Eur. J. Mech. B/Fluids 21 (2002) 61–73

A novel technique for including surface tension in PLIC-VOF methods

Markus Meier [a], George Yadigaroglu [a,*], Brian L. Smith [b]

[a] *Swiss Federal Institute of Technology, Nuclear Engineering Laboratory, ETH-Zentrum, CLT, CH-8092 Zurich, Switzerland*
[b] *Paul Scherrer Institute, Laboratory for Thermal Hydraulics, CH-5232 Villigen PSI, Switzerland*

Received 17 November 2000; received in revised form 31 July 2001; accepted 1 August 2001

Abstract

Various versions of Volume-of-Fluid (VOF) methods have been used successfully for the numerical simulation of gas-liquid flows with an explicit tracking of the phase interface. Of these, Piecewise-Linear Interface Construction (PLIC-VOF) appears as a fairly accurate, although somewhat more involved variant. Including effects due to surface tension remains a problem, however. The most prominent methods, Continuum Surface Force (CSF) of Brackbill et al. and the method of Zaleski and co-workers (both referenced later), both induce spurious or 'parasitic' currents, and only moderate accuracy in regards to determining the curvature. We present here a new method to determine curvature accurately using an estimator function, which is tuned with a least-squares-fit against reference data. Furthermore, we show how spurious currents may be drastically reduced using the reconstructed interfaces from the PLIC-VOF method. © 2002 Éditions scientifiques et médicales Elsevier SAS. All rights reserved.

Keywords: Surface tension modelling; Piecewise-Linear Interface Construction; PLIC; Volume-of-Fluid method; VOF; Spurious currents; Parasitic currents; Continuum Surface Force; CSF; Method of Zaleski and co-workers

1. Introduction

Two-phase flow of a liquid and a gas, separated by a moving, deforming interface, is of interest for many different applications, ranging from the environmental sciences to the petroleum and nuclear industries. Many numerical methods have been developed which can simulate particular cases of such flows. In the mesoscopic range, these methods explicitly resolve and track the phase interface with special numerical techniques of either Lagrangian (moving-grid) or Eulerian (fixed-grid) type. While Lagrangian techniques are superior for small deformations of the interfaces, Eulerian techniques are usually preferred for highly distorted, complex interfaces.

The Volume-of-Fluid (VOF) method is probably the most widely used Eulerian technique for interface tracking. An extra field variable – usually called C or F – is defined on the (fixed) grid to indicate the liquid-volume fraction in each computational cell. Approximate interface locations are 'reconstructed' from C, and special, non-diffusive, often geometry-based, techniques are used to advect C in a way that keeps the phase interface sharp. Earlier versions use rather simple, crude reconstructions but, more recently, Piecewise-Linear Interface Construction (PLIC-VOF) has been introduced [1–4], which approximates the interface by a straight line within each computational cell. Fig. 4 shows an example of such an interface approximation. While somewhat more involved than earlier versions in terms of geometric and topological information to be handled, PLIC-VOF is

[*] Correspondence and reprints.
E-mail addresses: meier@iet.mavt.ethz.ch (M. Meier), yadigaroglu@iet.mavt.ethz.ch (G. Yadigaroglu), Brian.Smith@psi.ch (B.L. Smith).

PII: S0997-7546(01)01161-X

more accurate and less diffusive, and can conserve mass exactly. A version of this method was used for the study of gas injection into water in our work [5].

For a number of flow problems, such as bubble or droplet flows, surface tension is the dominant force at the interface, and the interfacial momentum jump conditions can be reduced to the Young–Laplace equation (see for example Carey [6] for a derivation and discussion):

$$p_{l,i} - p_{g,i} = \sigma\kappa \tag{1}$$

with surface tension σ, local interface curvature κ, and liquid- and gas-side pressures at the interface $p_{l,i}$ and $p_{g,i}$. This relation may be formally included in the Navier–Stokes equation as follows [7,8,4]:

$$\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\nabla^2\mathbf{v} + \mathbf{g} + \frac{(\sigma\kappa\delta_s\mathbf{n})}{\rho} \tag{2}$$

with a Dirac-pulse δ_s of units [m^{-1}], an interface-normal unit vector $\mathbf{n}$, and notation commonly used in fluid mechanics [9] otherwise. This simplified equation is widely used, and leads to reasonable results for surface-tension driven flows. For a detailed derivation, see for example Kataoka [10].

Implementing (1) in a numerical scheme involves a twofold task: the curvature κ needs to be determined, and the resulting pressure jump $\Delta p = p_{l,i} - p_{g,i}$ must be applied appropriately to the fluids. Essentially two methods to accomplish this have been published in detail. The Continuum Surface Force (CSF) method was the first introduced [11], has since been used by various researchers [12–15], and is included in commercial codes such as CFX-4 [16] and Star-CD [17]. CSF converts the interfacial surface force into a corresponding volume force, which can then be included in the discretized Navier–Stokes equation. The curvature is determined from:

$$\kappa = -\nabla\cdot\mathbf{n} \tag{3}$$

which is the derivative of $\mathbf{n}$ along the surface. The unit interface-normal vector $\mathbf{n}$ is determined from the gradient of a smoothed phase field $\tilde{C}$, obtained by convolution of the liquid-volume-fraction field C with a B-spline kernel; the transition from one phase to the other takes place continuously over several cells. For inclusion into (2), CSF uses:

$$(\sigma\kappa\delta_s\mathbf{n}) = \sigma\kappa(\mathbf{x})\frac{\nabla\rho(\mathbf{x})}{(\rho_l - \rho_g)}\frac{2\rho(\mathbf{x})}{(\rho_l + \rho_g)}. \tag{4}$$

For $\rho(\mathbf{x})$, either a smoothed density function or, directly, the unsmoothed density field from the flow solution, may be used. The first fraction in (4) represents the vector $\mathbf{n}$ itself, and the second fraction corrects the force depending on the local density. This combination should ensure approximately equal acceleration in regions of either high or low density, and this prevents distortion of the interfacial region. All terms are approximated by finite differences.

An elegant alternative method has been presented by Zaleski and co-workers [7,8,4] in which the surface tension force is described according to:

$$(\sigma\kappa\delta_s\mathbf{n}) = -\nabla\cdot\mathbf{T}, \tag{5}$$

where the tensor is given as $\mathbf{T} = -\sigma(\mathbf{I} - \mathbf{n}\otimes\mathbf{n})$, with identity matrix $\mathbf{I}$ and tensor product $\otimes$. The unit normal $\mathbf{n}$ is defined in terms of a smoothed phase field $\tilde{C}$, similar to CSF, giving:

$$(\sigma\kappa\delta_s\mathbf{n}) = \sigma\nabla\cdot\left(|\nabla\tilde{C}|\mathbf{I} - \frac{\nabla\tilde{C}\otimes\nabla\tilde{C}}{|\nabla\tilde{C}|}\right) \tag{6}$$

which the authors approximated using finite differences. This formulation is conservative, meaning that no false net forces act on closed surfaces. It should be noted that the curvature κ is not explicitly computed.

Both CSF and the method of Zaleski and co-workers are, in principle, simple and robust, and involve only the solving of a field equation for $\tilde{C}$. However, both methods are known to induce strong, spurious currents near the interface. Alternative methods have only appeared recently and are not discussed here (for example [18–20]). To resolve the difficulties, we have developed a new method to determine the curvature κ (Section 2), and to include surface tension in a way that keeps spurious currents low (Section 3).

2. Curvature estimation

If the mesh is adequately refined, the liquid-volume fractions $C_{i,j}$ in cell (i,j) and its neighbours may be used to obtain the main orientation of the interface in that cell, and PLIC-VOF methods do indeed determine the interface-normal vector $\mathbf{n}_{i,j}$ from these liquid-volume fractions [1,2,4]. In 2D, a so-called 'stencil' of nine cells is used, including cell (i,j) and its direct and diagonal neighbours; in 3D, the stencil comprises 27 cells. Formally, **n** results from the gradient of C, which may be discretized in various ways [21].

Analogously, the stencils can also be used to estimate the curvature $\kappa_{i,j}$. Formally, the curvature may be determined from (3), which involves a second derivative of C and, in practice, the resulting discretisation requires several interpolations, and yields a rather inaccurate curvature estimate [11].

We have pursued a different approach: instead of using discretized equations, for a given stencil, we look for a function f_κ which yields a best estimate of the curvature. In 2D, we have:

$$\kappa_{i,j} = f_\kappa\left(\begin{bmatrix} C_{i-1,j+1} & C_{i,j+1} & C_{i+1,j+1} \\ C_{i-1,j} & C_{i,j} & C_{i+1,j} \\ C_{i-1,j-1} & C_{i,j-1} & C_{i+1,j-1} \end{bmatrix}\right). \tag{7}$$

We use a polynomial approximation for f_κ, the coefficients of which are determined only once, in advance, from a least-squares fit against reference data. In the following, we summarize the method for 2D; details are presented in [5].

2.1. Variable reduction

A function of nine variables (or 27 in 3D) for f_κ is not practical. Instead, we define, for each cell (i,j), only three parameters Σ, Ξ and Ω, which should contain the curvature information:

$$\Sigma_{i,j} = \left\{\sum_{i'=i-1}^{i+1}\sum_{j'=j-1}^{j+1} C_{i',j'}\right\} - A'_{i,j}, \tag{8}$$

$$\Xi_{i,j} = C_{i,j}, \tag{9}$$

$$\Omega_{i,j} = \arctan\left\{\frac{\min(|n_x|,|n_y|)}{\max(|n_x|,|n_y|)}\right\}. \tag{10}$$

We have constructed these parameters empirically in such a way that they should be strongly correlated with the curvature. The first, Σ, accounts for the liquid content in the stencil, compared to the case of a flat interface, as shown in Fig. 1, left. The area $A'_{i,j}$ can be computed easily from $C_{i,j}$ and **n** using planar geometry.

The second parameter, Ξ, is simply the liquid-fraction in the cell itself, and Ω is the tilt angle of the interface-normal vector **n** to the nearest horizontal or vertical. Fig. 2 shows situations constructed so that only one of the three parameters varies, and how this parameter influences κ. We postulate that Σ, Ξ, Ω contain all the essential information concerning the local curvature, and that, for any smooth interface with a radius of curvature greater than the mesh-width Δx, it is possible to estimate the curvature $\kappa_{i,j}$ solely from these numbers with a unique function:

$$\kappa = f_\kappa(\Sigma, \Xi, \Omega). \tag{11}$$

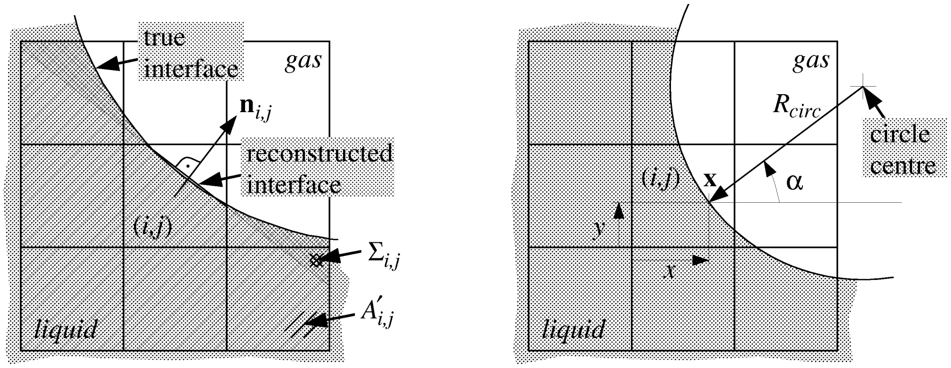

Fig. 1. Left: the definition of $\Sigma_{i,j}$: for a cell (i,j) containing an interface, the interface-normal vector $\mathbf{n}_{i,j}$, and a straight line perpendicular to $\mathbf{n}_{i,j}$ approximating the interface a, are determined first. If the interface were flat, the liquid in the stencil would cover the hatched area $A'_{i,j}$. Due to the curvature, the effective amount of liquid ($\sum C_{i,j}$) differs from $A'_{i,j}$; Σ is the difference (the cross-hatched area). Right: a circle as a calibration interface, defined by x, y, α and $\kappa_{\text{true}} = 1/R_{\text{circ}}$.

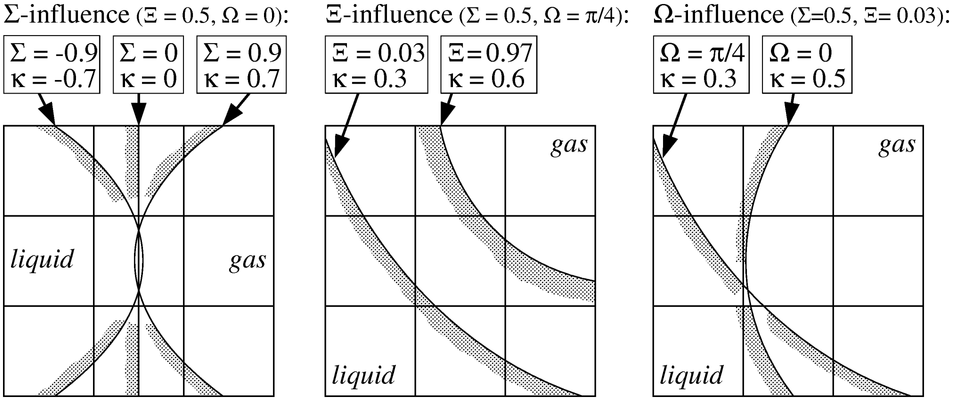

Fig. 2. Influence of the three numbers Σ, Ξ and Ω from (8), (9) and (10) on the curvature κ in the central cell. Left: Ξ and Ω are fixed, and only Σ is varied; this has a clear influence on κ. Centre, right: Ξ and Ω respectively are varied, showing equally an influence on κ.

For f_κ, we use a third-order polynomial:

$$\kappa = \frac{1}{\Delta x}\bigl(c_1 + c_2\Xi + c_3\Omega + c_4\Sigma + c_5\Xi^2 + c_6\Omega^2 + \cdots + c_{20}\Xi\Omega\Sigma\bigr) \tag{12}$$

with the twenty coefficients c_1–c_{20} determined, once only, from a least-squares fit against reference data. This function is then used in all flow simulations. Therefore, even if κ is evaluated a very large number of times, the computational expense is small.

2.2. Reference data and least-squares fit

Reference data are generated using a large set of calibration interfaces, for which the numbers Σ, Ξ, Ω and the true curvature κ_{true} are known. Circles are a natural choice as calibration interfaces, since their curvature, simply $\kappa_{\text{true}} = 1/R_{\text{circ}}$, is constant along the interface, and the C-field and the parameters Σ, Ξ, Ω are easily determined. We assume a square grid, so that the situation is geometrically identical for all cells. Fig. 1, right, gives an example of the stencil of nine cells surrounding the cell (i, j), intersected by a circular interface. By choosing a point on the circle $\mathbf{x} = (x, y)$ within cell (i, j), a tilt angle α and a radius R_{circ}, the circle is defined uniquely. The four parameters x, y, α and κ_{true} are then varied in discrete steps, defining a large set of such test cases. The tilt angle need only cover the first octant $[0, \pi/4]$; other angles lead to similar cases. We used:

$$\begin{aligned}
&\frac{x}{\Delta x} = 0.05, 0.15, \ldots, 0.95; \qquad \frac{y}{\Delta y} = 0.05, 0.15, \ldots, 0.95,\\
&\alpha = 0, 0.025\pi, \ldots, 0.25\pi,\\
&\Delta x \cdot \kappa_{\text{true}} = -1.15, -\frac{1}{2}, -\frac{1}{3}, \ldots, -\frac{1}{10}, -0.08, -0.06, \ldots, +0.08, +\frac{1}{10}, \ldots, +\frac{1}{2}, +1.15
\end{aligned} \tag{13}$$

which yields 31 900 combinations. For each of them, the pertinent C values in the stencil are computed by a simple 'pixel-counting method' [5], and then Σ, Ξ and Ω are computed from (8), (9) and (10). With standard least-squares fitting, we determine the coefficients c_1–c_{20} that minimize $\sum(\kappa_{\text{true}} - \kappa)^2$, with κ from (12). Thus, our estimator for $\kappa_{i,j}$ essentially consists of (8), (9), (10) and (12) with constant coefficients c_1–c_{20} obtained from the reference data created with (13).

We have further improved results somewhat for small curvatures by using a cascade of three estimators. If the first estimator yields a $|\kappa| < 0.4$, a second estimator optimized for this region is called, created with

$$\Delta x \cdot \kappa_{\text{true}} = -0.4, -0.32, \ldots, +0.4 \tag{14}$$

Finally, if this yields a $|\kappa| < 0.1$, a third estimator, optimized for this region, is called, which was created with:

$$\Delta x \cdot \kappa_{\text{true}} = -0.1, -0.08, \ldots, +0.1. \tag{15}$$

To avoid discontinuities at the estimator limits, the results are linearly interpolated near these limits, i.e., in the intervals [0.3−0.4] and [0.075−0.1], respectively. Table 1 lists the coefficients of these three estimators. Further improvements are certainly feasible. However, we have stopped the development here, and used this cascade of three estimators in all following tests.

Table 1
Coefficients of the three cascaded estimators

Coefficient	$f_i(\Xi, \Omega, \Sigma)$	1st estimator (κ_1)	2nd estimator ($\lvert\kappa\rvert < 0.4$)	3rd estimator ($\lvert\kappa\rvert < 0.1$)
c_1	1	$+8.809173 \cdot 10^{-3}$	$+8.744777 \cdot 10^{-3}$	$+6.411956 \cdot 10^{-3}$
c_2	Ξ	$-7.302646 \cdot 10^{-2}$	$-4.773088 \cdot 10^{-2}$	$-1.217681 \cdot 10^{-1}$
c_3	Ω	$+1.476918 \cdot 10^{-1}$	$+1.911265 \cdot 10^{-1}$	$+1.213353 \cdot 10^{-1}$
c_4	Σ	$+8.127531 \cdot 10^{-1}$	$+1.051815 \cdot 10^{0}$	$+1.091137 \cdot 10^{0}$
c_5	Ξ^2	$+1.662098 \cdot 10^{-1}$	$+9.073978 \cdot 10^{-2}$	$+3.268512 \cdot 10^{-1}$
c_6	Ω^2	$-3.158622 \cdot 10^{-1}$	$-3.656622 \cdot 10^{-1}$	$-1.658780 \cdot 10^{-1}$
c_7	Σ^2	$-7.899257 \cdot 10^{-2}$	$-1.242422 \cdot 10^{-1}$	$+9.856956 \cdot 10^{-2}$
c_8	$\Xi\Omega$	$-2.954707 \cdot 10^{-1}$	$-3.824833 \cdot 10^{-1}$	$-2.427756 \cdot 10^{-1}$
c_9	$\Xi\Sigma$	$-8.268886 \cdot 10^{-1}$	$-2.037791 \cdot 10^{-1}$	$-1.522460 \cdot 10^{-1}$
c_{10}	$\Omega\Sigma$	$+1.961225 \cdot 10^{-2}$	$-9.380094 \cdot 10^{-1}$	$-8.981913 \cdot 10^{-1}$
c_{11}	Ξ^3	$-1.107777 \cdot 10^{-1}$	$-6.043060 \cdot 10^{-2}$	$-2.178954 \cdot 10^{-1}$
c_{12}	Ω^3	$-4.160499 \cdot 10^{-4}$	$-1.001143 \cdot 10^{-3}$	$-2.737967 \cdot 10^{-4}$
c_{13}	Σ^3	$-1.507973 \cdot 10^{-2}$	$-5.859334 \cdot 10^{-2}$	$-2.349028 \cdot 10^{0}$
c_{14}	$\Xi^2\Omega$	$-1.792648 \cdot 10^{-4}$	$-3.742121 \cdot 10^{-4}$	$-2.355644 \cdot 10^{-5}$
c_{15}	$\Xi^2\Sigma$	$+8.268877 \cdot 10^{-1}$	$+2.037529 \cdot 10^{-1}$	$+1.522527 \cdot 10^{-1}$
c_{16}	$\Omega^2\Xi$	$+6.325443 \cdot 10^{-1}$	$+7.331659 \cdot 10^{-1}$	$+3.322431 \cdot 10^{-1}$
c_{17}	$\Omega^2\Sigma$	$-6.501391 \cdot 10^{-2}$	$+4.143641 \cdot 10^{-1}$	$+2.703116 \cdot 10^{-1}$
c_{18}	$\Sigma^2\Xi$	$+1.579836 \cdot 10^{-1}$	$+2.482401 \cdot 10^{-1}$	$-1.982763 \cdot 10^{-1}$
c_{19}	$\Sigma^2\Omega$	$+2.944518 \cdot 10^{-6}$	$+3.065368 \cdot 10^{-4}$	$+1.363379 \cdot 10^{-3}$
c_{20}	$\Xi\Omega\Sigma$	$+4.704595 \cdot 10^{-6}$	$+1.256257 \cdot 10^{-4}$	$-3.048371 \cdot 10^{-7}$

2.3. *Extension to three dimensions*

The method should be extendable to 3D, if a uniform grid with cubic cells is used. The stencil would then consist of 27 cells, and the estimator would be a function of four variables:

$$\kappa = f_\kappa(\Sigma, \Xi, \Omega_1, \Omega_2) \tag{16}$$

with two orientation angles Ω_1 and Ω_2, latitude and longitude, instead of Ω. Σ has an equivalent definition with volumes in 3D. Reference data could be produced using spheres rather than circles, defined by parameters x, y, z, α_1, α_2 and κ_{true}.

2.4. *Validation tests*

For a given ‘circular droplet’, the curvature is known and constant. Any algorithm for computing the curvature should, ideally, yield this unique curvature in all interfacial cells. The actual predictions would then reveal the quality of the algorithm.

We have set up C-matrices for 25 droplets with diameters $D/\Delta x = 10^E$, where $E = 0.5, 0.55, \ldots, 1.7$, which defines diameters ranging in size from about 3 to 50 cells. The centers of the droplets were placed on cell corners. In Fig. 3, results with our method are compared to CSF results. The method of Zaleski and co-workers does not yield explicit curvatures, and is therefore not included in this comparison. For CSF, we have implemented the ‘ALE-like scheme’ of [11], which is the best variant of CSF, according to [11]. To obtain the smoothed phase field $\tilde{C}$, we have convoluted the unsmoothed field C with kernel matrices of size 1×1, 3×3 and 5×5, respectively (normalized so that the sum of all elements is equal to unity):

$$[\,1\,], \qquad \frac{1}{36} \cdot \begin{bmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{bmatrix}, \qquad \frac{1}{3721} \cdot \begin{bmatrix} 4 & 30 & 54 & 30 & 4 \\ 30 & 225 & 405 & 225 & 30 \\ 54 & 405 & 729 & 405 & 54 \\ 30 & 225 & 405 & 225 & 30 \\ 4 & 30 & 54 & 30 & 4 \end{bmatrix}. \tag{17}$$

These are obtained as the product of two cubic, uniform B-splines [22] in the x- and y-directions, the non-zero portions of which cover the ranges $[-\Delta x - \Delta x]$, $[-2\Delta x - 2\Delta x]$, $[-3\Delta x - 3\Delta x]$, respectively. The first matrix obviously yields $\tilde{C} \equiv C$, which means that the unsmoothed phase field C is used.

The results in Fig. 3 show the high level of noise in the curvature estimates obtained using CSF. With our method, we achieve about 3 to 7 times better agreement of the mean curvature with the reference value (unity), and about 4 to 7 times lower standard deviation.

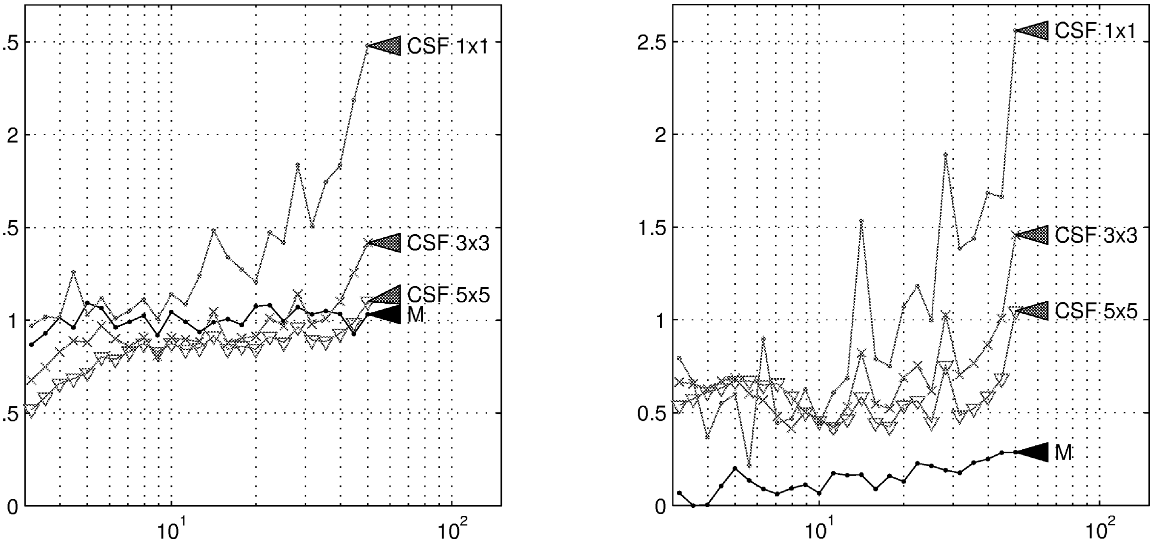

Fig. 3. Mean (left) and standard deviation (right) of the normalized curvature estimates versus normalized diameter $D/\Delta x$ obtained using our method ('M') and CSF with smoothing matrices 1×1, 3×3 and 5×5; the true curvature is 1.

Further tests could be made using curves such as sinusoids, ellipsoids, hyperbola or parabola. However, since such curves locally resemble circles (for $\Delta x \to 0$, the F-field converges against the field for an osculating circle), similar results may be expected. Indeed, further tests made with a decaying-wave interface showed equally good results.

3. Discretisation of the surface-tension term

In PLIC-VOF methods, interfaces in each cell are reconstructed with a straight line in 2D, or a plane in 3D. This enables the interface motion to be computed accurately and diffusion-free. Knowledge of the interface exact location can also be very useful in applying the surface tension force.

We consider the 2D case below; all steps are extendable to 3D, however. Fig. 4 shows an example with three cells, with a straight line in each cell approximating the true interface. Once the reconstruction is performed, the interfacial area $A_{i,j}$ in a cell follows from simple geometrical formulae. With this area known, the pressure jump $\Delta p_{i,j}$ of (1) corresponds to a force $\mathbf{K}_{i,j}$ which acts in cell (i, j):

$$\mathbf{K}_{i,j} = \sigma \kappa_{i,j} A_{i,j} \mathbf{n}_{i,j}. \tag{18}$$

For the discretized momentum equations, this surface force must be converted to the specific body force term in (2), which we abbreviate as $\mathbf{e} = (\sigma \kappa \delta_s \mathbf{n})/\rho$. The particular way in which the surface tension force is allocated in the numerical computations to the cells at the interface, is the main factor determining the magnitude of the spurious currents that may be generated. In our experience, restricting the application of the body force to the actual interfacial cells, and making use of the reconstructions, can effectively limit these spurious currents. Two successive steps are necessary. We have considered two variants for each, described in the following two sub-sections.

3.1. Cell density versus mean density

The force $\mathbf{K}_{i,j}$ must be divided by a mass to yield a specific body force $\mathbf{e}_{i,j}$ (units $\mathrm{m/s^2}$). Since the force is defined in cell (i, j), the natural choice seems to be the mass in this cell, which is simply the cell volume V_c times the density $\rho_{i,j}$. Thus:

$$\mathbf{e}_{i,j} = \frac{\mathbf{K}_{i,j}}{\rho_{i,j} V_c} = \frac{\sigma}{\rho_{i,j}} \kappa_{i,j} \frac{A_{i,j}}{V_c} \mathbf{n}_{i,j}. \tag{19}$$

However, if the cell density $\rho_{i,j}$ is used, this acceleration becomes dependent on the liquid fraction $C_{i,j}$ in the cell. Especially for a small liquid fraction $C_{i,j}$ and density $\rho_{i,j}$, this acceleration becomes large and must be balanced by the terms $\partial u/\partial t$, $\partial v/\partial t$ in (2). In this way, strong spurious fluid velocities are induced.

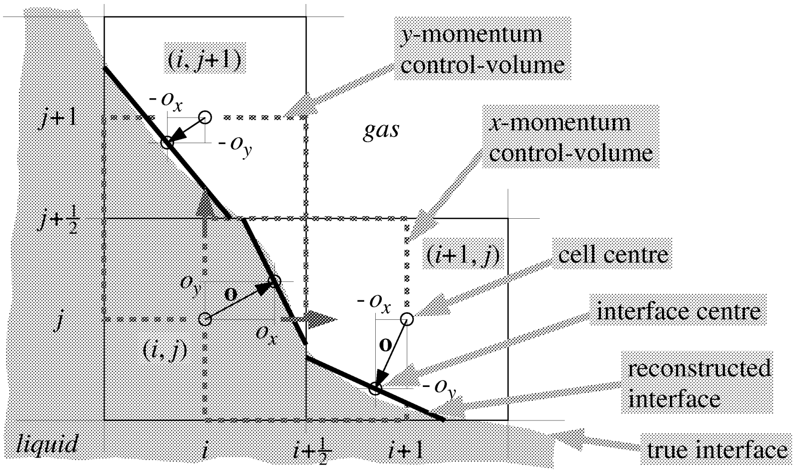

Fig. 4. Three cells with interface reconstructions by straight lines. Offset vectors **o** are used to weight the partition of the surface-tension body force of cell (i, j) on the staggered momentum control-volumes.

Instead, surface tension should rather act uniformly, regardless of the instantaneous liquid fraction and density in the cell. It may be argued that the force should act on the mass in a shifted cell, which is $\Delta x/2$ thick on both sides of the interface. In this cell, the density is constant and equal to the mean density $\bar{\rho} = (\rho_l + \rho_g)/2$. By dividing by $\bar{\rho}$ rather than $\rho_{i,j}$ in (19), spurious currents can indeed be reduced:

$$\mathbf{e}_{i,j} = \frac{\sigma}{\bar{\rho}} \kappa_{i,j} \frac{A_{i,j}}{V_c} \mathbf{n}_{i,j}. \tag{20}$$

For both variants, the integral effect is correct: in a spherical bubble or droplet, the correct pressure increase (1) is reproduced.

3.2. 50–50 partition versus weighted partition

Normally one uses a staggered-mesh layout in which (2) is solved on cell-face-centered control volumes. Therefore, the surface tension computed for a particular cell must be partitioned and distributed to the adjacent staggered momentum control-volumes. The simplest way is a '50–50 partition':

$$e_{x,i+\frac{1}{2},j} = 0.5 e_{x,i,j} + 0.5 e_{x,i+1,j}, \tag{21}$$

$$e_{y,i,j+\frac{1}{2}} = 0.5 e_{y,i,j} + 0.5 e_{y,i,j+1}. \tag{22}$$

However, the reconstructed interfaces can be used to produce a more accurate approximation. The contribution of $\mathbf{e}_{i,j}$ can be apportioned according to the offset of the interface from the cell centre. One option is to use the 'offset vector' $\mathbf{o} = (o_x, o_y)^{\mathrm{T}}$ of the interface centre from the cell centre, as shown in Fig. 4. Using o_x, o_y, for a linear partition, we have:

$$e_{x,i+\frac{1}{2},j} = (0.5 + o_{x,i,j}) e_{x,i,j} + (0.5 - o_{x,i+1,j}) e_{x,i+1,j}, \tag{23}$$

$$e_{y,i,j+\frac{1}{2}} = (0.5 + o_{y,i,j}) e_{y,i,j} + (0.5 - o_{y,i,j+1}) e_{y,i,j+1}. \tag{24}$$

With the two improvements described above, the surface tension acts only in cells where there actually is an interface and its action is properly distributed on the cell faces. Similar expressions for a weighted partition can be derived for non-square mesh layouts.

3.3. Tests with circular droplets

A circular, motionless droplet at zero gravity is a very useful test case for surface tension algorithms. In reality, this droplet should remain motionless forever. In a numerical simulation without surface tension, (2) should yield zero accelerations $\mathrm{D}\mathbf{v}/\mathrm{D}t$, and thus correctly keep the droplet motionless. However, if surface tension is included and artificial motions are generated, these can be ascribed directly and exclusively to the surface tension algorithm. Their magnitude reveals the efficacy of the treatment, and the pressure increase inside the droplet, compared to the analytical value of (1), reflects the accuracy of the algorithm.

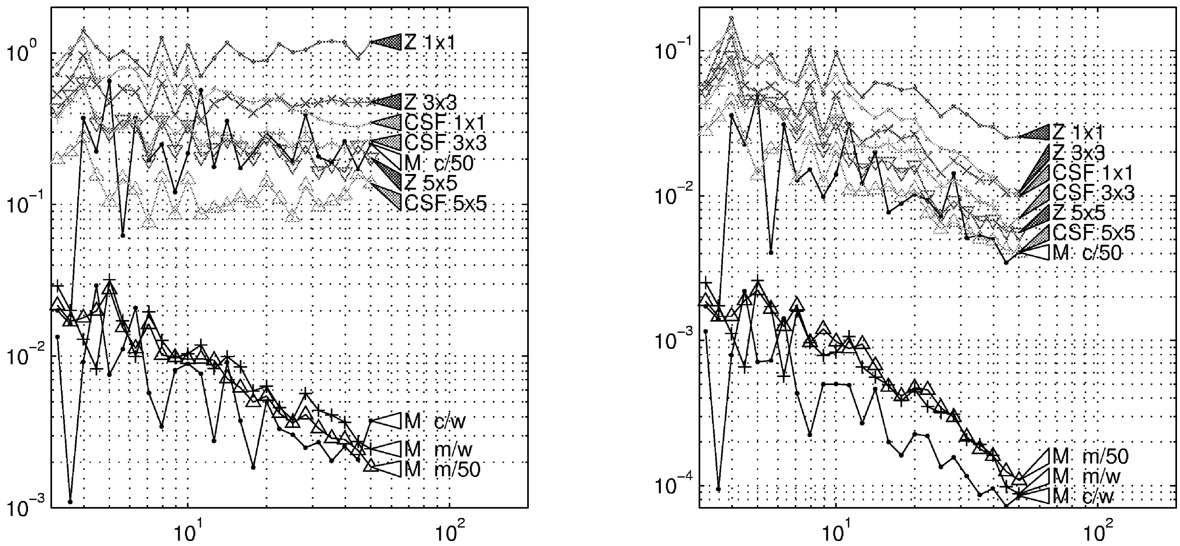

Fig. 5. After 1ms simulation time: maximum velocity (left) and mean velocity (right) in the computed domain, in [m/s], versus diameter $D/\Delta x$. Using our method ('M'; 'c': cell density, 'm': mean density; '50': 50–50 partition, 'w': weighted partition), and CSF and Zaleski method ('Z'), with smoothing matrices 1×1, 3×3 and 5×5 of (17).

We consider water and air at 20 °C, with $\sigma = 0.07274$ N/m. We use the same 25 droplets, with diameters $D/\Delta x \approx 3 - 50$, described in Section 2.4. The mesh size is $\Delta x = 1$ mm. We use our own, MAC-type, incompressible flow solver with (first-order) upwind differencing of the advection terms and (second-order) central differencing for the diffusion terms. Reference [5] should be consulted for details. Time-steps are smaller than the stability limit for surface tension [11], and are automatically adapted to meet the CFL condition. Starting from a motionless, circular droplet, the simulation is continued for 100 ms.

For the CSF scheme, we use the 'ALE-like scheme' mentioned above (Section 2.4) to determine $\kappa_{i,j}$ at cell centers, interpolate for the vertex value $\kappa_{i+\frac{1}{2},j+\frac{1}{2}}$, and determine the body force term (4) accordingly. To include the force in the cell-face-centered momentum equations, at $i+\frac{1}{2}, j$ and $i, j+\frac{1}{2}$, we interpolate from the adjacent vertex values; interpolating directly from cell centres gave worse results. The smoothed field $\tilde{C}$ is computed with kernels from (17), and the resulting term $(\sigma\kappa\delta_s\mathbf{n})$ is divided by the formally correct density of the momentum control-volume (which we determine from the effective liquid fraction therein).

For the method of Zaleski and co-workers, the body force term follows from (6). All terms are evaluated at cell centres i, j, using central differences for $\nabla\tilde{C}$, and for the divergence $\nabla \cdot (\,)$. From the body force term at cell centres i, j obtained in this way, we interpolate to the cell faces. As with the CSF method, (17) is used to obtain $\tilde{C}$, and $(\sigma\kappa\delta_s\mathbf{n})$ is divided by the density of the momentum control-volume.

With our method, we combine the two density options, mean versus cell, as described in Section 3.1 with the two ways of partitioning, as described in Section 3.2.

Already after 1 ms, spurious currents of non-negligible magnitude can be observed with all methods. Fig. 5 shows the maximum velocity in either direction, $\max(|u_{i+\frac{1}{2},j}|, |v_{i,j+\frac{1}{2}}|)$, and the mean of $|u_{i+\frac{1}{2},j}| + |v_{i,j+\frac{1}{2}}|$ over the entire computational domain. The averages over the 25 droplets are listed in Table 2. CSF and the method of Zaleski and co-workers both induce strong currents, with maxima up to 1 m/s. Smoothing apparently reduces the currents somewhat; CSF is somewhat better than the method of Zaleski and co-workers.

With our method, results are strongly dependent on the density and partitioning variants introduced in Sections 3.1 and 3.2, respectively. Use of cell densities and 50–50 partition yields the worst results, comparable to CSF, while replacing the cell densities with mean densities already improves results by more than a factor 10. Using a weighted partition instead of 50–50 further reduces spurious velocities somewhat. A further improvement is achieved if one returns to cell densities, but in combination with the weighted partition. Use of cell densities and weighted partition is in fact the best combination; it is almost two orders of magnitude better than the worst combination. The use of cell densities and 50–50 partition is not consistent (leading to cell-dependent surface forces), whereas the other three combinations do appear to be consistent.

These results confirm the usefulness of keeping the surface tension force at the interface. This can be achieved either by using a mean density, or by a weighted partition, or by both measures. Thus, the precise interface reconstruction is also very useful for accurately implementing the surface tension force. With our best combination, we have about 100 times smaller spurious currents than with CSF and the method of Zaleski and co-workers.

To further assess the accuracy of the methods, we determine the mean pressure inside the droplet, summing over those cells with at least 98% of liquid, an approach introduced in [11]:

$$\langle p\rangle = \frac{1}{N_d}\sum_{i,j=1}^{N_d} p_{i,j}. \tag{25}$$

This pressure should agree with the analytical droplet pressure $p_{\text{drop}} = 2\sigma/R$ according to (1). Table 2 lists the averages of the error $|\langle p\rangle/p_{\text{drop}} - 1|$ over all droplet diameters. The CSF method with 3×3 smoothing yields the best accuracy, with only 3.4% error; the method of Zaleski and co-workers is somewhat less accurate. With our method, we achieve reasonable accuracy with all combinations except with cell densities and 50–50 partition. With weighted partition, we can almost attain the best accuracy of CSF.

If the simulations are continued to 100ms, the picture changes drastically. The high levels of spurious currents with all versions of the CSF and Zaleski methods lead to deformation and break-up of the droplet. Most of the droplets are strongly deformed; for many of them, the deformation $\sum_{i,j}|C_{i,j} - C_{i,j}(0)|/\sum_{i,j} C_{i,j}(0)$ exceeded 1/2, and we then terminated the computation. Fig. 6, left, shows the deformation at the end of the computation. Most results with CSF and the method of Zaleski

Table 2
Averages of maximum velocity, mean velocity, pressure error and deviation from circular, over all diameters, after 1ms simulation time. (Abbreviations as in Fig. 5)

Method	Maximum velocity	Mean velocity	$\lvert\frac{\langle p\rangle}{p_{\text{drop}}} - 1\rvert$	Deviation from circular
	[m/s]	[m/s]	[%]	[%]
CSF 1×1	0.598	0.051 2	12.6	0.078
CSF 3×3	0.309	0.027 4	3.4	0.031
CSF 5×5	0.131	0.014 8	5.0	0.029
Zaleski and co-workers 1×1	1.008	0.062 7	31.2	0.135
Zaleski and co-workers 3×3	0.513	0.036 8	7.9	0.072
Zaleski and co-workers 5×5	0.284	0.026 8	12.2	0.053
M mean / 50−50	0.010	0.000 9	10.2	0.174
M cell / 50−50	0.248	0.013 7	25.0	0.230
M mean / weighted	0.011	0.000 8	4.7	0.163
M cell / weighted	0.007	0.000 5	4.4	0.080

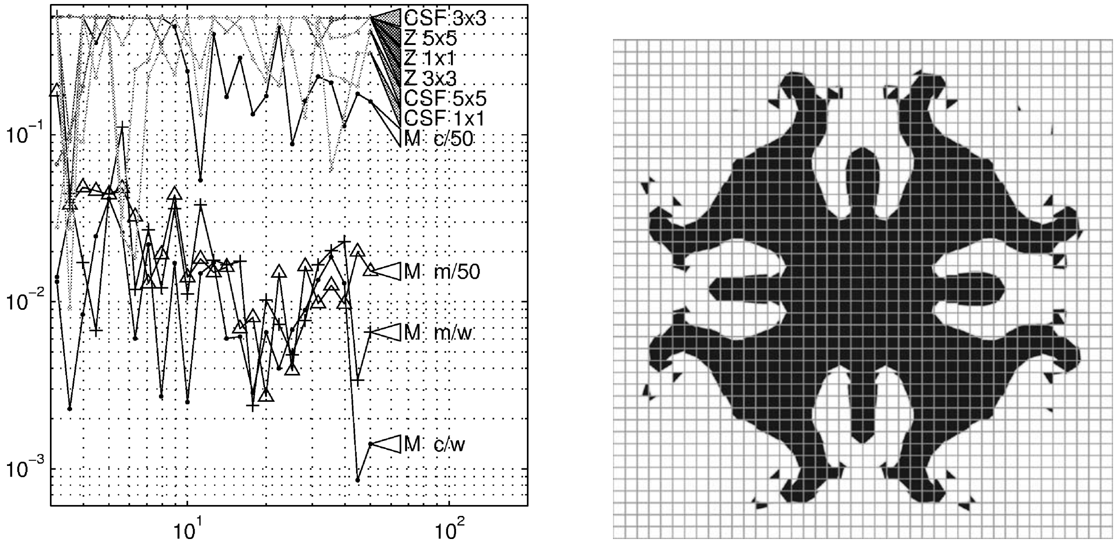

Fig. 6. Left: deviation from the original circular shape [%] at the end of the simulation (after 100 ms, or when 50% deformation is exceeded). Abbreviations as in Fig. 5. Right: typical shape of a droplet, using CSF 5 × 5.

Table 3
Averages of maximum velocity, mean velocity, pressure error and deviation from circular, respectively, over all diameters, after 100 ms simulation time

Method	Maximum velocity	Mean velocity	$\lvert \frac{\langle p \rangle}{p_{\text{drop}}} - 1 \rvert$	Deviation from circular
	[m/s]	[m/s]	[%]	[%]
M mean / 50–50	0.099	0.012 8	12.6	2.788
M mean / weighted	0.080	0.009 9	7.9	4.105
M cell / weighted	0.011	0.000 8	5.3	1.148

and co-workers are above 10%. Fig. 6, right, shows a typical example using CSF 5 × 5, where the droplet has reached 50% deformation after 57 ms.

With our method, the droplets are severely distorted only for the worst combination: cell densities and 50–50 partition. With the other three combinations, all droplets remain nearly intact. Table 3 lists the results for these three combinations. Cell densities and weighted partition are clearly the best combination, with only 1.1% deformation on the average. Furthermore, maximum and mean velocities of the spurious currents are still small, and the accuracy has suffered little. Since most results with CSF and the method of Zaleski and co-workers have completely diverged, statistics of their results are not listed in Table 3.

The spurious currents depend on the fluid properties. According to [4] and [7], numerical tests have shown maximum velocities of the order $u_{\max} \approx 0.01\sigma/\mu$ with the method of Zaleski and co-workers. Thus, reducing σ and increasing μ should help. We have performed tests which confirm this trend; however, the relative superiority of our best variants remains unchanged.

The reduction of spurious currents with our method is mainly an effect of the way in which the body force term was discretized. The improved accuracy of the curvature estimation shown in Fig. 3 has only little effect. In a specific test, we have compared results with our estimator with results obtained using the exact curvature of the circular droplet. They show that the exact knowledge of the curvature only moderately reduces spurious currents.

3.4. Dynamic tests

We have carried out a number of dynamic tests with our method, including droplet oscillation driven by surface tension, rising bubbles in the six different regimes identified by Clift et al. [23], and break-up of a liquid jet. Here, we show three examples. More examples, and details about all results may be found in [5].

An initially ellipsoidal droplet in a gravity-free environment, starting to oscillate driven by surface tension only, is a good starting point. Droplet oscillations are omnipresent in nature, and of significance in some industries, for example for ink-jet printers or combustion. Good simulation results with a Level Set method have been presented by [24].

We have performed a simulation of an oscillating droplet over a few cycles, and compared simulations on three different meshes (15 × 30, 30 × 60, 60 × 120 cells, respectively). The domain has a radius of 15 mm and a height of 30 mm. The initial drop has an ellipsoidal cross-section with radial diameter 20 mm and axial diameter 6 mm. We assume water in air at 20 °C. The Courant number range is $C = 0.1-0.25$. Fig. 7 shows a sequence of interfaces. First deviations between results with the different grids become visible in the fourth picture already, but the main shapes remain comparable for a while. The solutions inevitably diverge, due to the non-linear governing equations. Due to numerical viscosity of the flow solver, the oscillations are strongly damped. The surface-tension model, however, induces apparently realistic motions.

The rise and deformation of gas bubbles in liquid, under different conditions, is of interest especially in the chemical and nuclear industry. Clift et al. [23] present bubble velocity values according to a regime map (their Fig. 2.5) based on experimental results. Six different bubble shapes are distinguished (see below), as a function of the Eötvös, Morton and Reynolds numbers:

$$Eo = \frac{g\Delta\rho {d_e}^2}{\sigma}, \qquad M = \frac{g\mu^4\Delta\rho}{\rho^2\sigma^3}, \qquad Re = \frac{\rho d_e u_T}{\mu}, \tag{26}$$

where d_e is the diameter of a sphere of equal volume, and u_T is the terminal rise velocity. From the Clift et al. [23] figure, the Re number and thus the terminal velocity u_T can be determined. For each of the six regimes, we have chosen the point where Clift et al. have drawn a small image of the respective bubble shape. Fig. 8 shows our simulation results, obtained on an axisymmetric grid of 35 × 175 cells (the images show 20 × 160 cells). Table 4 lists the conditions for each case. Generally, we find good qualitative agreement with the shapes in the regime map, and good agreement of certain quantitative measures (aspect ratio for case 3, wake angle in case 6) with the experimental data from [23]. A more detailed discussion is contained in [5].

To conclude, we consider the break-up of a cylindrical jet of water in air, as may be observed at a water tap, for example. The wavelength of the fastest-growing disturbances and the time of break-up may be compared with theory. Carey [6] presents the results of the linear stability analysis of Rayleigh (1878). A sinusoidal perturbation on a non-moving, inviscid cylinder of

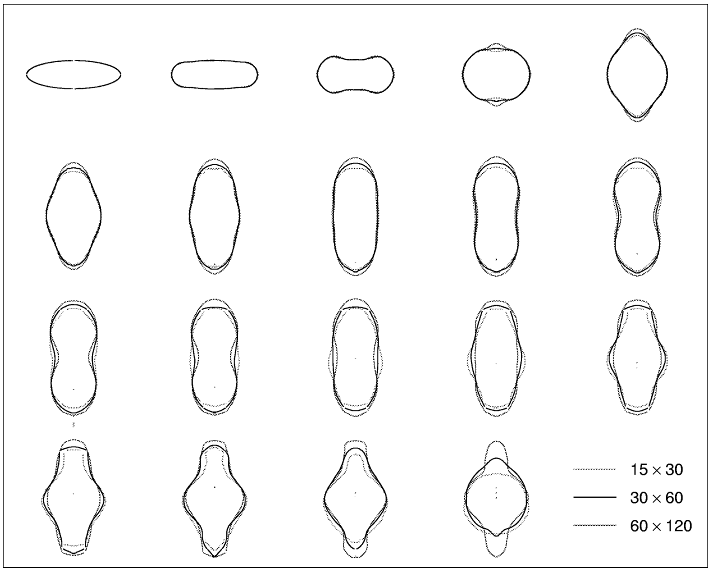

Fig. 7. Oscillation of a water droplet in air. Reconstructed interfaces with an image time-step of 0.01 s, with three different meshes. The change in area is due to the cylindrical coordinates.

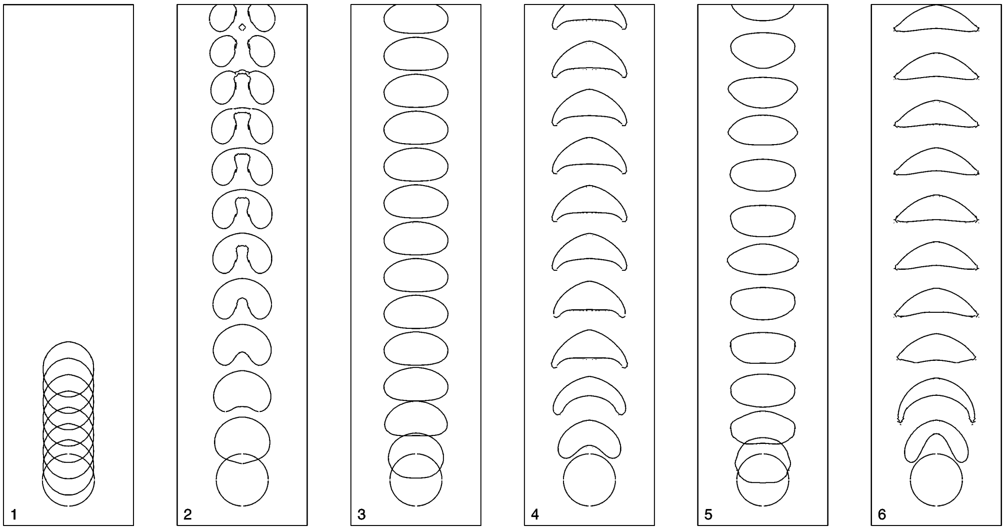

Fig. 8. Simulation results for the six cases, according to Table 4.

Table 4
Six different bubble regimes, from Clift et al. [23], their Fig. 2.5

#	d_e (m)	image-Δt (s)	Eo	M	Re	Shape (terminology of [23]):
1:	0.0015	0.01	0.35	$10^{-3.9}$	1	‘spherical’
2:	0.032	0.1	160	$10^{3.5}$	2	‘dimpled ellipsoidal-cap’
3:	0.008	0.025	10	$10^{-2.1}$	12	‘ellipsoidal’
4:	0.062	0.1	600	$10^{2.2}$	23	‘skirted’
5:	0.004	0.012	2.5	$10^{-11.8}$	2000	‘wobbling’
6:	0.045	0.08	320	$10^{-7.6}$	4200	‘spherical-cap’

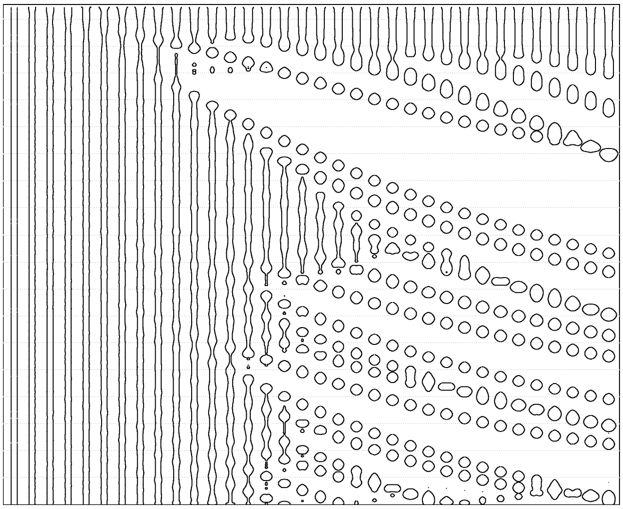

Fig. 9. Sequence of reconstructed interfaces for the jet break-up simulation. Image time-step 0.003 s, from $t = 0$ to 0.099 s. Box height 200.1 mm. The horizontal lines are $\lambda_{\max} = 10.8$ mm apart, the fastest-growing wave-length according to [6].

liquid of radius R_0 is analysed, where the Young–Laplace equation (1) holds. The fastest-growing disturbance has a wavelength $\lambda_{\max} = 9R_0$ and a frequency $\beta_{\max} = 0.34\sqrt{\sigma/\rho_l R_0{}^3}$. It takes a time $t_B = 1/\beta_{\max} \ln(R_0/\delta_0)$ for a disturbance of size δ_0 to grow to size R_0.

We consider a jet of water at 20°C, with $R_0 = 1.2$ mm and $u = 1$m/s. For this case, $\lambda_{\max} = 10.8$ mm. Since δ_0 is unknown, one can only estimate t_B. For the $\delta_0/R_0 = 10^{-3}$–10^{-4} suggested by Carey, $t_B = 0.10 - 0.13$ s.

We have carried out a simulation, assuming an initial jet with a random disturbance $\delta_0/R_0 \in [-3 \cdot 10^{-4}, 3 \cdot 10^{-4}]$, using 16×667 cells and $\Delta r = 0.3$ mm. The initial jet has a radius of four cells, plus or minus the disturbances. Liquid is continuously injected with $u = 1$ m/s through a nozzle of diameter $2R_0 = 2.4$ mm. Gravity and viscosity are neglected. The Courant number range is $C = 0.06{-}0.15$. Fig. 9 shows a sequence of interfaces from the start of the calculation. The growth of instabilities on the interface, which lead to break-up, is well visible. A spectral analysis [5] showed a clearly dominating wavelength about 15% below $\lambda_{\max}$ according to theory [6]. The simulation yielded a too short time to break-up, however; this appeared to be an effect of the random, numerical noise of our surface tension model.

4. Conclusions

We have developed a novel technique to determine interface curvature in VOF methods that uses an estimator function with coefficients fitted to reference data only once. With this method, we can estimate the curvature about 3 to 7 times more accurately than with CSF of [11]. The method is restricted to 2D uniform grids in its present form.

For the discrete approximation of the surface tension term in the momentum equation, we determine the surface force in each cell, and apply it in that cell only, rather than over several cells, as in the CSF and Zaleski methods. The surface area, which is required to calculate the force, is provided by the PLIC-VOF interface reconstruction. Results may be improved by dividing the surface tension term by the mean density instead of the (formally correct) cell density, and by using a partition on the momentum control-volumes, weighted according to the location of the reconstructed interfaces, instead of a 50–50 partition. We have tested and compared all four combinations with CSF and the method of Zaleski and co-workers: With the best variant – cell density together with weighted partition – we have up to 100 times less spurious currents with our method, compared with CSF and the method of Zaleski and co-workers.

Acknowledgments

We are indebted to Professor S. Zaleski of Laboratoire de Modélisation en Mécanique (LMM), Université Pierre et Marie Curie (Paris 6), Paris, France, for many helpful discussions.

References

[1] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, in: Numerical Methods for Fluid Dynamics: Proceedings of a First Conference, Reading, UK, Mar. 1982, pp. 273–285.

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

[3] J. Li, Calcul d'interface affine par morceaux (Piecewise linear interface calculation), C. R. Acad. Sci. II B 320 (1995) 391–396.

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

[5] M. Meier, Numerical and experimental study of large steam-air bubbles injected in a water pool, Dissertation no. 13091, Swiss Federal Institute of Technology, Zurich, Switzerland, 142 pages, 1999, http://www.lkt.iet.ethz.ch/~meier/.

[6] V.P. Carey, Liquid–Vapor Phase-Change Phenomena: An Introduction to the Thermophysics of Vaporization and Condensation Processes in Heat Transfer Equipment, Hemisphere Publishing Corporation, 1992.

[7] B. Lafaurie, C. Nardone, R. Sardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, J. Comput. Phys. 113 (1994) 134–147.

[8] S. Zaleski, J. Li, S. Succi, R. Scardovelli, G. Zanetti, Direct numerical simulation of flows with interfaces, in: Proceedings of the 2nd International Conference on Multiphase Flow, Kyoto, Apr. 3–7, Vol. 2, 1995, pp. PT2-1–PT2-12.

[9] R.B. Bird, W.E. Stewart, E.N. Lightfoot, Transport Phenomena, Wiley, New York, 1960.

[10] I. Kataoka, Local instant formulation of two-phase flow, Int. J. Multiphase Flow 12 (1986) 745–758.

[11] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modelling surface tension, J. Comput. Phys. 100 (1992) 335–354.

[12] D.B. Kothe, R.C. Mjolsness, RIPPLE: a new model for incompressible flows with free surfaces, AIAA J. 30 (11) (1992) 2694–2700.

[13] G.P. Sasmal, J.I. Hochstein, Marangoni convection with a curved and deforming free surface in a cavity, in: Gas-liquid Flows 1993: Fluids Engineering Conference, Washington, DC, June 20–24, 1993, pp. 199–207.

[14] J.R. Richards, A.N. Beris, A.M. Lenhoff, Drop formation in liquid–liquid systems before and after jetting, Phys. Fluids 7 (1995) 2617–2630.

[15] J. Wu, S.-T. Yu, B.-N. Jiang, Simulation of two-fluid flows by the least-squares finite element method using a continuum surface tension model, Int. J. Num. Meth. Eng. 42 (1998) 583–600.

[16] AEA Technology, Engineering Software, CFX Computational Fluid Engineering, http://www.aeat.co.uk/cfx/.

[17] Computational Dynamics Ltd., CFD software products, Star-CD, http://www.cd.co.uk/.

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

[19] M.W. Williams, D.B. Kothe, E.G. Puckett, Convergence and accuracy of kernel-based continuum surface tension models, in: W. Shyy (Ed.), Fluid Dynamics at Interfaces, Cambridge, MA, 1999, pp. 347–356.

[20] M. Bussmann, J. Jostaghimi, S. Chandra, On a three-dimensional volume tracking model of droplet impact, Phys. Fluids 11 (1999) 1406–1417.

[21] D.B. Kothe, W.J. Rider, S.J. Mosso, J.S. Brock, J.I. Hochstein, Volume tracking of interfaces having surface tension in two and three dimensions, AIAA Paper 96-0859, Presented at the 34th Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 15–18, 1996 (LANL Report LA-UR-96-88, http://www.lanl.gov/home/Telluride), 1–24.

[22] C. De Boor, A Practical Guide to Splines, Springer, New York, 1978.

[23] R. Clift, J.R. Grace, M.E. Weber, Bubbles, Drops and Particles, Academic Press, New York, 1978.

[24] M. Sussman, P. Smereka, Axisymmetric free boundary problems, J. Fluid Mech. 341 (1997) 269–294.