurements using Particle Image Velocimetry (PIV). Therefore, processed images of the drop impact event provide not only droplet motion and deformation but also fluid velocity data. Two different kinds of silicon oil are used for the phase surrounding the droplet. Consequently, the study by Mohamed-Kassim and Longmire [16] provides an extensive amount of experimental data to validate our numerical method.

Table 1
Material properties [16].

	Case 1		Case 2		Units
	Silicon oil	Water/glycerin	Silicon oil	Water/glycerin	
ρ	949	1128	960	1131	kg/m^3
μ	19	6.3	48	6.7	mPa s
σ	29.1		29.5		mN/m

Table 2
Experimental parameters [16].

	Case 1	Case 2	Units
w_i	13.2	9.8	cm/s
D	1.03	1.03	cm
t_i	0.078	0.105	s
Re_i	68	20	–
We_i	5.9	3.2	–
Fr_i	0.91	0.53	–
ζ	1.189	1.178	–
λ	0.332	0.140	–

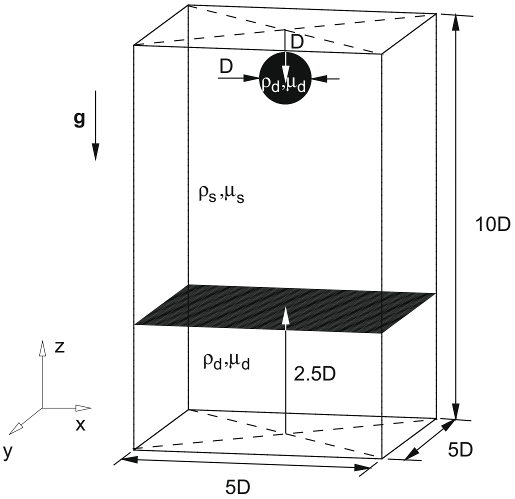
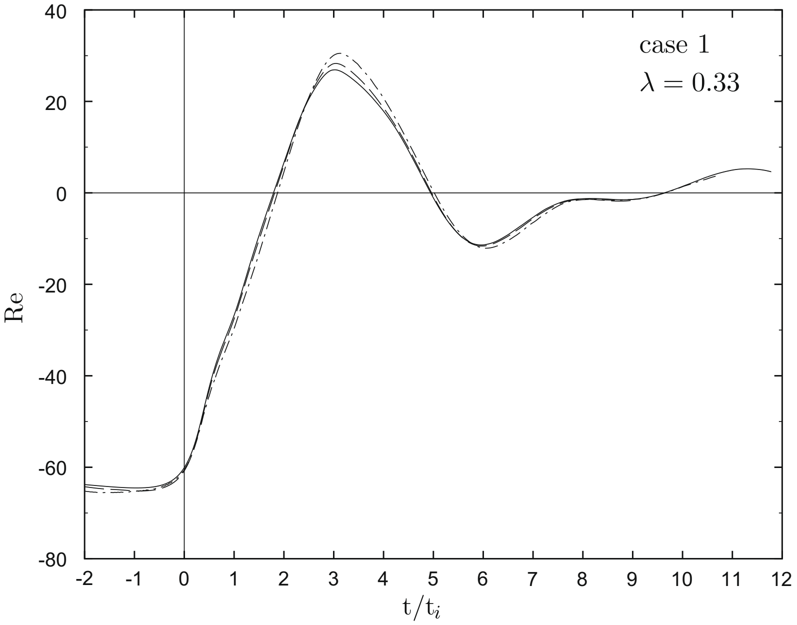


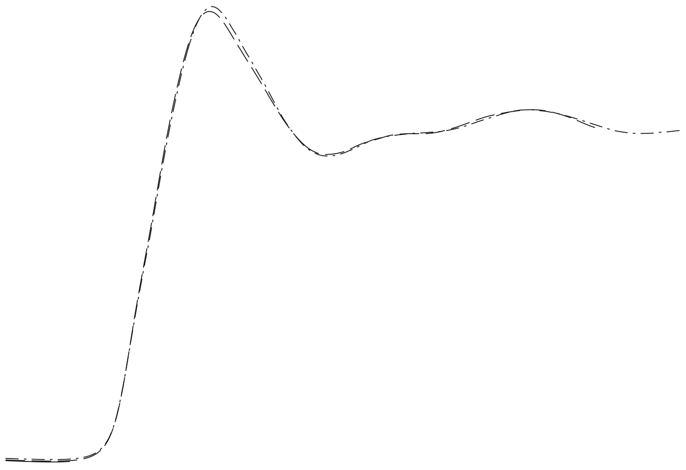
Fig. 7. Computational domain with initial configuration of interfaces. Subscripts '*d*' and '*s*', respectively, denote material properties of the droplet (water/glycerin) phase and the phase surrounding the droplet (silicon oil).

In the following sections, dimensionless variables are presented using the scaling from [16] to facilitate direct comparison of numerical and experimental results: length is scaled with the sphere equivalent droplet diameter D and time is scaled with the impact velocity time scale, defined by $t_i = D/w_i$. Here, the droplet impact velocity w_i is equivalent to the droplet terminal velocity. The dimensional values of D, w_i and t_i in the experiment are listed in Table 2, as well as corresponding Reynolds, Weber and Froude numbers, defined by:

$$Re_i = \rho_s w_i D/\mu_s, \quad We_i = \rho_s D w_i^2/\sigma, \quad Fr_i = \rho_s w_i^2/\Delta\rho g D, \tag{25}$$

where the subscript ‘s’ denotes material properties belonging to the silicon oil phase surrounding the droplet. Note that the Eötvös number, $Eo = \Delta\rho g D^2/\sigma$, is almost identical for both cases: For case 1, $Eo = 6.4$ and for case 2, $Eo = 6.0$.





5.2. Numerical set-up

The geometry of the numerical set-up is displayed in Fig. 7. In the initial condition the droplet shape is spherical and all fluid is at rest. At the boundaries of the computational domain, the no-slip condition is used for the fluid velocity.

In the numerical set-up, the size of the computational domain is significantly smaller than the fluid container in the experiment by Mohamed-Kassim and Longmire [16]. On the one hand, a larger domain size is preferred for optimal resemblance of conditions in the comparison with experimental results. On the other hand, accurate simulation of the impact event requires adequate numerical resolution in the vicinity of the droplet interface. Important constraints in balancing these requirements are the restriction of a uniform computational grid in the current method and a restriction on the number of computational cells because of limited computational resources. To motivate the current domain size and grid resolution, their influence is investigated and the following results are reported:

- The droplet needs a certain approach distance to reach terminal velocity before impact. Simulations have been performed to investigate the optimal distance in three different domains with heights of $8D, 10D$ and $12D$, a cross-section of $5D^2$ and a resolution $D/h = 16$. The height of the bottom liquid layer is maintained at $2.5D$ in these different cases. The current choice of $10D$ (see Fig. 7) is found adequate as it leads to a maximum droplet velocity within 2% compared to the result obtained at $12D$. More importantly, the interface dynamics after impact are indistinguishable, which is not the case when the total domain size is only $8D$.
- To investigate the influence of the grid resolution on the results, simulations are performed for $D/h = 16$, 24 and 32 in a domain of dimensions $5D \times 5D \times 10D$. In Fig. 8, time evolution of the droplet velocity is displayed for these resolutions. Here, the impact time $t/t_i = 0$ is defined by the instant when the lower drop surface crosses the quiescent interface level. The results show regular convergence and essentially the same behavior during the impact event for all resolutions. However, additional investigation of the drop deformation reveals that a minimum resolution of $D/h = 24$ is required to adequately capture the details of the interface dynamics after impact ($t/t_i = 0$).

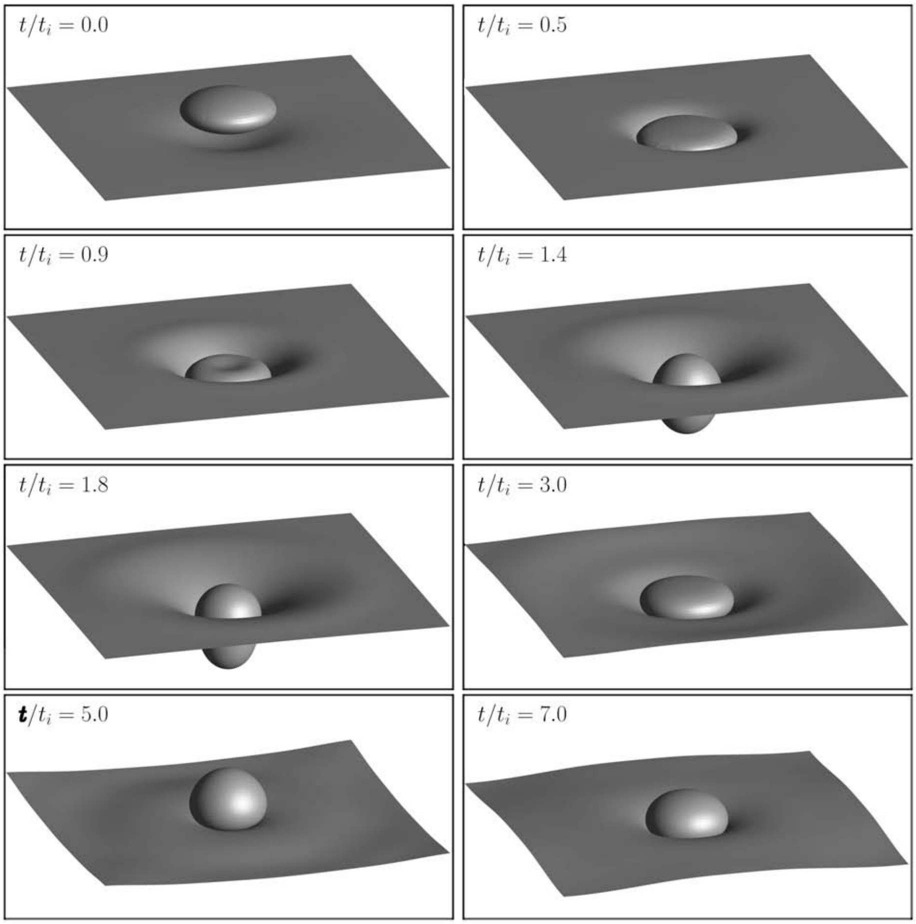


Fig. 9. Snapshots of the droplet and liquid–liquid interface surfaces from simulation of case 1.

Table 3
Grid convergence study of overflow error.

D/h	Case 1 ($\lambda = 0.33$)	Case 2 ($\lambda = 0.14$)
	$\max(\psi^1 + \psi^2 - 1)$	
16	1.4×10^{-1}	5.8×10^{-2}
24	7.6×10^{-2}	3.6×10^{-2}
32	2.6×10^{-2}	1.4×10^{-2}

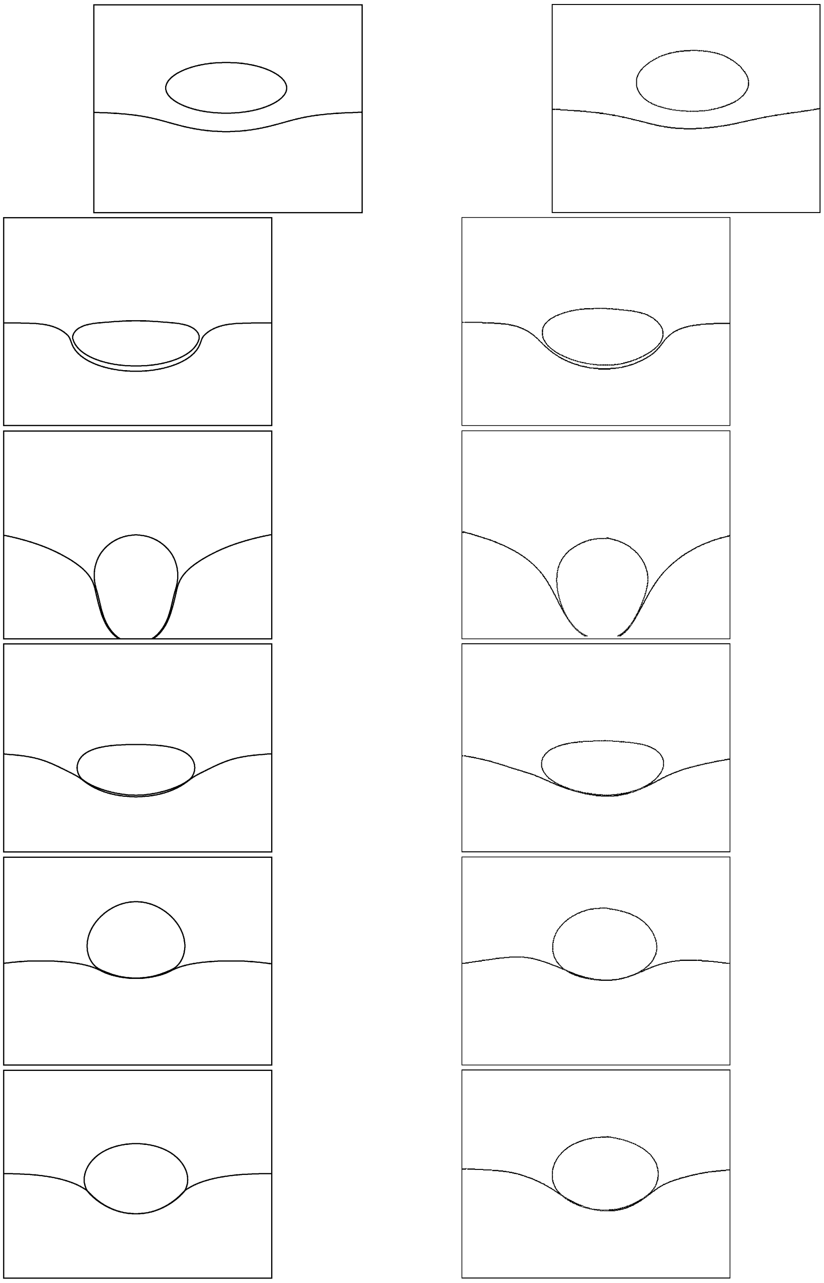


Fig. 10. Time evolution of drop impact for case 1 ($\lambda = 0.33$), comparing numerical (*left*) and experimental results (*right*). Figures display contours of the interfaces over a mid-section of the domain at $t/t_i = 0.0$, 0.5, 1.8, 3.0, 5.0, 7.0 (–).

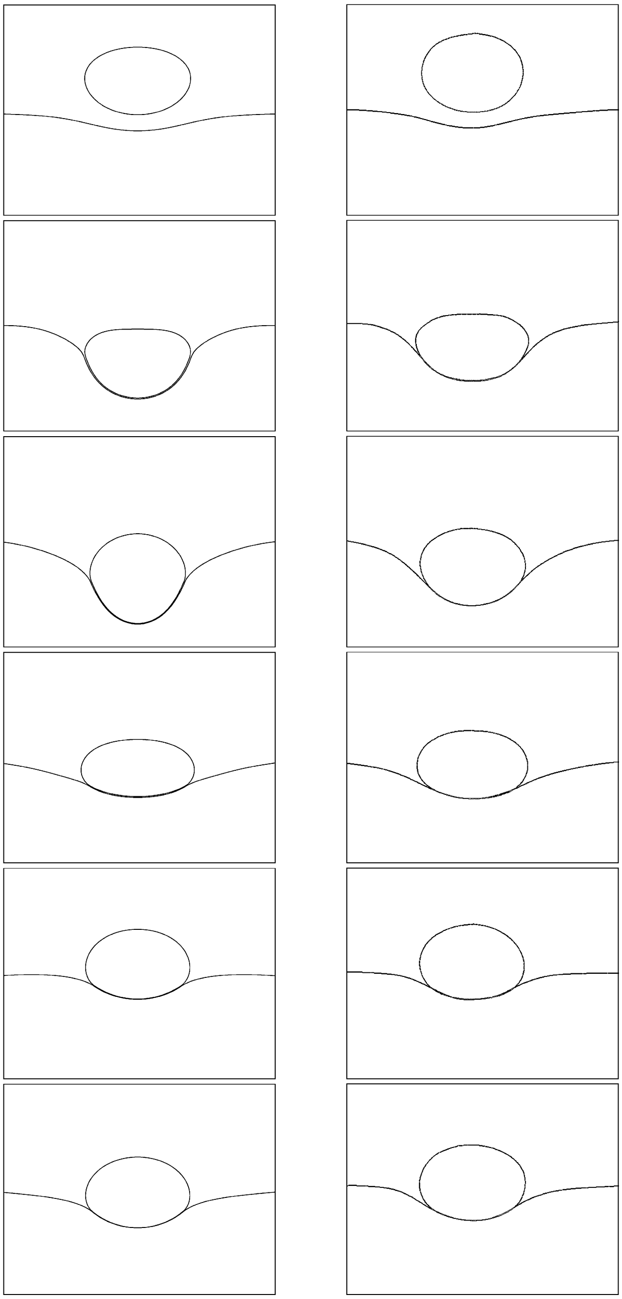


Fig. 11. Time evolution of drop impact for case 2 ($\lambda = 0.14$), comparing numerical (*left*) and experimental results (*right*). Figures display contours of the interfaces over a mid-section of the domain at $t/t_i = 0.0, 0.9, 1.4, 2.2, 3.9, 6.0$ (–).

In the following, only results computed on the finest grid ($D/h = 32$) are presented. For the selected domain size and the finest resolution the total mesh consists of $8.2 \cdot 10^6$ computational cells ($160 \times 160 \times 320$). By distributing the computational domain over 80 nodes on the SGI3800[2] at SARA, a single timestep is completed in approximately 55 s. As the total number of time steps amounted to approximately 15,000, both computations are performed within 10 days. To conclude on the numerical performance of the code, a maximum variation of mass between fluid phases of 2.5×10^{-5} and a maximum discrete divergence of the fluid velocity of 1.2×10^{-8} is reported for the fine grid simulations of both sets of parameters.

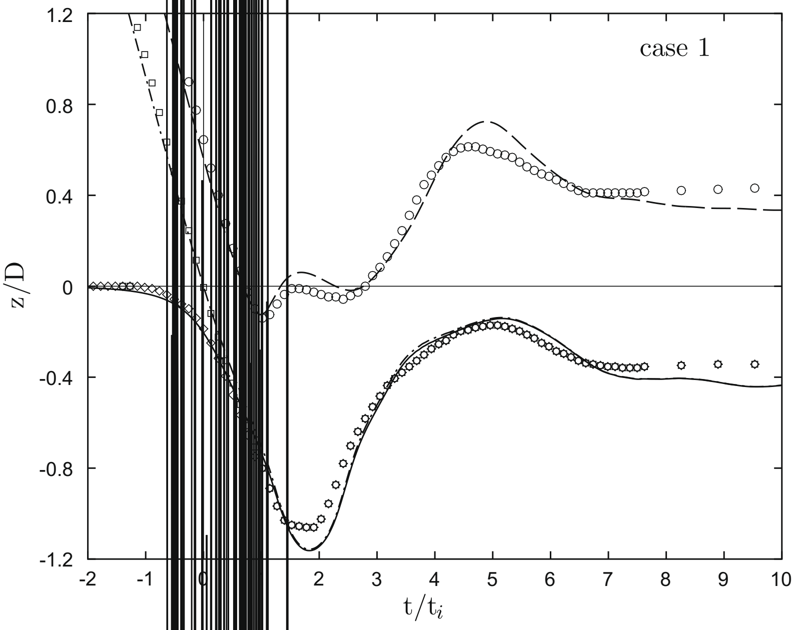
[2] 500 MHz per node.

5.3. Results

In the experimental study by Mohamed-Kassim and Longmire [16], an extensive discussion of the drop impact event is found. In the following sections, focus is on validation of the numerical approach. Results of the numerical simulations are compared with figures reproduced from the experimental data by Mohamed-Kassim and Longmire [16].

5.3.1. Drop trajectory and interfacial deformation

In Fig. 9, three-dimensional snapshots of the droplet and liquid–liquid interface are displayed at various time instances during the simulation of the impact for the parameters of case 1 (Table 1). Although no explicit condition is used to avoid overlap of separate colliding domains of the same fluid during the advection procedure, the maximum excess of the combined volume fraction in a cell during the simulation, defined by:

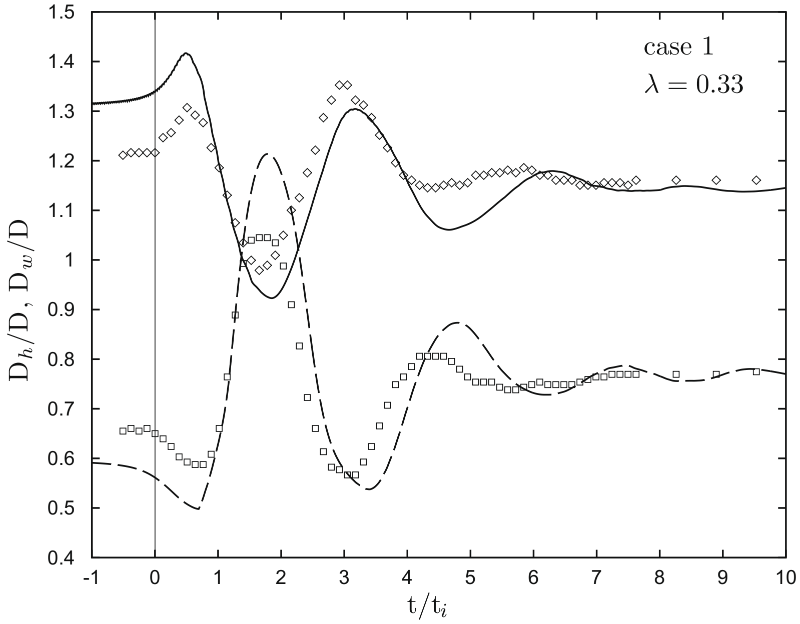


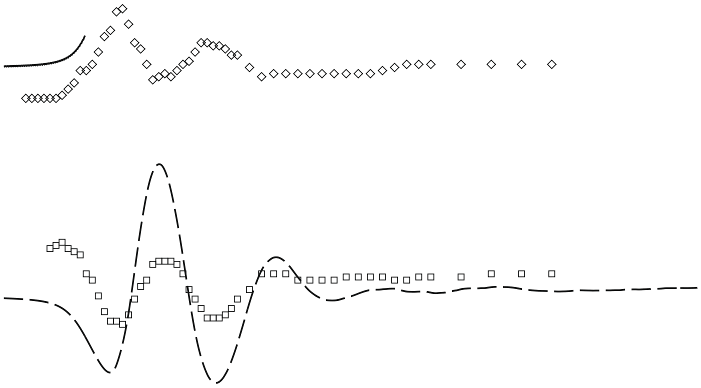
$$\max_{i,j,k}(\psi^1_{i,j,k} + \psi^2_{i,j,k} - 1),$$

is found to be small and to converge with mesh refinement (Table 3).

For direct comparison of numerical and experimental results, contour lines are presented of the droplet and the liquid–liquid interface over a vertical cross-section through the center of the domain in Figs. 10 and 11. Good qualitative agreement between the drop deformation in the simulation and experiment is found, but the magnitude of the simulated drop deformation appears slightly larger compared to the experimental results for both cases.

Further quantitative comparison is displayed in Fig. 12, showing time evolution of the normalized centerline locations of the upper and lower droplet surface and the liquid–liquid interface. Good agreement is found for $\lambda = 0.33$ (case 1). For $\lambda = 0.14$ (case 2), penetration of the droplet through the liquid–liquid interface is significantly larger in the simulation than





in the experiment. Also for case 2, a slight oscillation of the upper droplet surface between $t/t_i = 1$ and $t/t_i = 3$ is observed in the simulation, similar to case 1, which is not found in the experiment.

The evolution of the droplet width and height is presented in Fig. 13, confirming the qualitative agreement of the drop deformation suggested by Figs. 10 and 11. In agreement with the centerline locations of the lower and upper droplet surface (Fig. 12), whose difference is equivalent to droplet height D_h/D, the largest quantitative difference between the numerical and experimental droplet width D_w/D is found for case 2.

In conclusion, the results in Figs. 10–13 show that the multiple marker front-capturing method is effective for the simulation of drop impact without artificial (numerical) coalescence. Good qualitative agreement of the droplet dynamics and interface deformation is found. In the next section the investigation is extended with fluid velocity data to study the origin of the observed quantitative differences in detail.

5.3.2. Flow evolution

During the initial stage of impact ($t/t_i < 1$), the droplet decelerates while fluid is pushed out of the gap between the lower drop surface and liquid–liquid interface, until only a thin film of the ambient fluid remains. In the experiment, the thin film is

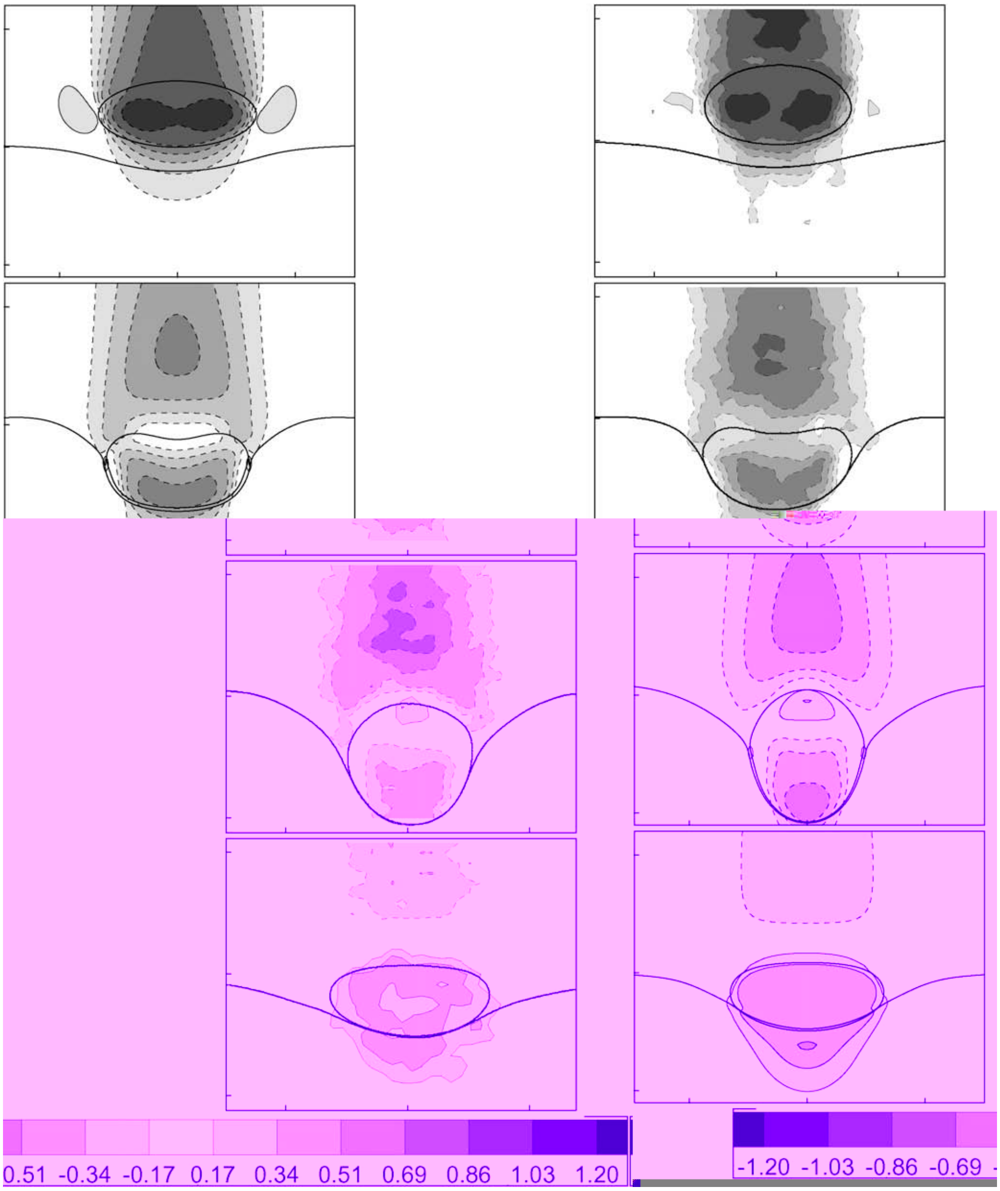


Fig. 14. Normalized vertical velocity contour plots for case 1 ($\lambda = 0.33$), comparing numerical (*left*) and experimental results (*right*). Figures are displayed at $t/t_i = 0.0$, 0.9, 1.4, 3.0 (–).

reported to vary between 0.1 mm and 0.4 mm during the droplet rebound, equal or below the width of the numerical mesh. Therefore, after the initial stage of the impact, insufficient resolution is available in the simulation to resolve fluid motion in the thin film between the droplet and the liquid–liquid interface. In the previous section it is noted that especially in case 2, the initial penetration of the lower droplet surface into the liquid–liquid interface is relatively large compared to the experimental observation (Fig. 12). On the one hand, this could be caused by the higher impact energy in the simulation: Fig. 8 displays that the simulated maximum droplet velocity Reynolds number $Re_i = 22$, while in the experiment $Re_i = 20$. On the other hand, lack of numerical resolution in the thin film region obviously leads to under-prediction of viscous dissipation as the gap between the lower drop surface and the interface is smaller than the width of the mesh right after impact time $(t/t_i = 0)$.

After the initial stage of impact, fluid from the droplet wake continues to impinge on the droplet due to inertia. An impression of fluid motion in the wake is given in Figs. 14 and 15, presenting contour plots of the normalized vertical veloc-

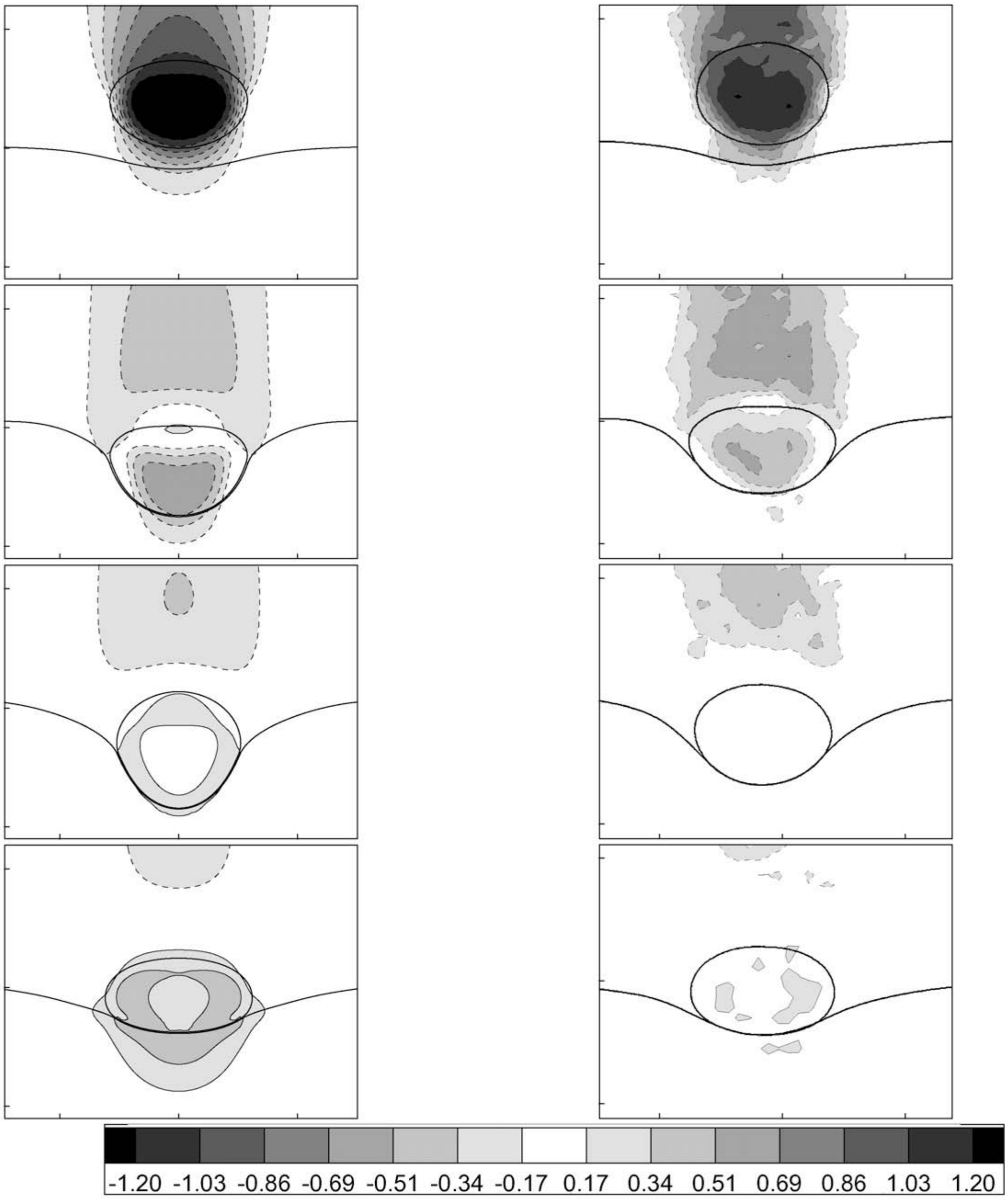
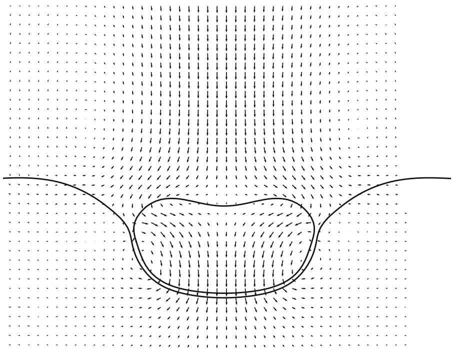


Fig. 15. Normalized vertical velocity contour plots for case 2 ($\lambda = 0.14$), comparing numerical (*left*) and experimental results (*right*). Figures are displayed at $t/t_i = 0.0, 0.9, 1.4, 2.2$ (–).

ity component w/w_i over a vertical cross-section of the domain for case 1 and case 2, respectively. As the droplet rebounds ($t/t_i > 1$), gradual decrease of downward momentum in the wake is observed, primarily at the upper surface of the droplet where fluid from the wake area is deflected away from the centerline. The temporal rate of this decrease is obviously higher for the case in which the viscosity of the fluid surrounding the drop, μ_s, is larger (case 2). The phenomenon of wake impingement is well represented in the simulations of case 1 and case 2, both qualitatively and quantitatively in comparison with the experimental results.

During rebound, the droplet is stretched vertically and oscillates on the liquid–liquid surface due to the effect of surface tension. In the previous section, good representation of the oscillatory motion of the upper droplet surface (Fig. 12) and the entire droplet shape (Fig. 13) was found in the simulation of case 1 in comparison with experimental observations. For the simulation of case 2, the magnitude of the droplet and interface deformation is found to be larger throughout the entire impact event in comparison with the experimental results. The difference between the performance of the numerical method for both cases may be explained by the role of the thin film between the droplet and the lower liquid layer. Table 1 shows that for case 2 the viscosity of the fluid in the thin film is 2.5 times larger than in case 1 and that the primary difference between the two cases concerns the viscosity of the fluid surrounding the droplet.



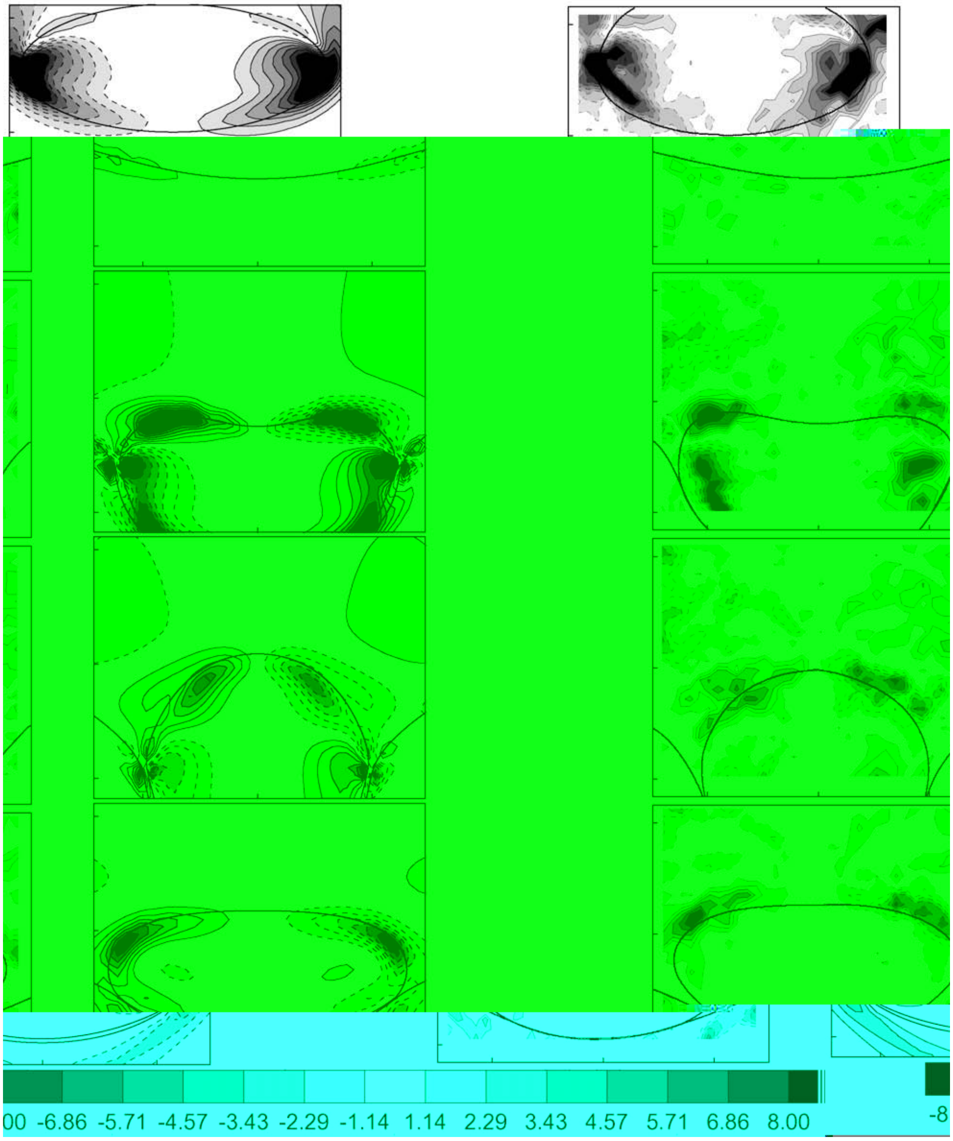


Fig. 17. Normalized vorticity contour plots for case 1 ($\lambda = 0.33$), comparing numerical (*left*) and experimental results (*right*). Figures are displayed at $t/t_i = 0.0$, 0.9, 1.4, 3.0 (–).

In the simulation, however, lack of spatial resolution results in inadequate performance of the continuum approach for the discontinuous viscosity in the representation of the boundary conditions.

A detailed account of the fluid motion and the computational resolution in the region of the thin film is provided in Fig. 16, showing instantaneous vector fields of the fluid velocity at ($t/t_i = 0.9$) for both viscosity ratios.

Note that in case of a viscosity ratio equal to unity, the simulation does not depend on the continuum approach for the representation of boundary conditions, which explains the better performance of the numerical method for case 1 ($\lambda = 0.33$) compared to case 2 ($\lambda = 0.14$).

Finally, in Figs. 17 and 18, contour plots of the normalized vorticity $\omega_y D/w_i$ are displayed over a vertical cross-section of the domain, along with contour lines of the droplet and liquid–liquid interface, for case 1 and 2. Here, positive vorticity is associated with counterclockwise rotation. The vorticity plots provide additional evidence of the proper performance of the method for the parameters of case 1.

6. Buoyant rise of bubbles at moderate reynolds number

In this section the multiple marker front-capturing method is used to simulate the rise of an array of bubbles in a periodic domain. In Fig. 19, a snapshot of the bubble array is presented. Previously, such calculations were only performed using the

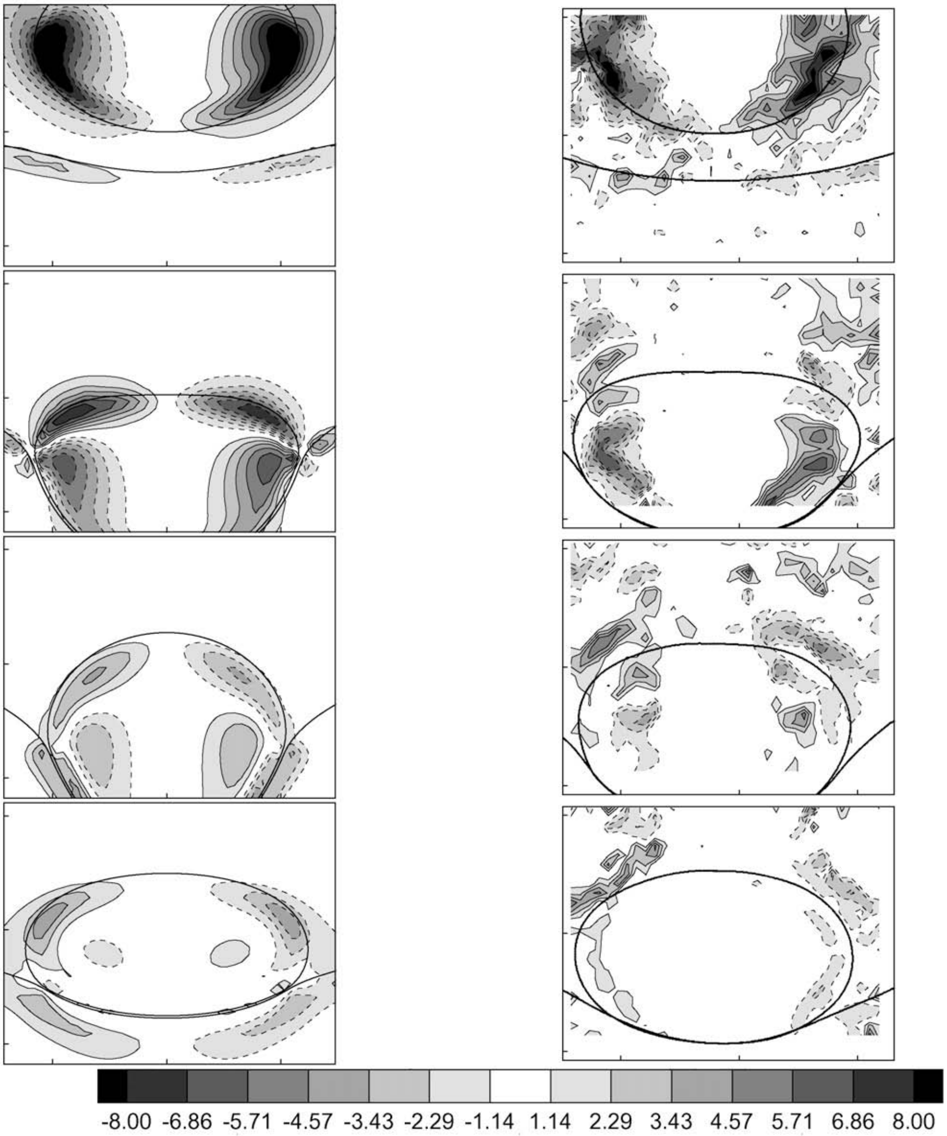


Fig. 18. Normalized vorticity contour plots for case 2 ($\lambda = 0.14$), comparing numerical (*left*) and experimental results (*right*). Figures are displayed at $t/t_i = 0.0, 0.9, 1.4, 2.2$ (–).

front-tracking methodology. To allow direct comparison with published results, our configuration and parameter set complies with those of Esmaeeli and Tryggvason [6]: The Reynolds number $Re = 29.9$, the Eötvös number $Eo = 2$, the density ratio $\zeta = 0.1$, the viscosity ratio $\lambda = 0.1$ and the volume fraction $\alpha = 0.1256$. First, results are shown for the rise of a single bubble (regular array), subsequently the results of a system of eight bubbles (free array).

In Fig. 20, time evolution of the rise velocity of a single bubble (regular array) is presented for grids containing 24^3, 48^3 and 96^3 computational cells. For these grids, the number of computational cells over the sphere equivalent bubble diameter, i.e. D/h, respectively amounts 15, 30 and 60. The rise velocity is presented as a Reynolds number, $Re = \rho_l w_b(t) D/\mu_l$, where subscripts b and l respectively denote the bubble and the surrounding liquid phase and $w_b(t)$ is the rise velocity of the bubble. The time scale is made dimensionless using $\sqrt{D/g}$. The results in Fig. 20 show good convergence for the bubble rise velocity as the difference between the 48^3 and 96^3 grid amounts only 0.5%. The estimated converged value of the terminal Reynolds number is 20.64, which corresponds well with the results by [6]. Time evolution of the bubble relative mass error, defined by:

$$\left| \frac{\sum_{i,j,k} \psi_{i,j,k}(t) - \sum_{i,j,k} \psi_{i,j,k}(t=0)}{\sum_{i,j,k} \psi_{i,j,k}(t=0)} \right|, \tag{26}$$

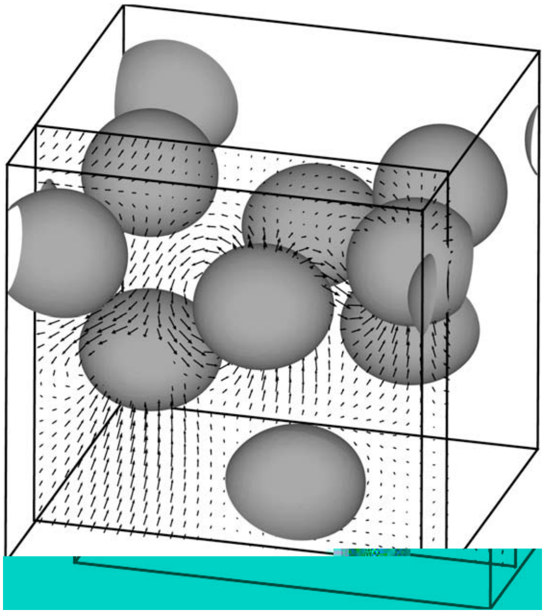


Fig. 19. Buoyant rise of eight droplets in a periodic domain ($Re = 29.9, Eo = 2, \zeta = 0.1, \lambda = 0.1$ and $\alpha = 0.1256$). In the cross-sectional plane, one out of four vectors in each coordinate direction is shown as the computational grid amounts 128^3 cells.

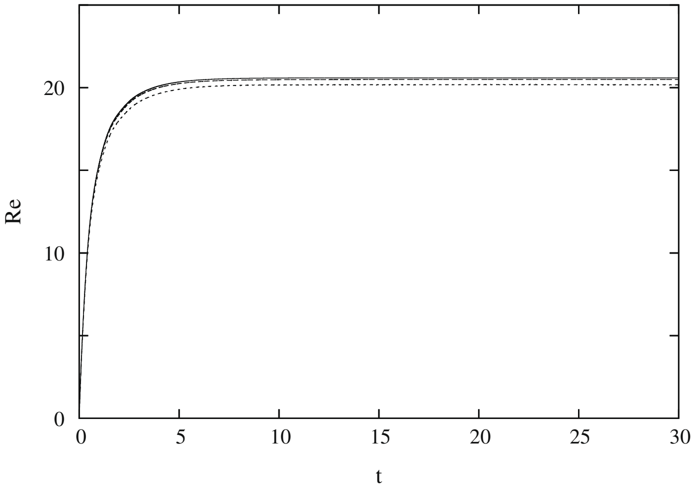


Fig. 20. Time evolution of the Reynolds number of a single bubble in a regular array on three grids: 24^3 (*dotted*), 48^3 (*dashed*) and 96^3 (*solid*).

is displayed in Fig. 21. The results show excellent convergence of the relative mass error and a very small mass loss over a considerable time on any of the grids.

The simulation of the eight bubble system (free array) is performed on a grid of 128^3 computational cells, which corresponds to a resolution of approximately 40 computational cells over the sphere equivalent bubble diameter. In the initial condition, the bubble positions are perturbed slightly compared to a regular cubic array by an off-set equal to 0.04D, however the initial velocity field is unperturbed ($\mathbf{u} = \mathbf{0}$). In Fig. 22, time evolution of the rise velocity of the bubbles is presented, comparing the results of the free array (eight bubbles) with the case of the regular array (single bubble). The rise velocity is once more presented as a Reynolds number, $Re = \rho_l w_b(t)D/\mu_l$, however with $w_b(t)$ the average rise velocity of the bubbles, defined by:

$$w_b(t) = \frac{1}{N_b}\sum_{i=1}^{N_b} w_{b_i}(t). \tag{27}$$

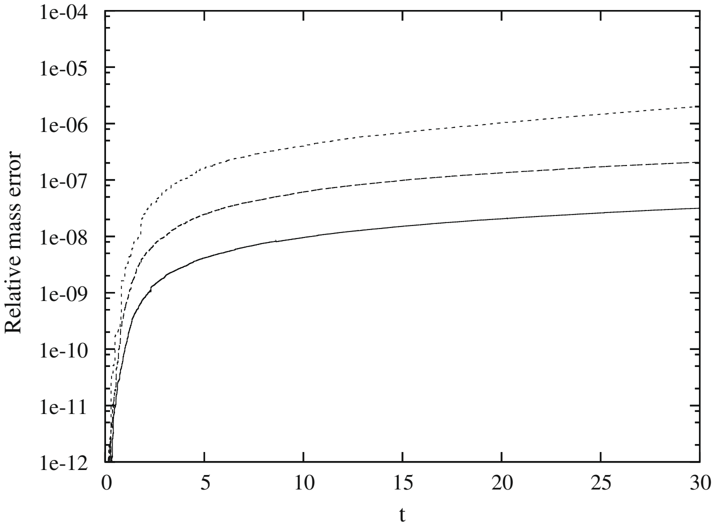


Fig. 21. Time evolution of the relative mass error of a single bubble in a regular array on three grids: 24^3 (*dotted*), 48^3 (*dashed*) and 96^3 (*solid*).

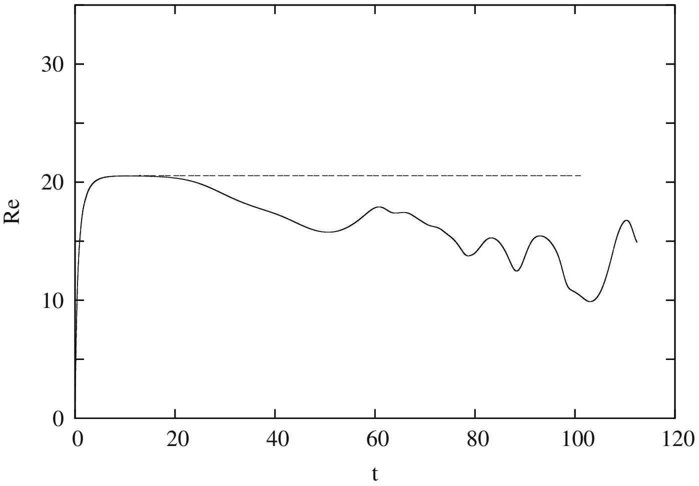


Fig. 22. Time evolution of the average Reynolds number of rising buoyant bubbles, comparing results of the free array (eight bubbles, *solid*) with the regular array (single bubble, *dashed*).

Here w_{b_i} is the velocity of bubble i and N_b is the number of bubbles in the system. Similar to [6], the results in Fig. 22 show that the rise velocity of the free array is smaller than the rise velocity of the regular array. Compared to the results by [6], the difference in rise velocity between the regular and freely evolving array appears to be somewhat larger in magnitude. We note however that an accurate time-averaged rise velocity of the bubbles can not yet be determined because of the limited extent of this computation. While the total simulation time is equivalent to the simulation time by [6], the dynamics of the freely evolving array appear to require more time to develop. This conclusion is supported by Figs. 23 and 24. In Fig. 23, the average velocity of the bubbles (also referred to as the velocity of the centroid of all the bubbles) is presented, displaying each component of the velocity vector which is made dimensionless using $\sqrt{Dg}$. Fig. 23 shows that the average horizontal velocity components of the bubble system, which are approximately one order in magnitude smaller than the vertical component, only start to develop from $t = 55$. In Fig. 24, the vertical and horizontal velocity fluctuations of the bubbles are presented relative to the instantaneous centroid velocity of the bubbles, respectively denoted k'_v and k'_h and defined by:

$$k'_v(t) = \frac{1}{N_b}\sum_{i=1}^{N_b}(w_{bi}(t) - w_b(t))^2, \tag{28}$$

$$k'_h(t) = \frac{1}{N_b}\sum_{i=1}^{N_b}(u_{b_i}(t) - u_b(t))^2 + (v_{b_i}(t) - v_b(t))^2. \tag{29}$$

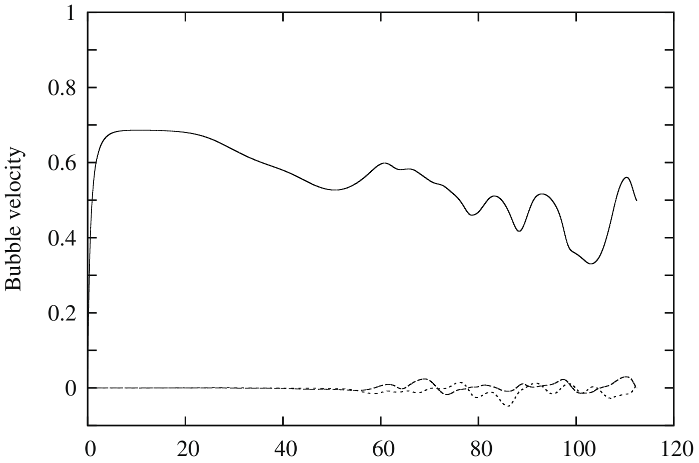


Fig. 23. Dimensionless velocity of the centroid of all bubbles, with horizontal components of the velocity vector u_b (*dashed*) and v_b (*dotted*), and vertical component w_b (*solid*).

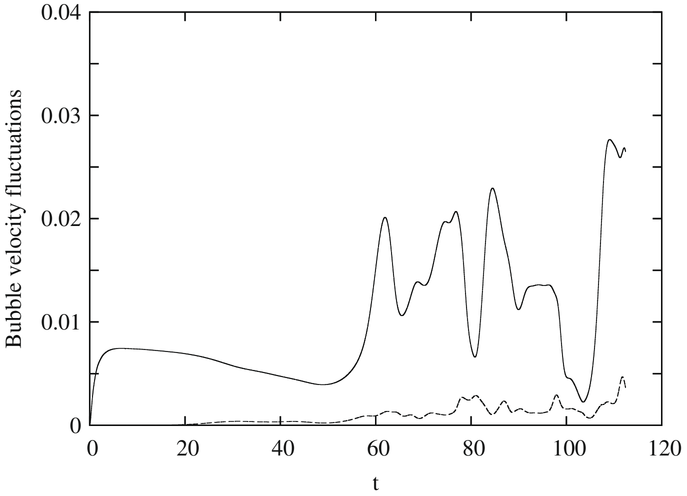


Fig. 24. Vertical and horizontal velocity fluctuations of the bubbles w.r.t. the centroid bubble velocity, respectively denoted by k'_v (*solid*) and k'_h (*dashed*).

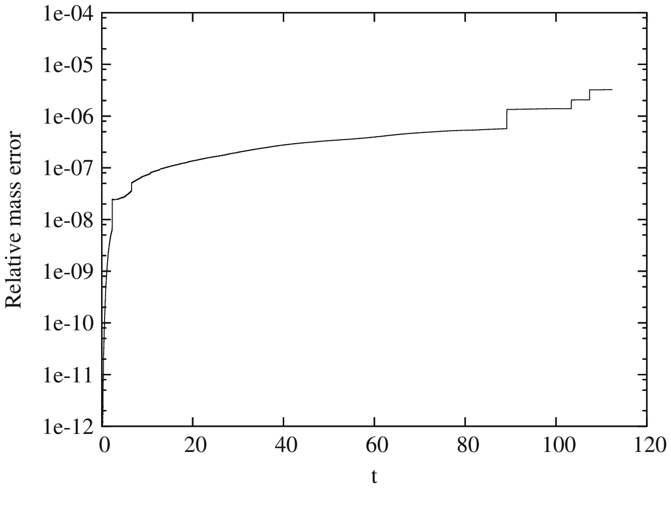


Fig. 25. Average relative mass error of the bubbles (free array with eight bubbles).

Fig. 24 indicates that the amplitudes of the bubble velocity fluctuations are still growing, suggesting that the instability of the system continues to grow (i.e. its dynamics have not yet fully developed as suggested previously). Nevertheless, the results of Fig. 24 compare well with the results by [6], showing a similar magnitude of the horizontal and vertical bubble fluctuations. Finally, the results of the average relative mass errors of the bubbles is presented in Fig. 25. The final number of 3.3×10^{-6}, i.e. $3.3 \times 10^{-4}\%$ loss of the initial mass, shows excellent conservation of bubble mass over an extensive simulation time and a large number of timesteps, as the current computation is performed for more than 200,000 timesteps.

7. Conclusion

The multiple marker front-capturing method is introduced as a novel approach for the numerical simulation of colliding bodies of the same fluid without numerical coalescence in the framework of the Level-Set/Volume-Of-Fluid methodology. The method is validated by its application to the case of gravity-driven drop impact on a liquid–liquid interface. The results of the drop impact study imply that the multiple marker method can be used effectively for the simulation of droplets in a liquid, accurately representing the effect of surface tension on the dynamics of the flow. While the multiple marker approach successfully prevents numerical coalescence of interfaces through the representation of separate interfaces with different marker functions, it is acknowledged that situations may occur where insufficient numerical resolution is available to represent fluid motion in the thin film between the interfaces. This implies that application of the multiple marker front-capturing method for the simulation of liquid–liquid dispersions is appropriate when one of the following conditions is satisfied:

(1) Combination of two fluid phases with a viscosity ratio equal or close to unity.
(2) Low volume fraction of the dispersed phase, resulting in a relatively low collision frequency of drops.

For dense dispersions of fluids with small viscosity ratio ($\lambda \ll 1$), additional (local) grid refinement appears to be indispensable to accurately represent all relevant scales of the flow.

The results presented in section six show that the multiple marker front-capturing method can be used to study dispersions with multiple (eight) droplets.

Obviously the computational effort of the simulation increases with the number of droplets, each one being represented by separate marker functions. Currently, the LS and VOF function of each droplet is defined in the entire three-dimensional domain. The required memory storage and computational effort which is associated with interface representation and advection scales with N_dN_h, N_d being the number of droplets and N_h the number of computational cells. Consequently, the simulation of dispersions with a substantial amount of droplets, i.e. $O(100)$ or more, is not practical with the present implementation. The issue of computational expense can be resolved by localization of the marker functions in the vicinity of the interface. Using the distance function property of the LS function, cells in the vicinity of each droplet can be identified to store and update the VOF and LS functions only within a layer of a few cells on either side of the droplet interface. As a result, the memory storage and computational time for representation and advection of the interface can be substantially reduced and its scaling limited to $N_dN_h^{2/3}$.

Acknowledgments

The present work was supported financially by the Netherlands Organisation for Scientific Research (NWO). Simulations have been performed using the facilities of SARA Computing and Networking Services in the Netherlands. Computational resources were sponsored by the National Computing Facilities Foundation (NCF). Zulfaa Mohamed-Kassim and Ellen Longmire are gratefully acknowledged for providing the experimental data of the drop impact study. Mark Sussman is gratefully acknowledged for providing his algorithm for the Level-Set reinitialization procedure.

References

[1] E. Aulisa, S. Manservisi, R. Scardovelli, A mixed markers and Volume-of-Fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows, J. Comp. Phys. 188 (2003) 611–639.

[2] Y.C. Chang, T.Y. Hou, B. Merriman, S. Osher, A Level-Set formulation of Eulerian interface capturing methods for incompressible fluid flows, J. Comp. Phys. 124 (1996) 449–464.

[3] B.K. Chi, L.G. Leal, A theoretical study of the motion of a viscous drop toward a fluid interface at low Reynolds number, J. Fluid Mech. 201 (1989) 123–146.

[4] R. Delfos, U. Miessner, E. Coyajee, R. Lindken, J. Westerweel, An oil droplet colliding with an oil–water interface: an experimental flow facility for the validation of numerical simulation data, in: Euromech Colloquium 479: Numerical Simulation of Multiphase Flow with Deformable Interfaces, Scheveningen, The Netherlands, 14–16 August 2006. <www.ahd.tudelft.nl/euromech/>.

[5] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle Level-Set method for improved interface capturing, J. Comp. Phys. 183 (2002) 83–116.

[6] A. Esmaeeli, G. Tryggvason, Direct numerical simulations of bubbly flows. Part 2. Moderate Reynolds number arrays, J. Fluid Mech. 385 (1999) 325–358.

[7] M.M. Francois, S.J. Cummins, D.D. Dendy, D.B. Kothe, J.M. Sicilian, M.W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, J. Comp. Phys. 213 (2006) 141–173.

[8] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.

[9] J. Helmsen, P. Colella, E.G. Puckett. Non-Convex Profile Evolution in Two-Dimensions Using Volume-of-Fluids, Technical Report LBNL-40693, Lawrence Berkeley National Laboratory, 1997.

[10] M. Kang, R.P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.

[11] K.M.T. Kleefsman, G. Fekken, A.E.P. Veldman, B. Iwanowski, B. Bunchner, A Volume-of-Fluid based simulation method for wave impact problems, J. Comp. Phys. 206 (2005) 363–393.

[12] X.-D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comp. Phys. 160 (2000) 151–178.

[13] J. Lopez, J. Hernandez, P. Gomez, F. Faura, An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows, J. Comp. Phys. 208 (2005) 51–74.

[14] D. Lorstadt, M. Francois, W. Shyy, L. Fuchs, Assessment of Volume-of-Fluid and immersed boundary methods for droplet computations, Int. J. Numer. Meth. Fluids 46 (2004) 109–125.

[15] M. Manga, H.A. Stone, Low Reynolds number motion of bubbles, drops and rigid spheres through fluid–fluid interfaces, J. Fluid Mech. 287 (1995) 279–298.

[16] Z. Mohamed-Kassim, E.K. Longmire, Drop impact on a liquid–liquid interface, Phys. Fluids 15 (2003) 3263–3273.

[17] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comp. Phys. 141 (1998) 112–152.

[18] M. Rudman, A volume-tracking method for incompressible multifluid flows with large density variations, Int. J. Numer. Meth. Fluids 28 (1998) 357–378.

[19] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.

[20] P.J. Shopov, P.D. Minev, The unsteady motion of a bubble or drop towards a liquid–liquid interface, J. Fluid Mech. 235 (1992) 123–141.

[21] M. Sussman, A second-order coupled Level-Set and Volume-of-Fluid method for computing growth and collapse of vapor bubbles, J. Comp. Phys. 187 (2003) 110–136.

[22] M. Sussman, E.G. Puckett, A coupled Level-Set and Volume-of-Fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comp. Phys. 162 (2000) 301–337.

[23] M. Sussman, P. Smereka, S. Osher, A Level-Set approach for computing solutions to incompressible two-phase flow, J. Comp. Phys. 114 (1994) 146–159.

[24] S.H. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible multi-fluid flows, J. Comp. Phys 100 (1992) 25–37.

[25] S.P. Van der Pijl, A. Segal, C. Vuik, P. Wesseling, A mass-conserving Level-Set method for modelling of multi-phase flows, Int. J. Numer. Meth. Fluids 47 (2005) 339–361.

[26] J.J.I.M. Van Kan, A second-order accurate pressure correction method for viscous incompressible flow, SIAM J. Sci. Stat. Comp. 7 (1986) 870–891.

[27] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithm for fluids, J. Comp. Phys. 24 (1979) 671–691.

[28] X. Zheng, J. Lowengrub, A. Anderson, V. Cristini, Adaptive unstructured volume remeshing – II: application to two- and three-dimensional Level-Set simulations of multiphase flow, J. Comp. Phys. 208 (2005) 626–650.