ELSEVIER

A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics

Emilie Marchandise [a,c,*], Philippe Geuzaine [d], Nicolas Chevaugeon [a],
Jean-François Remacle [a,b]

[a] *Université catholique de Louvain, Department of Civil Engineering, Place du Levant 1, 1348 Louvain-la-Neuve, Belgium*
[b] *Center for Systems Engineering and Applied Mechanics (CESAME), Université catholique de Louvain, 1348 Louvain-la-Neuve, Belgium*
[c] *Fonds National de la Recherche Scientifique (FNRS), Rue d'Egmont 5, 1000 Bruxelles, Belgium*
[d] *CENAERO, Multiphysics Group, Avenue Jean Mermoz 30, 6041 Gosselies, Belgium*

Abstract

A novel numerical method for solving three-dimensional two phase flow problems is presented. This method combines a quadrature free discontinuous Galerkin method for the level set equation with a pressure stabilized finite element method for the Navier Stokes equations.

The main challenge in the computation of such flows is the accurate evaluation of surface tension forces. This involves the computation of the curvature of the fluid interface. In the context of the discontinuous Galerkin method, we show that the use of a curvature computed by means of a direct derivation of the level set function leads to inaccurate and oscillatory results. A more robust, second-order, least squares computation of the curvature that filters out the high frequencies and produces converged results is presented.

This whole numerical technology allows to simulate a wide range of flow regimes with large density ratios, to accurately capture the shape of the deforming interface of the bubble and to maintain good mass conservation.
© 2007 Elsevier Inc. All rights reserved.

Keywords: Bubble dynamics; Surface tension; Curvature; Multiphase flow; Level set; Incompressible Navier–Stokes; Discontinuous Galerkin method

1. Introduction

This paper is in the continuation of two other papers [1,2]. The first one [1] describes the high order Runge–Kutta discontinuous Galerkin method to solve the level set equation. The second one [2] combines this level

set method with a stabilized finite element method for the Navier Stokes equations in order to compute two phase flows. In those articles, we show that, contrary to a popular belief, the poor mass conservation obtained with the level set method is more inherent to the discretization method and to the need for reinitialization than to the level set method itself. With the discontinuous Galerkin discretization of the level set equation, we show that we are able to obtain excellent conservation properties. Furthermore, contrary to the common practice of reinitializing the level set at each time step, we only need to reinitialize the level set a limited number of times.

The present paper's concern is the accurate representation of surface tension effects that have to be taken into account for computing bubble dynamics.

Bubble dynamics is frequently encountered in a variety of industrial processes. An accurate modeling of bubbly flows is a challenging task due to (i) the fact that the interface separating the two fluids needs to be tracked accurately without introducing excessive numerical smearing, (ii) the existence of a discontinuity in material properties (density and viscosity) (density ratio is about 1–1000 for air bubbles in water), (iii) the need to take into account a surface tension, that introduces a jump in the pressure field.

There are essentially two techniques for approximating the interface: front tracking and front capturing methods. The main difference between the methods is that interface tracking methods are Lagrangian, i.e. the mesh explicitly represents the interface, while interface capturing methods are Eulerian, i.e. the interface is an implicit function defined on a fixed mesh.

Examples of *interface tracking* techniques are the arbitrary Eulerian Lagrangian (ALE) method [3,4], the deformable-spatial-domain/stabilized space–time deformation (DSD/SST) [5], and methods that make use of markers (for instance triangles) connected to a set of points to track the interface [6]. Those methods are very accurate and very efficient for flexible moving boundaries with small deformations. However, they are difficult to use when the topology of the interface changes (re-connections or disconnections). In addition, significant re-meshing is needed when large deformations of the interface occur.

In *interface capturing* methods, an auxiliary function is needed to represent the interface. Those methods are very robust and have a wide range of applicability. However, they usually require higher mesh resolution. Examples of interface capturing methods are volume of fluid methods (VOF) [7,8] and level set methods [9,10]. Applications of level sets to multiphase flow calculations have been extensively described by Sussman et al. in [11–13].

A common problem to all methods is the accurate representation of the surface tension force. This is of great importance, since an inaccurate modeling of those forces may generate spurious currents that in turn can contaminate the solution and create oscillations that may be strong enough to destroy the interface [14]. The surface tension forces are evaluated according to the continuum surface force (CSF) method of Brackbill et al. [15]:[1]

$$\mathbf{F}^{\mathrm{CSF}} = \sigma \kappa \mathbf{n} \delta_\Gamma^\epsilon, \tag{2}$$

where σ is the surface tension coefficient, κ is the curvature, $\mathbf{n}$ is the unit normal to the interface, and δ_Γ^ϵ is a regularized Dirac delta function of the interface. The CSF has been used extensively over the last 14 years to model surface tension in various Eulerian formulations, in particular in the VOF method [16] and the level set method [17,18].

In the CSF-methodology, there are two ingredients for the accurate representation of surface tension forces. The first is the idea that for interfaces in equilibrium, the pressure gradient must exactly balance the surface tension force, and so result in a no-flow field [16]. This requires a proper discretization of the regularized delta function δ_Γ^ϵ. Furthermore, the surface tension force is computed at the same location as the pressure gradient is evaluated. In this way, at the discrete level, the pressure gradient discretization errors cancel out the surface tension discretization errors, which largely reduces the spurious currents [14,19]. The second ingredient is the accurate estimation of the interface curvature κ.

---

[1] The CSF method was initially developed in the context of the VOF methods and reads as:

$$\mathbf{F}^{\mathrm{CSF}} = \sigma \kappa \nabla f, \tag{1}$$

where f is the volume fraction and goes smoothly from zero to one across the interface.

In the case of VOF's, the computation of the curvature is a difficult task because of the lack of regularity of the volume fraction of liquid. To overcome this problem, some authors have developed geometric methods to reconstruct the interface from the values of the VOF (color) function [16,20]. Those techniques are sophisticated and computationally expensive in 3D. In the context of the level set method, normals and curvatures can in principle be easily expressed in terms of the level set function:

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}, \quad \kappa = -\nabla \cdot \mathbf{n}.$$

Thus, in order to evaluate the curvature, second-order derivatives of ϕ have to be computed. The fact that the level set method provides continuous, differentiable representation of the geometrical properties has often been recognized as an advantage of level set techniques compared with VOF techniques.

However, in this paper, we question this apparent easy differentiation of the geometric properties. This was also pointed out by Cummins et al. in [21] where they showed that, contrary to a popular belief, a straightforward differentiation of the level does not always give accurate and converging results of the curvature estimates. This is essentially due to high frequencies introduced in the level set function by the velocity field (that is solution of the Navier–Stokes equation) and/or a reinitialization procedure. In order to compute those geometric properties accurately either smoothing techniques or least squares approaches have to be used [17,21].

In our work, the level set function ϕ is advected using an upwind high order discontinuous Galerkin (DG) scheme in space. We show in [1] that this upwind DG discretization of the level set possesses spectral and spatial super-convergence properties, showing super accuracy for the resolved numerical wave numbers. Moreover, the derivatives are L_2 stable, *i.e.* producing curvatures that converge in the sense of L_2. However, as the level set is advected by the velocity field (that is solution of the Navier–Stokes equations discretized with a P1/P1 stabilized finite element method), high frequencies aliasing errors are introduced in the level set function. This in particular does not yield accurate and converging estimates for the interface curvature anymore and hence, prevents us from computing the derivatives with a direct derivation of the level set. It is important to note that those high frequencies aliasing errors are independent of the choice of our spatial discretization for the level set equation. Thus, a continuous finite-element discretization of the level set equation would suffer from a similar lack of convergence.

Fig. 1 shows a bubble (represented by the iso-zero of the level set function). A zoom at the interface shows the high frequencies contained in the solution. Those oscillating components will be magnified when derivates are calculated. Authors [17] have proposed to regularize the level set function using the following diffusion equation:

$$\int_{\Omega_e} \phi \hat{w} \, \mathrm{d}v = \int_{\Omega_e} \tilde{\phi} \hat{w} \, \mathrm{d}v + \int_{\Omega_e} \epsilon \nabla \tilde{\phi} \nabla \hat{w} \, \mathrm{d}v,$$

where $\tilde{\phi}$ is the filtered level set function and $\hat{w}$ a test function and ϵ is a diffusion parameter. The major drawback of this smoothing technique is its high computational cost, since it requires to solve a diffusion equation.

As far as we are concerned, we show that a least square computation of the curvature can be robust, second order and enables to filter out the undesired high frequencies.

Our discontinuous Galerkin level set method has good conservation properties and exhibits very small mass losses [1]. It is coupled with a stabilized finite element method for the solution of the incompressible fluids [2].

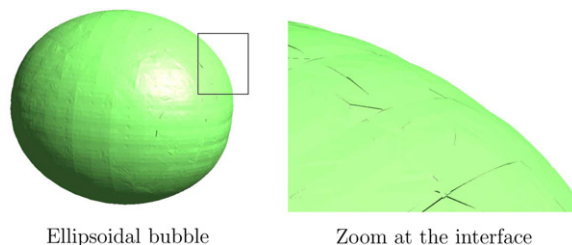Ellipsoidal bubble   Zoom at the interface

Fig. 1. Iso-zero of the level set function (left) and zoom at the interface (right). The oscillating components in the level set function will be magnified when derivates are calculated leading to poor approximation of the curvature at the interface.

The finite element approach enables us to integrate exactly the discontinuities of density and viscosity without introducing any artificial parameter and the pressure stabilization [22,23] enables to use equal order interpolations for velocity and pressure.

The proposed numerical method allows to simulate a wide range of flow regimes with large ratio of densities and to accurately capture the shape of the deforming interface of the bubble, while maintaining a good mass conservation.

The remainder of this paper is organized as follows. Section 2 is devoted to the mathematical formulation of the continuous two phase problem. Then we introduce the CSF model. In Section 3, we present the discretization methods that are used and in Section 4 we focus on the numerical approximation of the normal and the curvature. Finally, in Section 5, we present results of simulations.

2. Mathematical formulation

For incompressible two-phase flows, the Navier Stokes equations can be combined into a single set of equations for the velocity $\mathbf{u}$ in the entire domain of interest taking into account surface tension through a local volumetric surface tension force [24].

We consider that the interface is represented by the iso-zero of the level set function ϕ. The level set function ϕ is positive in the liquid (denoted fluid 1) and negative in the gas (denoted fluid 2).

The governing equations describing the motion are then given by the non-dimensional Navier Stokes equations and by the level set equation:

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla p}{\rho(\phi)} + \frac{1}{\rho(\phi)Re} \nabla \cdot (\mu(\phi)(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) + \frac{\mathbf{g}}{Fr^2} + \frac{\kappa \mathbf{n} \delta_\Gamma}{\rho(\phi)We} \tag{3}$$

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

$$\partial_t \phi + \nabla \cdot (\mathbf{u}\phi) = 0. \tag{5}$$

This is the continuous problem that we use to model our two-phase flow problem. It is also used in [11,17,24–26]. This formulation, after we regularize the delta function, is very similar to the one obtained by Brackbill et al. [15], the Continuum Surface Force (CSF) method.

From the level set function one can derive the density $\rho(\phi)$, viscosity $\mu(\phi)$, the normal $\boldsymbol{n}$ and the curvature κ. Density and viscosity are written as

$$\rho(\phi) = \rho_2(1 - H(\phi)) + \rho_1 H(\phi), \tag{6}$$

$$\mu(\phi) = \mu_2(1 - H(\phi)) + \mu_1 H(\phi), \tag{7}$$

where $H(\phi)$ is the Heaviside function that is equal to 1 in the liquid and 0 in the gas. The normal and curvature are given by:

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}, \quad \kappa = -\nabla \cdot \mathbf{n}. \tag{8}$$

Here $\mathbf{u}$ is the velocity field, p the pressure field, ρ and μ are the density and dynamic viscosity of the fluid all in non-dimensional forms; these are variables in whole domain but constant and in general different in each phase. $\mathbf{g}$ denotes the gravitational field.

The key flow parameters are the ratio of densities and viscosities (ρ_1/ρ_2, μ_1/μ_2), the Reynolds number $Re = \rho_R U_R L_R/\mu_R$, the Froude number $Fr = U_R/\sqrt{gL_R}$ and the Weber number $We = \rho_R U_R^2 L_R/\sigma_R$. σ is the surface tension coefficient and the subscript R denotes a reference value.

This formulation 3, 4 implicitly contains the boundary conditions at the interface as found in standard references:

$$[\mathbf{u}] = 0, \quad [-pI + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^\mathrm{T})] \cdot \mathbf{n} = \sigma\kappa\mathbf{n}, \tag{9}$$

where the brackets denote the jump across the interface.

2.1. Modeling of surface tension

The surface tension effects enter into the momentum equations (3) as a source term. The surface tension forces, i.e the last term in Eq. (3), are, according to the Continuum Surface Force (CSF) model [15], expressed as:

$$\mathbf{F}^{\mathrm{CSF}} = \frac{\kappa\mathbf{n}\delta_\Gamma^\epsilon}{\rho(\phi)We}, \tag{10}$$

where δ_Γ^ϵ is the regularized delta function at the interface.

A important ingredient for the accurate representation of surface tension forces is that for interfaces in equilibrium, the pressure gradient force must exactly balance the surface tension force, and so result in a no-flow field. This requires a proper discretization of the regularized delta function δ_Γ^ϵ.

In this work, we use the following regularized delta function (hat function):

$$\delta_\Gamma^\epsilon(\phi) = \begin{cases} 0 & \|\phi\| > \epsilon \\ \frac{1}{2\epsilon} & \|\phi\| \leqslant \epsilon \end{cases}, \tag{11}$$

where ϕ represents the signed distance to the interface and ϵ is the interface thickness defined as $1.5h$, and h is the mesh spacing. The reason for this is the following. For interfaces in equilibrium, the pressure term should balance exactly the surface tension term:

$$\frac{\nabla p^h}{\rho(\phi)} = \frac{\kappa\mathbf{n}\delta_\Gamma^\epsilon}{\rho(\phi)We}. \tag{12}$$

More specifically, the gradient of the pressure field should be equal (up to a constant) to the Dirac delta function (Fig. 2). At the discrete level this is only the case if the pressure field is discontinuous (2(a)). Meanwhile, as we use linear P1 finite elements to represent the pressure field, the pressure jump created by the surface tension will establish itself over a certain thickness and will vary linearly over this thickness (see Fig. 2(b)). Hence, an appropriate choice of smoothened Delta function is the *hat* function given by Eq. (11) and illustrated at Fig. 2(b).

In the example of Fig. 3, we have placed a bubble of radius $R = 1$ at rest in a box $[0,2] \times [0,2]$. The initial conditions are zero velocity and pressure field. The two fluids have same density $\rho_1 = \rho_2 = 1$ and viscosity $\mu_1 = \mu_2 = 1$. Due to the effect of surface tension ($\sigma = 10\ \mathrm{J/m^2}$) the pressure inside the bubble increases and
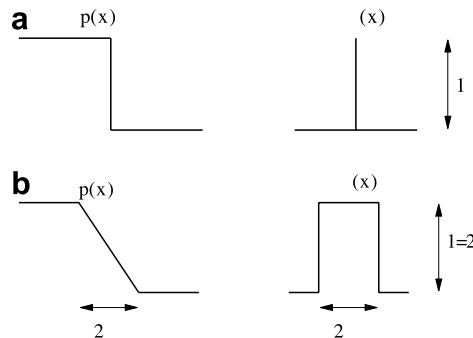

Fig. 2. The pressure gradient should balance exactly the Dirac delta function. (a) Sharp pressure jump and exact Dirac delta function, (b) linear pressure jump and "*hat*" Dirac delta function.
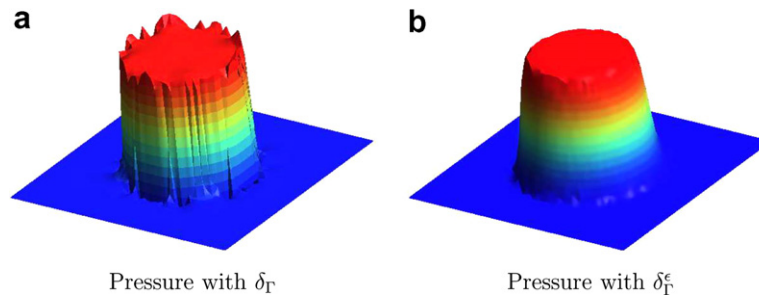
Fig. 3. Instantaneous pressure field for a static bubble. Computations are performed using (a) an exact $\delta_\Gamma(\phi)$ and (b) smoothened $\delta_\Gamma^\epsilon(\phi)$ delta function.

should reach the equilibrium state: $p_{\text{in}} - p_{\text{out}} = \sigma/R$ (Laplace law). The computational mesh is a non-regular mesh of size $h = 0.04$. We compare the results obtained using a the exact delta function δ_Γ and the smoothened delta function $\delta_\Gamma^\epsilon(\phi)$. The computations with the exact delta function included the localization of the interface and the computation of a surface integral. To achieve this, we used a recurse contouring algorithm described in [27]. We clearly see that the use of an exact Dirac delta function leads to oscillatory results in the pressure field while the use of a smoothened Dirac delta function gives a correct and smooth pressure jump.

2.2. Reinitialization of the level set

In the formulation described above, the interface will have a uniform thickness as long as ϕ remains a distance function. However, within time the level set does not necessary remain a distance function. One solution is to reshape the level set to a distance function. One of the major drawbacks of this method referred to as reinitialization is the difficulty in not moving the original location of the interface. Several methods have been developed, such as fast marching methods [10,28] and reinitializing techniques developed by Sussman et al. [11] based on the resolution of a first-order partial differential equation.
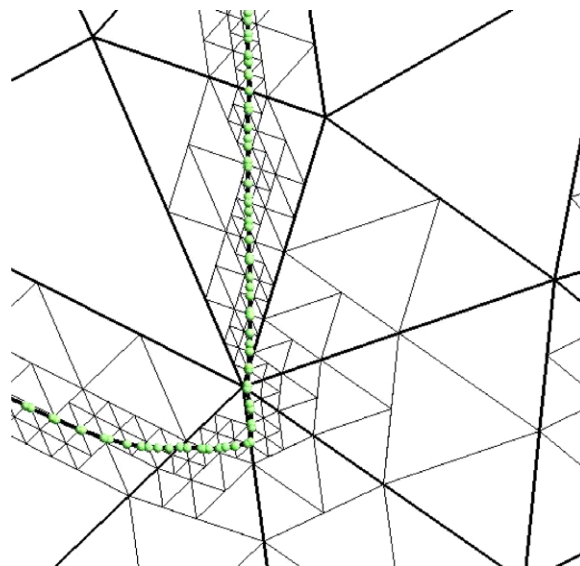

Fig. 4. Contouring algorithm to accurately localize the interface. In black, the computational mesh. In grey, the 4 level refined elements. In green, a set of points on the interface. (For interpretation of the references in colour in this figure legend, the reader is referred to the Web version of this article.)

In this paper we avoid the cost of the renormalisation procedure at each time step by computing a set of points N_i on the level set and by computing the delta function defined in (11) as a function of:

$$\phi_{\text{dist}}(\mathbf{x}) = (\min_i(\text{dist}(\mathbf{x}, N_i))). \tag{13}$$

The set of points is computed using a recursive contouring algorithm (described in [27]) with a fast search tree method. The approach is quite simple. It consists of dividing the elements that are crossed by the interface recursively into sub-elements and perform a linear approximation on every edge of those sub-elements to find the points on the interface (points with a zero value of the level set). This recursive algorithm is of course only mandatory if the level set is approximated by high order polynomials $p > 1$.

Fig. 4 shows an example of the recursive contouring algorithm of level 4 to localize the interface.

Those points N_i are then stored in a search tree (Approximate Nearest Neighbor (ANN) [29]). Next, using the search tree, we compute the smallest distance (Eq. (13)).

Meanwhile, the complete reinitialization procedure is nonetheless applied 5–10 times over 500 time steps during the simulation in order to maintain good conservation properties. The reason for this is that the level set might become very flat at the vicinity of the interface and as a consequence, an accurate determination of the interface is not possible anymore.

To the best knowledge of the authors, not performing a reinitialisation procedure at the end of each time step is a novel approach in the context of the level set based methods [11,30,31].

3. Numerical framework

To derive the finite element discretization, we assume that we have some appropriate finite dimensional function spaces for the trial $(\mathscr{S}_\mathbf{u}^h, \mathscr{S}_p^h, \mathscr{S}_\phi^h)$ and weighting $(\mathscr{V}_\mathbf{u}^h, \mathscr{V}_p^h, \mathscr{V}_\phi^h)$ function spaces corresponding to the velocity, pressure and the level set function.

Concerning the function spaces, we use linear continuous approximations for the velocity and pressure and piecewise continuous approximations of order p on each element for the level set. The physical domain Ω of boundary Γ is discretized into a collection of $\mathscr{N}_e$ elements (Ω_e) called a mesh.

3.1. The Navier–Stokes solver

We use a pressure stabilized finite element (PSPG) [23,32] method for the discretization of the Navier Stokes equations. For the stabilization of the upwind term, we follow the work of Barth [33] and use a finite volume stabilization with the finites volumes being the median cells of the mesh.

The stabilized finite element formulation of Eqs. (3) and (4) can be written as follows: find $\mathbf{u}^h \in \mathscr{S}_\mathbf{u}^h$ and $p^h \in \mathscr{S}_p^h$ such that $\forall \mathbf{v}^h \in \mathscr{V}_\mathbf{u}^h$ and $\forall q^h \in \mathscr{V}_p^h$:

$$\int_{\Omega_e} \mathbf{v}^h \cdot \left(\mathbf{u}_t^h + \mathbf{u}^h \cdot \nabla \mathbf{u}^h + \frac{\nabla p^h}{\rho} - \frac{\kappa \mathbf{n} \delta_\Gamma^\epsilon}{\rho We} - \frac{\mathbf{g}}{Fr^2}\right) \mathrm{d}v + \int_{\Omega_e} \nabla \mathbf{v}^h \cdot \frac{\mu}{\rho Re}(\nabla \mathbf{u} + \nabla \mathbf{u}^\mathrm{T}) + \mathrm{ST} = \int_\Gamma \mathbf{v}^h \cdot \mathbf{h}^h \, \mathrm{d}s, \tag{14}$$

where $\mathbf{h}^h$ denotes the Neumann-type boundary condition associated with the momentum equations $\mathbf{h}^h = \sigma \cdot n$ on the interface Γ, where $\sigma = -pI + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^\mathrm{T})$ is the stress tensor.

The stabilization term ST

$$\mathrm{ST} = \sum_e \int_{\Omega_e} \tau_e \nabla q^h \mathbf{R}(p^h, \mathbf{u}^h) \, \mathrm{d}v \tag{15}$$

contains the residual of the momentum equation

$$\mathbf{R}(p^h, \mathbf{u}^h) = \mathbf{u}_t^h + (\mathbf{u}^h \cdot \nabla)\mathbf{u}^h - \frac{1}{\rho Re}\nabla \cdot \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^\mathrm{T}) + \frac{\nabla p^h}{\rho} - \frac{\mathbf{g}}{Fr^2} - \frac{\kappa \mathbf{n} \delta^\epsilon}{\rho We}. \tag{16}$$

The stabilization parameter τ_ϵ is of order $\mathscr{O}(\rho h_e^2/\mu)$ in the diffusion dominated case and of order of $\mathscr{O}(h_e/\|\mathbf{u}\|)$ in the advection dominated case [23,34,35].

For the numerical solution of (14), a second-order three-point backward difference scheme is employed for the time-integration. An inexact Newton method based on a finite difference Newton–Krylov algorithm [36] is used to solve at each time step the system of nonlinear equations. The iterative solution of the large sparse linear system of equations that arises at each Newton iteration is solved by the GMRES method preconditioned by the RAS [37] algorithm.

3.1.1. Discontinuous density and viscosity

As the density and the viscosity are constant in each phase but may be discontinuous across the interface (the interface does not conform with the mesh), some integrals of Eq. (14) are discontinuous. Those discontinuous integrals are computed exactly by numerical integration:

$$\int_{\Omega_e} \mathbf{v}^h \frac{\nabla p^h}{\rho} \, dv = \int_{V^1} \mathbf{v}^h \frac{\nabla p^h}{\rho_1} \, dv + \int_{V^2} \mathbf{v}^h \frac{\nabla p^h}{\rho_2} \, dv, \tag{17}$$

$$\int_{\Omega_e} \nabla \mathbf{v}^h \frac{\mu}{\rho Re} \nabla \mathbf{u}^h \, dv = \int_{V^1} \nabla \mathbf{v}^h \frac{\mu_1}{\rho_1 Re} \nabla \mathbf{u}^h \, dv + \int_{V^2} \nabla \mathbf{v}^h \frac{\mu_2}{\rho_2 Re} \nabla \mathbf{u}^h \, dv, \tag{18}$$

where the subscripts 1 and 2 denote the fluid 1 and the fluid 2. More details of implementation can be found in [2].

3.1.2. Evaluation of the surface tension term

For the evaluation of the surface tension term in Eq. (14), a density scaling of the CSF, given by,

$$\tilde{\mathbf{F}}^{CSF} = \frac{\rho(\phi)}{\langle \rho \rangle} \mathbf{F}^{CSF}, \tag{19}$$

where $\rho(\phi)$ is the local density and $\langle \rho \rangle$ is an average density, was found to improve the CSF method's ability to model surface tension at high density ratio (>10) interfaces [15,38]. With this density scaling the CSF becomes a true body force, where fluid acceleration is independent of its density. Eq. (19) can also be thought as a modification of δ_Γ since this density scaling will shift the CSF distribution away from the denser portion of the interface transition region. However, it is not clearly stated in [15,38] which average density should be chosen.

We have observed that the following specific choice for the mean density $\langle \rho \rangle$:

$$\langle \rho \rangle = \frac{1}{3} \rho_1 + \frac{2}{3} \rho_2 \tag{20}$$

was producing convergent results for every combination of ρ_1 and ρ_2. We have therefore decided to investigate more this issue.

To address this issue, we have implemented a simple MATLAB 1D code to solve the following 1D finite element problem:

$$\int_{-1}^{1} \mathbf{v}^h \cdot \frac{\nabla p^h}{\rho(\phi)} \, dr = \int_{-1}^{1} \mathbf{v}^h \cdot \frac{\kappa \mathbf{n} \delta_\Gamma^\epsilon}{\langle \rho \rangle We} \, dr, \quad \forall \mathbf{v}^h \in \mathcal{V}_{\mathbf{u}}^h \tag{21}$$

$$p(1) = 0. \tag{22}$$

This simplified problem is solution of the Navier Stokes equations in the absence of gravity and for which the curvature κ is constant. In that special case, the surface tension force is canceled by a pressure gradient and the fluid remains at rest. For a bubble of radius R, the solution to Eq. (21) is the Young–Laplace law (see Example 5.1 in Section 5): $p_{in} - p_{out} = \sigma/R$ and this solutions is independent of the ratio of density of the two fluids. For the 1D MATLAB problem, the bubble has a radius $R = 1$ and is centered at $r = -1$. The domain is $\Omega : x \in [-1,1]$ and the mesh is made of $N = 13$ equally spaced segments ($h = 2/13$). The surface tension coefficient is $\sigma = 1$ and the interface thickness is chosen to be $\epsilon = 2.5h$. We see in Fig. 5 that when the two fluids have the same density, the pressure jump develops over a thickness of 2ϵ and is centered at the interface. This is also the case when the exact discontinuous density is considered for the evaluation of Eq. (21). Taking a mean value for the density in Eq. (21) shifts the pressure jump towards the densest fluid. We also observe that the pressure jump starts at the element next to the element containing the interface and that it develops only over a thickness of $2.5h - 0.5h = 2h$. In that case, the best choice for $\langle \rho \rangle$, i.e the one that gives the correct pressure jump is:
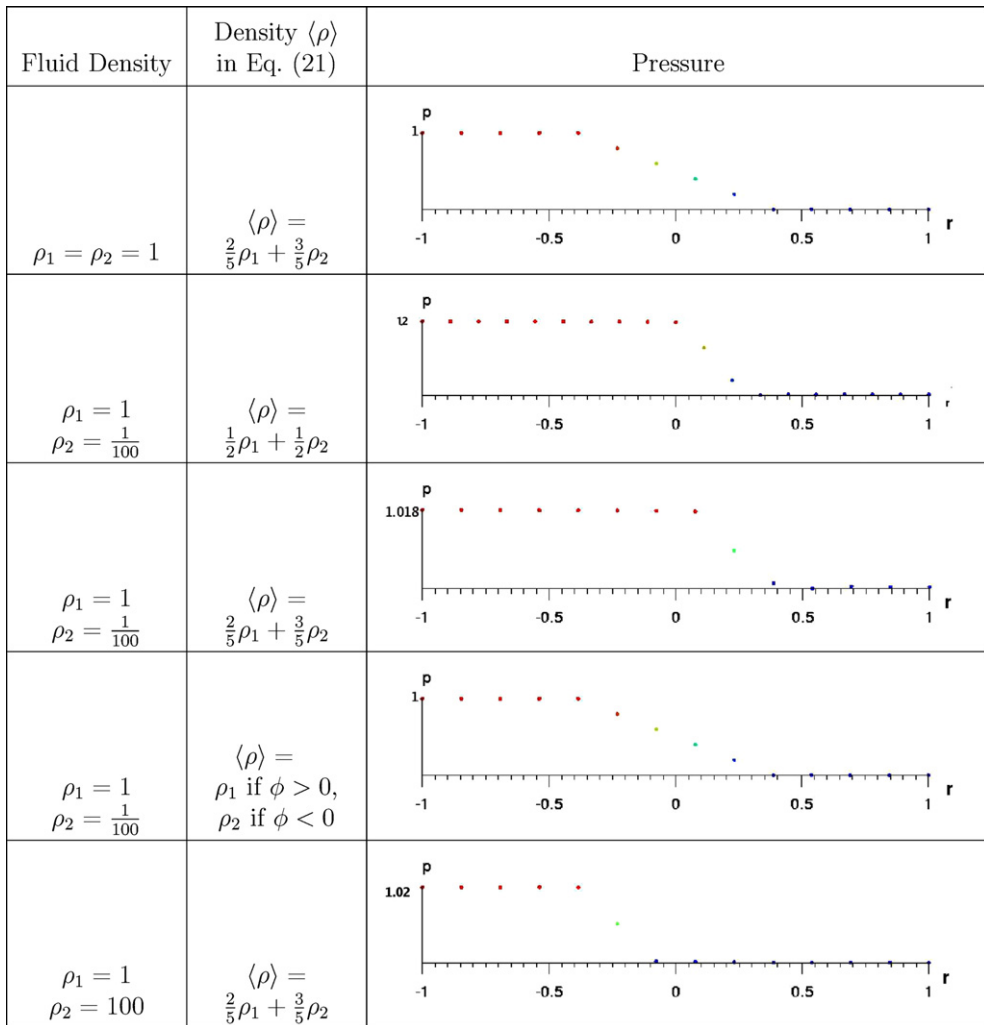
Fluid Density	Density $\langle\rho\rangle$ in Eq. (21)	Pressure
$\rho_1 = \rho_2 = 1$	$\langle\rho\rangle = \frac{2}{5}\rho_1 + \frac{3}{5}\rho_2$	
$\rho_1 = 1$ $\rho_2 = \frac{1}{100}$	$\langle\rho\rangle = \frac{1}{2}\rho_1 + \frac{1}{2}\rho_2$	
$\rho_1 = 1$ $\rho_2 = \frac{1}{100}$	$\langle\rho\rangle = \frac{2}{5}\rho_1 + \frac{3}{5}\rho_2$	
$\rho_1 = 1$ $\rho_2 = \frac{1}{100}$	$\langle\rho\rangle = \rho_1$ if $\phi > 0$, ρ_2 if $\phi < 0$	
$\rho_1 = 1$ $\rho_2 = 100$	$\langle\rho\rangle = \frac{2}{5}\rho_1 + \frac{3}{5}\rho_2$	

Fig. 5. Pressure jump for the 1D stationary bubble test case for different ratio's of density and different values for the density in (21). The interface is located at $x = 0$ and the interface is smoothed over a distance 2ϵ, with $\epsilon = 2.5h$. The density ρ_1 hold for $x < 0$ and the density ρ_2 for $x > 0$.

$$\langle\rho\rangle = \frac{2h}{5h}\rho_1 + \frac{3h}{5h}\rho_2.$$

In our 3D problems, we have chosen an interface thickness of $\epsilon = 1.5h$. Following our 1D conclusions, the mean density value $\langle\rho\rangle$ is the value given by Eq. (20).

3.1.3. Stability issues and time scale

As we use an implicit time integration method for the Navier Stokes equations, we avoid the standard convective and diffusive stability restrictions:[2]

$$\Delta t_{\text{num}}^{(\text{conv})} \leqslant \frac{h}{\|\mu\|}, \quad \text{and} \quad \Delta t_{\text{num}}^{(\text{diff})} \leqslant \frac{\rho h^2}{\mu}. \tag{23}$$

---

[2] Those time step restrictions are highly restrictive in case one has to model the boundary layers (proportional to $1/\sqrt{Re}$) and in case one has to mesh very complex geometries (very small mesh size h).

However, our computational method has two stability constraints due to the explicit treatment of the gravity and surface tension forces. Those constraints can be easily derived by the following empirical analysis [39]:

- **the stability restriction due to the gravity** can be expressed in the form of a CFL-like condition $\frac{u_g \Delta t_{num}^{(g)}}{h} \leqslant 1$, where the corresponding gravity-induced velocity $u_g = g \Delta t_{num}^{(g)}$. Then we obtain the maximal allowed time-step size as :

$$\Delta t_{num}^{(g)} \approx \sqrt{\frac{h}{\|\mathbf{g}\|}}; \tag{24}$$

- **the stability restriction due to the capillarity** can be expressed in the form of a CFL-like condition $\frac{u_{ca} \Delta t_{num}^{(ca)}}{h} \leqslant 1$, where the corresponding capillary-induced velocity $u_{ca} = \frac{\sigma \kappa_h \delta_h}{\rho} \Delta t_{num}^{(ca)} \approx \frac{\sigma}{h^2 \rho} \Delta t_{num}^{(ca)}$. Then we obtain the maximal allowed time-step size as [15]:

$$\Delta t_{num}^{(ca)} \approx \sqrt{\frac{\rho}{\sigma}} h^{3/2}. \tag{25}$$

Finally, the global stability constraint is given by

$$\Delta t_{NS} \leqslant \min \left\{ \Delta t_{phys}^{(re)}; \Delta t_{num}^{(g)}; \Delta t_{num}^{(ca)} \right\}. \tag{26}$$

3.2. The interface solver

We employ a quadrature free discontinuous Galerkin level set method [1] for the discretization of the level set equation.

The discontinuous finite element formulation of Eq. (5) can be written as follows: find $\phi^h \in \mathscr{S}_\phi^h$ such that $\forall \hat{w}^h \in \mathscr{V}_\phi^h$:

$$\int_{\Omega_e} \hat{w}^h \cdot \partial_t \phi \, dv + \int_{\Omega_e} \nabla \hat{w}^h \cdot (\mathbf{u}^h \phi) \, dv - \int_{\partial \Omega_e} \hat{w}^h \cdot f \, ds, \tag{27}$$

where $f = (\phi \mathbf{u} \cdot n)$ is the normal trace of the fluxes and is chosen to be the upwind flux.

The resulting system is solved in time using an explicit Runge–Kutta method. The time step restriction is given by:

$$\Delta t_{DG} \leqslant \frac{h}{\|\mathbf{u}\|(2p+1)}. \tag{28}$$

More details of implementation of the discontinuous Galerkin method as well as super-convergence analysis can be found in [1].

3.3. Coupling algorithm

The flow solver uses continuous linear approximations for the velocity ($\mathbf{v}^h = N^1$) while the interface DG solver uses piecewise continuous high order p approximations for the level set ϕ ($\hat{w}^h = N^p$):

$$\mathbf{u} = \sum_{i=1}^{4} \mathbf{u}_i N_i^1 \quad \text{and} \quad \phi = \sum_{i=1}^{n_p} \phi_i N_i^p.$$

The reason why higher order polynomials are used for discretizing the level set is that Eq. (5) involves $\nabla \phi$ and $\mathbf{u}$. We choose a level set for which the gradient is at least in the space of the velocity ($p = 2$). In practice, we have observed that $p = 2$ is a good compromise between accuracy (few mass loss) and computational cost [2].

Projection operators are used to project the velocity space to the level set space and conversely. The elementary projection operator $\mathbb{P}$ that projects the velocity variable from a polynomial space of order $p = a$ to a space of order $p = b$ is given by:

$$u^b = \mathbb{P}u^a = M^{-b}M^{ab}u^a, \tag{29}$$

where the mass matrix for the element Ω_e is given by:

$$M_{ij}^{ab} = \int_{\Omega_e} N_i^a N_j^b \, \mathrm{d}v. \tag{30}$$

On the other side, we take the average of ϕ at the nodes of the tetrahedron as ϕ's for the flow solver. The overall algorithm is summarized here below (see Algorithm 1).

Algorithm 1. Coupling algorithm

At initial time t^0
- Choose an appropriate time step Δt satisfying (26).
- Initialize the level set ϕ at $t = 0$

For each time t^n, $n = 1, 2, \ldots$
 (1) Solve Eq. (14) in time $t \in [t, t + \Delta t]$ to find $\mathbf{u}^h(t + \Delta t)$ and $p^h(t + \Delta t)$
 (2) Project the velocity field onto the dofs of the level set (29)
 (3) If $n\%100 = 0$, reinitialize the level set
 (4) Solve Eq. (27) in time $t \in [t, t + \Delta t]$ using sub-time steps to find $\phi^h(t + \Delta t)$
 (5) Project the level set function onto the dofs of the velocity (average)
 (6) Increment in time $t = t + \Delta t$ and go back to step (2)

---

4. Numerical computation of normals and curvature

In this section, we present and compare two numerical methods to compute the normal $\boldsymbol{n}$ and the curvature κ at the interface (represented by the iso-zero of the level set function ϕ). The level set being represented by piecewise continuous polynomials of order p (DGM), the normal and curvature are given by:

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} = \frac{\vec{G}}{|\vec{G}|}, \quad \kappa = -\nabla \cdot \mathbf{n}. \tag{31}$$

The first method computes those geometric properties in the context of the DGM, while the second approach uses a least square procedure. An accurate computation of those geometric properties is essential for the computation of the surface tension forces (Eq. (10)). We show that a high order DGM computation of those terms leads to poor results while a least square method filters out the high frequencies and gives us good results with a second-order accuracy.

4.1. The discontinuous Galerkin method

As the level set function ϕ is represented by piecewise continuous polynomials of order p in the context of the discontinuous Galerkin method, stable gradients of ϕ should take into account solution jumps (discontinuities of the level set). Second-order derivatives of ϕ should also take into account gradient jumps. Despite being L^2-stable (i.e. producing curvatures that converge in the sense of L^2), we will show that this procedure leads to highly oscillatory curvature fields that cannot be used for computing surface tension forces.

The gradient $\vec{G}$ of the level set function ϕ function is computed as:

$$\int_{\Omega_e} (\vec{G} - \nabla\phi)\hat{w} \, \mathrm{d}v = 0 \quad \forall\hat{w} \tag{32}$$

and by subsequently integrating this by part we get:

$$\int_{\Omega_e} \vec{G}\hat{w} \, \mathrm{d}v = -\int_{\Omega_e} \phi\nabla\hat{w} \, \mathrm{d}v + \int_{\partial\Omega_e} \phi\hat{w}\mathbf{n} \, \mathrm{d}s, \quad \forall\hat{w} \tag{33}$$

where $\mathbf{n}$ is the unit normal to the boundaries of the element.

Since the DGM allows discontinuities at the interface, ϕ is not strictly defined on $\partial\Omega_e$, and we need to introduce some choice here. We take the centered value [40]: $\phi^e = \frac{1}{2}(\phi^e + \phi_r^e)$ where ϕ_r^e is the value of ϕ on the boundary ∂e as computed on the element Ω_{e_r}, the neighbor of Ω_e. Subsequently we compute the normal values of the gradient for each degree of freedom i:

$$\mathbf{n}_i = \frac{\vec{G}_i}{\sqrt{(G_{x,i}^2 + G_{y,i}^2 + G_{z,i}^2)}}. \tag{34}$$

The curvature defined as:

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

is then subsequently multiplied by a test function $\hat{w}$, integrated over the domain and integrated by parts:

$$\int_{\Omega_e} \kappa \hat{w}\, \mathrm{d}v = \int_{\Omega_e} \nabla \hat{w} \cdot \mathbf{n}\, \mathrm{d}v - \int_{\partial\Omega_e} \mathbf{n} \cdot \mathbf{n}\hat{w}\, \mathrm{d}s \quad \forall \hat{w}. \tag{35}$$

We can show that for DG schemes with polynomials of order p and a mesh of mesh size h, the L_2 norm of the error is of order h^{p+1}. Therefore, we expect the normal to converge like h^p and the curvature to converge like h^{p-1}.

4.2. The least squares method

The least squares approach has been proved to be optimal for linear and quadratic reconstructions in the sense that, if the reconstruction function is not exactly a polynomial of degree p but one that has been perturbed by the addition of Gaussian noise, it minimizes the expected deviation from the unperturbed polynomial in a certain rather natural norm [41].

For the evaluation of the nodal gradient $\vec{G} = \nabla\phi$, a least squares procedure is used in which the data surrounding each node is assumed to behave linearly or quadratically. We refer to Fig. 6 as a 2D example of the linear and quadratic least squares approach. Considering a linear approximation, the data at the center of gravity of each element surrounding the i_{th} node may be expressed as:

$$\phi_i(\mathbf{r}) = \phi_i + \Delta\mathbf{r}^{\mathrm{T}}\check{\nabla}\phi_i, \tag{36}$$

where $\Delta\mathbf{r} = \mathbf{r} - \mathbf{r}_i$ and $\check{\nabla}\phi_i$ is a first-order approximation of the gradient. Considering a quadratic reconstruction at the center of gravity of each element will give us a second approximation of the gradient $\check{\nabla}\phi_i$ and a first-order approximation of the Hessian matrix $\check{\mathscr{H}}_i$:

$$\phi_i(\mathbf{r}) = \phi_i + \Delta\mathbf{r}^{\mathrm{T}}\check{\nabla}\phi_i + \frac{1}{2}\Delta\mathbf{r}^{\mathrm{T}}\check{\mathscr{H}}_i\Delta\mathbf{r}. \tag{37}$$

By expressing the data in a similar way at each of the $\mathscr{N}$ elements of a chosen stencil, a $\mathscr{N} \times 3$ ($\mathscr{N} \times 4$ in 3D) system of equations can be solved to obtain a first (Eq. (38)) or second-order (Eq. (39)) approximation of the gradient at node i:
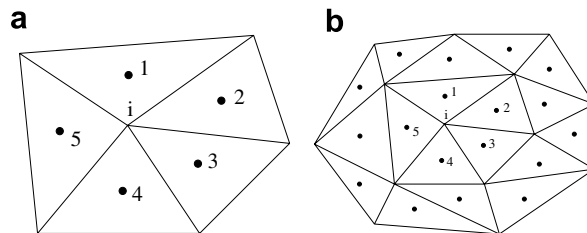

Fig. 6. Stencil used for the reconstruction of the gradient at node i. (a) Distance-1 stencil for the linear reconstruction (left), and (b) distance-2 stencil for the quadratic reconstruction (right).

$$
\begin{pmatrix}
1 & \Delta x_1 & \Delta y_1 & \Delta z_1 \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
1 & \Delta x_{\mathcal{N}} & \Delta y_{\mathcal{N}} & \Delta z_{\mathcal{N}}
\end{pmatrix}
\begin{pmatrix}
\phi_i \\
\breve{\partial}_x \phi_i \\
\breve{\partial}_y \phi_i \\
\breve{\partial}_z \phi_i
\end{pmatrix}
=
\begin{pmatrix}
\phi(\mathbf{r}_1) \\
\vdots \\
\phi(\mathbf{r}_{\mathcal{N}})
\end{pmatrix}
\tag{38}
$$

$$
\begin{pmatrix}
1 & \Delta x_1 & \Delta y_1 & \Delta z_1 & \frac{1}{2}\Delta x_1^2 & \frac{1}{2}\Delta y_1^2 & \frac{1}{2}\Delta z_1^2 & \Delta x_1 \Delta y_1 & \Delta x_1 \Delta z_1 & \Delta y_1 \Delta z_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & \Delta x_{\mathcal{N}} & \Delta y_{\mathcal{N}} & \Delta z_{\mathcal{N}} & \frac{1}{2}\Delta x_{\mathcal{N}}^2 & \frac{1}{2}\Delta y_{\mathcal{N}}^2 & \frac{1}{2}\Delta z_{\mathcal{N}}^2 & \Delta x_{\mathcal{N}} \Delta y_{\mathcal{N}} & \Delta x_{\mathcal{N}} \Delta z_{\mathcal{N}} & \Delta y_{\mathcal{N}} \Delta z_{\mathcal{N}}
\end{pmatrix}
\begin{pmatrix}
\phi_i \\
\breve{\partial}_x \phi_i \\
\breve{\partial}_y \phi_i \\
\breve{\partial}_z \phi_i \\
\breve{\partial}_{xx} \phi_i \\
\breve{\partial}_{yy} \phi_i \\
\breve{\partial}_{zz} \phi_i \\
\breve{\partial}_{xy} \phi_i \\
\breve{\partial}_{xz} \phi_i \\
\breve{\partial}_{yz} \phi_i
\end{pmatrix}
=
\begin{pmatrix}
\phi(\mathbf{r}_1) \\
\vdots \\
\phi(\mathbf{r}_{\mathcal{N}})
\end{pmatrix}.
\tag{39}
$$

In order to solve (38), at least 4 coefficients ($\mathcal{N} = 4$) are to be determined while for solving (39), the number of coefficients raises to 10 ($\mathcal{N} = 10$). This implies that at least 4 (resp. 10) pieces of information have to be provided to compute the required accurate derivatives for the linear (resp. quadratic) reconstruction at the node i. Those pieces of information constitute the reconstruction stencil of the considered node. We take a distance-1 neighbors stencil for the linear case (vertex-neighbors) and a distance-2 neighbors stencil for the quadratic case (vertex-neighbors of vertices of vertex-neighbors). The distance-1 and distance-2 neighbors stencil involve more information than necessary according to the above number of unknowns. The only exception is with the distance-1 neighbors for the nodes located at the boundaries of the domain.

The resulting over-determined system of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ can then be solved in a least squares sense:

$$
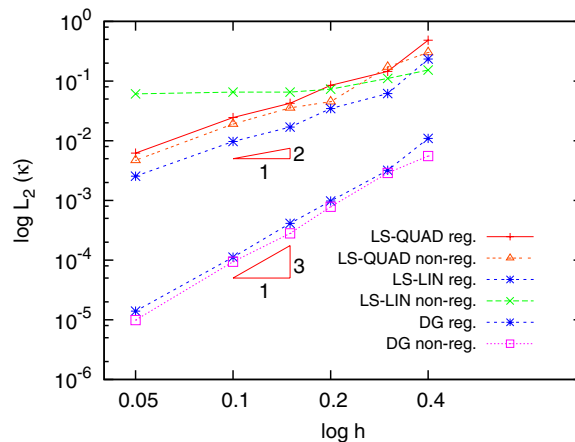\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}.
\tag{40}
$$

Fig. 7. Convergence analysis of the curvature approximation κ. Error $E = \|\kappa - \kappa_e\|_\epsilon$ as a function of the mesh size h for the stationary circular bubble test case. We compare the three following methods on regular (reg.) and non-regular (non-reg.) grids: the discontinuous Galerkin (DG) method, the least squares method using a linear reconstruction (LS-LIN) and the least squares method using a quadratic reconstruction (LS-QUAD).

Subsequently, we compute the normal following (34) and compute the curvature by least squares in the same way as described above.

4.3. Comparison

First we compare the accuracy of the curvature approximation with the two above presented methods. The accuracy of the curvature computation is evaluated by computing the L_2 error in curvature using subsequent mesh refinements. We consider a disc of radius $R = 1$ placed at the center of a 4×4 square. The level set is initially defined as the signed distance function $\phi = \sqrt{x^2 + y^2} - 1$. We define the following norm of the error for the curvature:

$$E = \|\kappa - \kappa_e\|_\epsilon = \sqrt{\int_\Omega (\kappa - \kappa_e)^2 \delta_\Gamma^\epsilon \, dv}, \tag{41}$$

where κ_e is the exact value of the curvature

$$\kappa_e = \frac{-1}{\sqrt{(x^2 + y^2)}}.$$
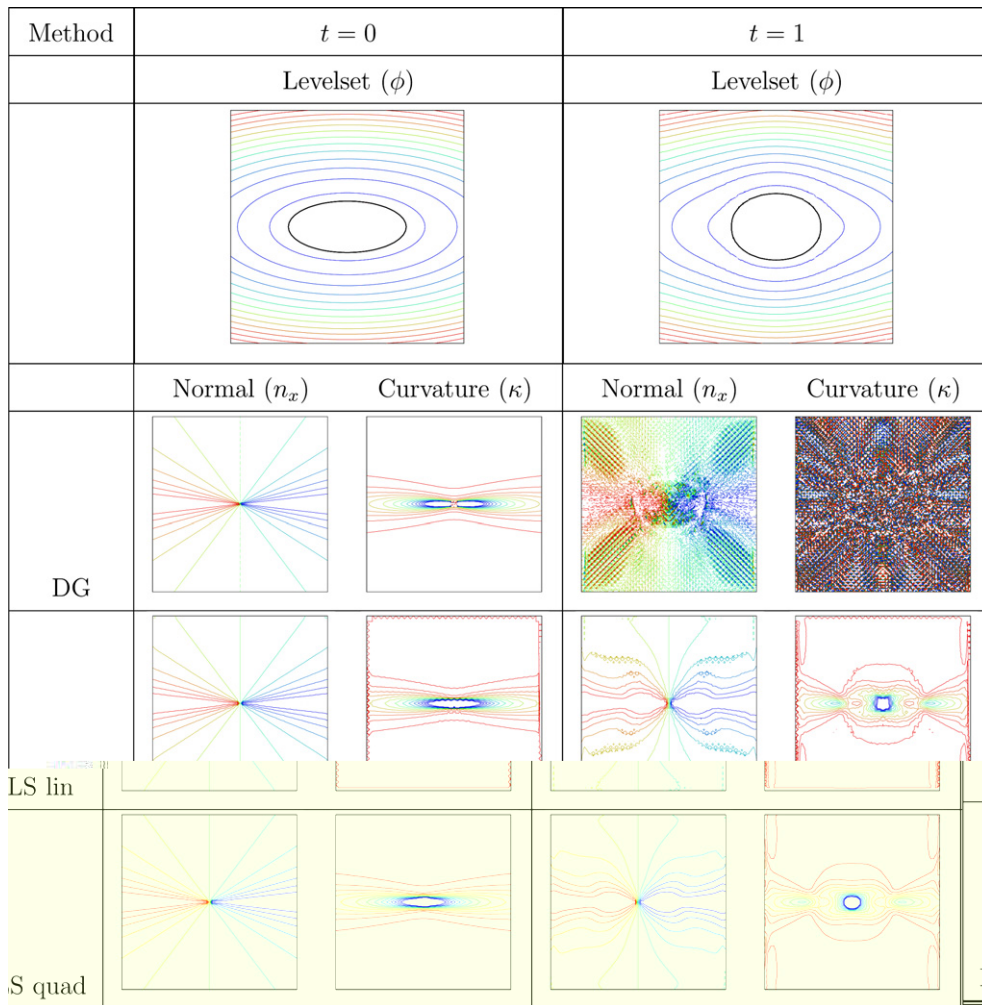
Fig. 8. *x*-Component of the normal ($=n_x$) and curvature κ at $t = 0$ s and $t = 1$ s. Computations are performed with two different numerical methods: the discontinuous Galerkin (DG) method and the least squares (LS) approach (linear and quadratic reconstructions).

and δ_Γ^ϵ is the smoothed delta function of Eq. (11) for which we have taken a constant thickness $\epsilon = 0.5$. For this example, we have taken polynomials of order $p = 4$ to approximate the level set function. Fig. 7 shows the L_2 norm of the error in curvature as a function of the mesh size for regular and non-regular (denoted respectively reg. and non-reg.) grids. We see that the curvature approximation is of order $h^{p-1} = h^3$ for the discontinuous Galerkin (DG) method for regular and non-regular grids. As far as the least squares method is concerned, the curvature approximation is of order h^2 when using a quadratic reconstruction on regular and non-regular grids and when using a linear reconstruction on regular grids but it does not converge using a linear reconstruction on a non-regular grid!

Second we analyze normal and curvature fields when the level set evolves in time. We consider the test case of an ellipsoidal-shaped bubble of size semi-axis $a = 0.8$, $b = 0.35$ with a surface tension coefficient $\sigma = 0.1$. The bubble recovers a circular shape due to the surface tension. We use a regular mesh of size $h = a/13$. Fig. 8 shows the x-component of the normal and the curvature fields computed with the DGM and the least squares method (linear and quadratic reconstruction) at the initial time as well as at an intermediate time. We clearly see that the DGM fails to predict an accurate curvature. This is due to the high frequencies contained in the level set solution. On the other hand, the least squares method can be seen as a smoothing method that filters out the high frequencies. We can also observe that the distance-1 stencil used for the linear reconstruction is not sufficient to compute the derivatives for the nodes located at the boundaries.

From the observations made in this section, we establish the following rule. If we use a regular grid, we will compute the curvature by means of the least squares approach using a linear reconstruction (less computationally expensive), while if we use a non-regular grid, we will compute the curvature by means of the least squares approach using a quadratic reconstruction.

5. Numerical results and discussion

In this section, our numerical method is tested and applied to several 2D and 3D two-phase flows problems. As we are only working with tetrahedral meshes, we use for the computation of the 2D flows a 2D mesh that is extruded over a thin layer in the third dimension.

For those computations, we chose polynomials of order $p = 2$ to approximate the level set function. The linear system arising from the linearisation of the discretized Navier Stokes equations is solved with a GMRES update technique with a Krylov space of 50. The time step used for the implicit solution of the Navier Stokes equations is chosen to be maximum 10 times the time step given by the CFL condition when solving explicitly the level set equation. The level set is reinitialized for unsteady 3D computations every 100 time steps.

Our methods are implemented in standard C++, compiled with INTEL icc v8.0, and run on the NEC/Clustervision Linux cluster installed at CENAERO.

5.1. Parasitic currents

The first test case is the verification of the stationary Laplace solution for a circular droplet. This test case was suggested by Popinet and Zaleski in [19] and used later on by [14,16,42]. A circular bubble of one phase is immersed into a fluid of another phase. In the absence of viscous, gravitational or external forces a stationary solution exists (zero velocity field) and the Young–Laplace law should be verified exactly:

$$\Delta p = \frac{\sigma}{R} \ (2D), \quad \Delta p = \frac{2\sigma}{R} \ (3D), \tag{42}$$

where σ is the surface tension coefficient and R the bubble radius. However, as shown in [14,16], subtle numerical inaccuracies can generate parasitic currents in the vicinity of the interface. These unphysical velocities depend strongly on many factors such as grid resolution, viscosity, surface tension, etc.

We placed a circular drop of diameter $d = 2$ in a square domain of size $L = 4$ with a non-regular grid of mesh size h. We set the density and viscosity equal to 1 everywhere in the domain. The gravity is neglected; the surface tension coefficient σ will be varied, thus yielding different values of the Laplace number $La = \sigma \rho d / \mu^2$. The computations are measured after 250 characteristic time scales ($t_{phys} = 250 t_{scale}$ with $t_{scale} = d\mu/\sigma$). It was conjectured by Lafaurie [43] that the amplitude of the spurious currents must be

Table 1
Independence of the capillary number Ca with respect to the Laplace number La (mesh size of $h = L/32$, L is the size of the domain)

$La = \sigma\rho d/\mu^2$	12	120	1200	12,000
$Ca = U\sigma/\mu$				
Our method	8.51E^{-5}	8.62E^{-5}	8.59E^{-5}	8.31E^{-5}
Front tracking [19]	8.5E^{-6}	6.76E^{-6}	5.71E^{-6}	5.99E^{-6}
VOF [44]	2.18E^{-6}	2.18E^{-6}	2.18E^{-6}	2.22E^{-6}

Table 2
Convergence of the capillary number Ca with spatial resolution (the Laplace number is $La = 200$, L is the size of the domain)

h	$Ca = U\sigma/\mu$
$L/30$	1.29E^{-4}
$L/60$	6.43E^{-5}
$L/120$	3.18E^{-5}
$L/240$	1.12E^{-6}

proportional to σ/μ, which is equivalent to having an approximately constant value of the capillary number $Ca = U\mu/\sigma$. Table 1 illustrates the constant character of Ca over a broad range of Laplace numbers. A mesh size of $h = L/32$ was used for those computations and compares the results with results from Popinet et al. [19] and Shin et al. [14,44]. Table 2 shows that our method exhibits a first-order convergence. This series of convergence tests were performed for a value of the Laplace number of $La = 100$ and with an increasing spatial resolution.

Yet, parasitic currents generated by our method are quite small in magnitude and do not degrade the solution when the surface tension coefficient is high.

5.2. Capillary pressure

At non-zero surface tension and zero gravity, any fluid particle whose shape deviates from circular (spherical in 3D) should recover the circular (spherical) status and stable shape. When the drop has reached its final state, the velocity field must vanish and the pressure drop at the interface must satisfy the Young–Laplace law (Eq. (42)).

A two-dimensional and three-dimensional test case are presented. The fluid properties outside and inside the bubble are $\rho_1 = \rho_2 = 1$, $\mu_1 = \mu_2 = 0.1$, $g = 0$. With $\sigma = 0.1$, which is considered to be a large surface tension coefficient, the flow quickly reaches the equilibrium state. The chosen time step is $\Delta t = 0.05$ s.

First, a square of length $a = \sqrt{\pi}/4$ whose area is equivalent to the area of a circle of radius $R = 1/4$ is placed at the center of a unit square discretized with a $60 \times 60 \times 1$ grid.

As shown in Fig. 9(d), the square restores a circular shape within 20 time steps and the final pressure drop is $\Delta p = 0.4001$. One of the instantaneous velocity and pressure fields are also shown in Fig. 9 as well as the equilibrium pressure field. The Young–Laplace law (42) is thus satisfied up to a relative error of 0.025% and the mass loss is less than 0.1% at the 100th time step (Fig. 9).

Second, an ellipsoid defined by semi-axes of length $a = 0.42$, $b = 0.2$, and $c = 0.2$ is initially placed in a cubic cavity discretized with a $25 \times 25 \times 25$ grid. The volume of this ellipsoid is equivalent to the volume of a sphere of radius 0.2561. The observed pressure drop $\Delta p = 0.792$ verifies the Young–Laplace's relation $\Delta p = 2\sigma/R = 0.7808$ up to a relative error of 1.1%. Fig. 10 shows the initial ellipsoidal shape and the final spherical shape of the bubble.

5.3. Bubble rise

In this test case, we consider the rise of a single bubble (initially at rest) due to the effects of buoyancy.

The problem of the rise of buoyant bubbles is often characterized in the literature by the Morton number, the Eotvos number and the Reynolds number:
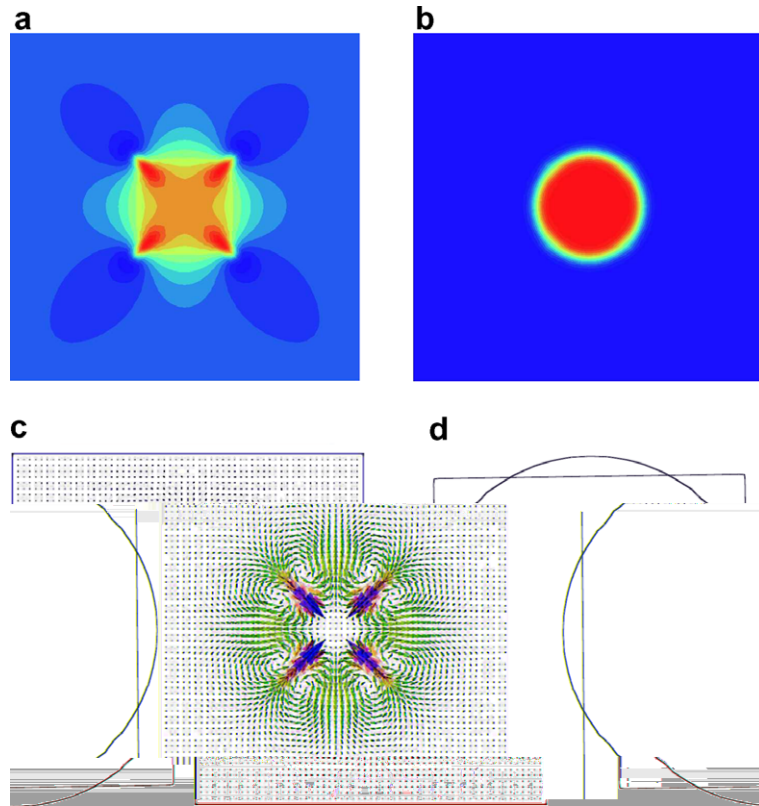
Fig. 9. Relaxation of an initial square bubble in a viscous fluid: (a) pressure at $t = 0.05$ s, (b) pressure at $t = 1$ s, (c) velocity at $t = 0.05$ s, (d) front evolution.
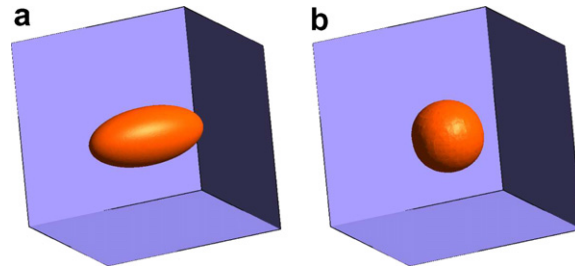

Fig. 10. Relaxation of an initial ellipsoidal bubble in a viscous fluid: (a) initial shape, (b) equilibrium shape.

$$Mo = \frac{g\mu_1^4}{\rho_1\sigma^3}, \quad Eo = \frac{\Delta\rho g d^2}{\sigma}, \quad Re = \frac{\rho_1 U_\infty d}{\mu}. \tag{43}$$

Here ρ_1 and μ_1 denote the viscosity and the density of the outer fluid and U_∞ is the terminal rise velocity of the bubble. The *Mo* and *Eo* number are related to the Reynolds and the Weber number $\left(Re = \left(\frac{Eo^3}{Mo}\right)^{1/4}, \ We = Eo\right)$ if one choose the reference velocity to be equal to $U_R = U_\infty = \sqrt{gd}$, the reference length to be the bubble diameter $L_R = d$ and if $\Delta\rho = \rho 1$ (this is true for air bubbles in water).

Our problem can then characterized by four dimensionless parameters: the density ratio ρ_1/ρ_2, the viscosity ratio μ_1/μ_2, either the Eotvos and Morton numbers or the Reynolds and Weber numbers.
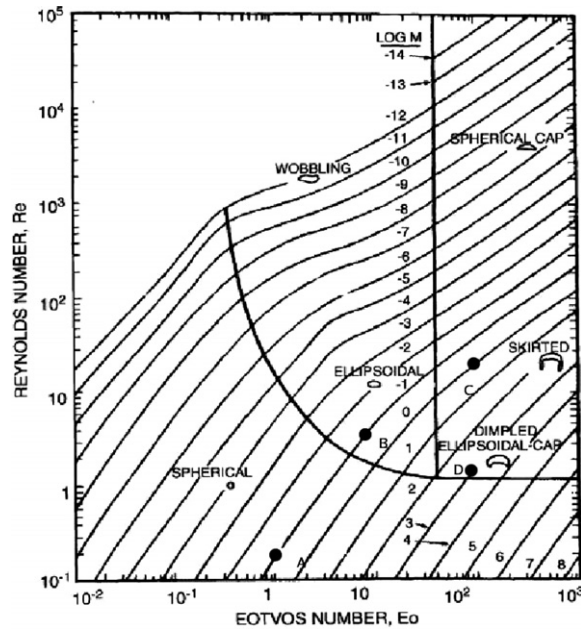
Fig. 11. Bubble diagram of Grace reproduced from [46]: shape regimes for bubbles and drops. The simulated cases are indicated with capitals.
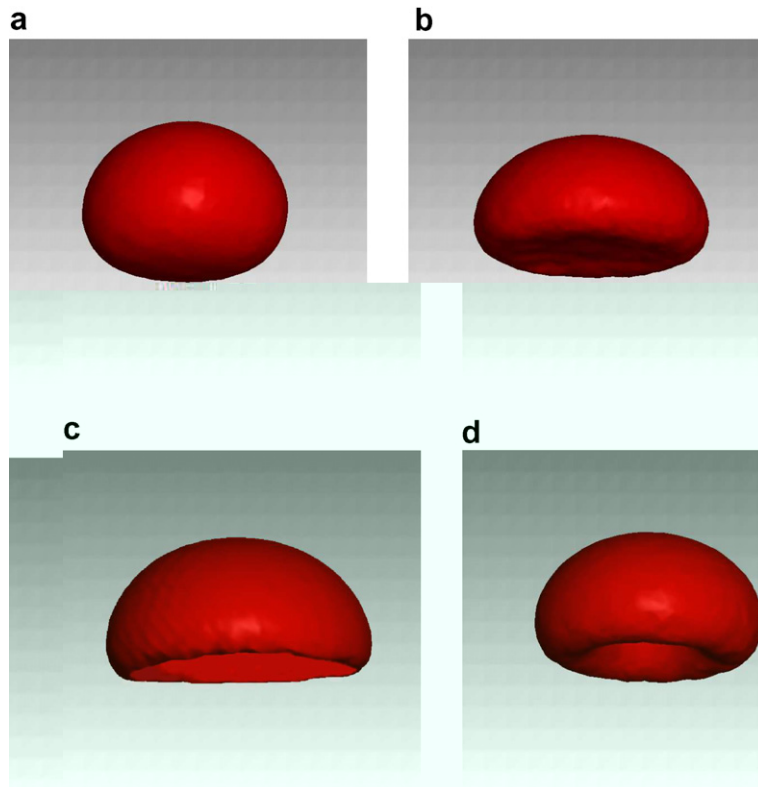

Fig. 12. Single bubble rise: computed bubble shape for the bubble regimes indicated in Table 3 and Fig. 11. The physical parameters are: $\rho_1/\rho_2 = 100$ and $\mu_1/\mu_2 = 100$. (a) Spherical ($Eo = 1$, $M = 10$), (b) ellipsoidal ($Eo = 10$, $Mo = 0.1$), (c) skirted ($Eo = 100$, $Mo = 1$), (d) dimpled ($Eo = 100$, $Mo = 1000$).

For the simulations of single bubbles, a fixed density and viscosity ratio of one thousand was chosen. This density and viscosity ratio mimics gas–liquid systems with sufficient accuracy and is a much higher ratio than the one used in [45]. Prior to computation, a couple of simulation were carried out to assess that the size of the computational domain was sufficiently large to mimic the conditions of an infinite quiescent liquid. It was observed that the bubble shape was not affected anymore if the dimensions of the box in the horizontal directions exceeded 3–4 times the bubble diameter. All the walls of the domain are considered as free slip boundary conditions.

We will compare the computed terminal Reynolds number and shapes of the bubble with data taken from the Bubble diagram of Grace [46] (1973) (Fig. 11). Grace has analyzed a large body of experimental data on shapes in quiescent viscous liquids and has shown that this data can be condensed into one diagram, using the non-dimensional Re, Eo, and Mo numbers.

The bubble has a radius R equal to 0.5 and is placed inside a $3 \times 3 \times 6$ box. The box is resolved by a fairly coarse $30 \times 30 \times 60$ mesh made of tetrahedra. The simulations results for four different regimes given in the bubble diagram of Grace are presented at Fig. 12. In the first case 12(a), for a low Eo number, the bubble remains almost spherical and rises with a constant velocity. In the second case 12(b), the bubble deforms slightly to an ellipsoidal shape and then rises with constant velocity (see Fig. 13). In the third case 12(c), for higher Eo, the bubble deforms to a skirted shape and then rises with constant velocity and shape and finally in the last case 12(d), the bubble has a dimpled ellipsoidal shape.

In Table 3, we observe that despite the coarse mesh, the mass loss is smaller than 0.04% in all of these simulations.

Fig. 14 plots the behavior of the velocity and pressure field for two different bubble regimes. Fig. 14(a) is a velocity vector plot through a vertical center plane for the skirted bubble at Fig. 12(c). Fig. 14(b) shows the pressure profile through a vertical center plane for the ellipsoidal bubble 12(b). As expected the pressure field is
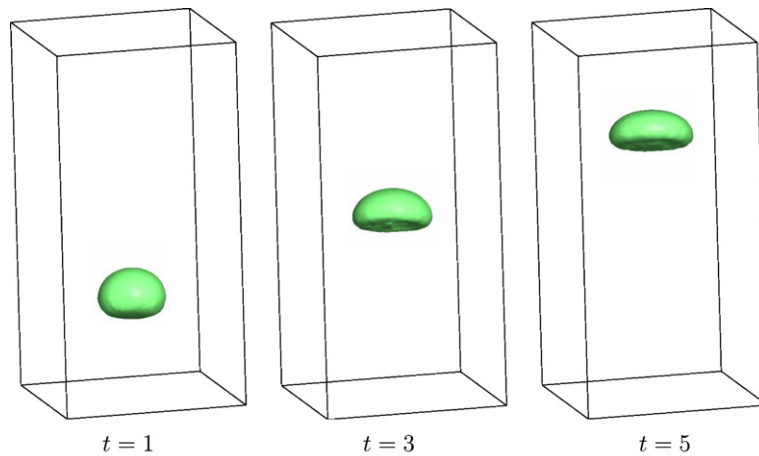

$t = 1$ $t = 3$ $t = 5$

Fig. 13. Single bubble rise with a steady ellipsoidal shape $Eo = 10$, $Mo = 0.1$. Bubble shape at non-dimensional times. The time is non-dimensionalized with the reference velocity $U_R = \sqrt{gd}$ and the reference length $L_R = d$.

Table 3
Simulation of bubble rise in four different regimes

Bubble regime	Case in Fig. 12	Eo	Mo	Re_e	Re_c	Mass loss (%)
Spherical	A	1	10	0.2	0.34	0.004
Ellipsoidal	B	10	0.1	4.6	4.25	0.005
Skirted	C	100	1	20	22.25	0.04
Dimpled	D	100	1000	1.5	1.9	0.005

Comparison of the experimental Reynolds number (Re_e) obtained from the Grace diagram 12 with the computed Reynolds number (Re_c) and mass loss computations.
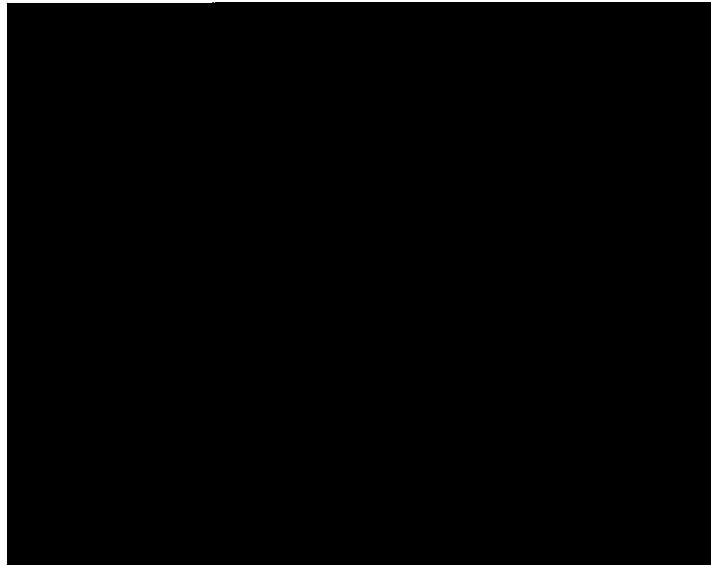
Fig. 14. Velocity and pressure profile through the vertical center plane for two different bubble regimes: (a) dimpled bubble: velocity field, (b) spherical bubble: pressure field.

dominantly hydrostatic while near the bubble, hydrodynamic, and surface tension forces come into play. The bubble jump across the bubble surface due to surface tension is clearly seen.

5.4. Convergence study

We now study the shape convergence of the bubbles computed with our numerical method. We consider the rise of a single 2D bubble and study the shape convergence at the non-dimensional time $t^* = 1$. The time is non-dimensionalized with the reference velocity $U_R = \sqrt{gd}$ and the reference length $L_R = d$.

The non-dimensional parameters for this test case are: $Eo = 100$, $Mo = 0.001$, $\rho_1/\rho_2 = 1000$ and $\mu_1/\mu_2 = 100$. We consider three different grids with resolution 15×30, 30×60 and 60×120 corresponding respectively to a mesh size of $h = 1/15, 1/30, 1/60$.

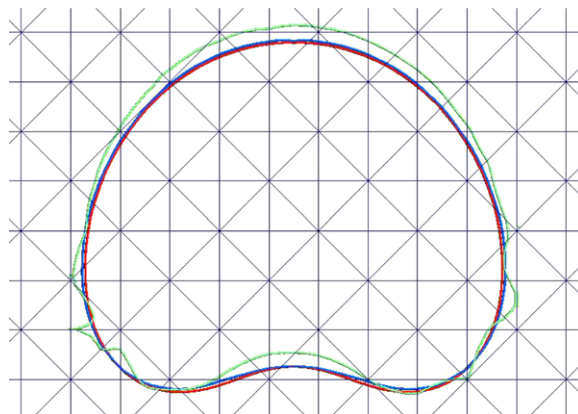Fig. 15 illustrates the shape convergence at non-dimensional time $t^* = 1$.

Fig. 15. Rising bubble. Convergence of the bubble shape with a grid refinement at non-dimensional time $t^* = 1$. $Eo = 100$, $Mo = 0.001$, $\rho_1/\rho_2 = 1000$ and $\mu_1/\mu_2 = 100$. The background grid is the coarser grid $h = 1/15$. The bubble shape is plotted for the three different grids (green $h = 1/15$, blue $h = 1/30$, red $h = 1/60$). (For interpretation of the references in colour in this figure legend, the reader is referred to the Web version of this article.)
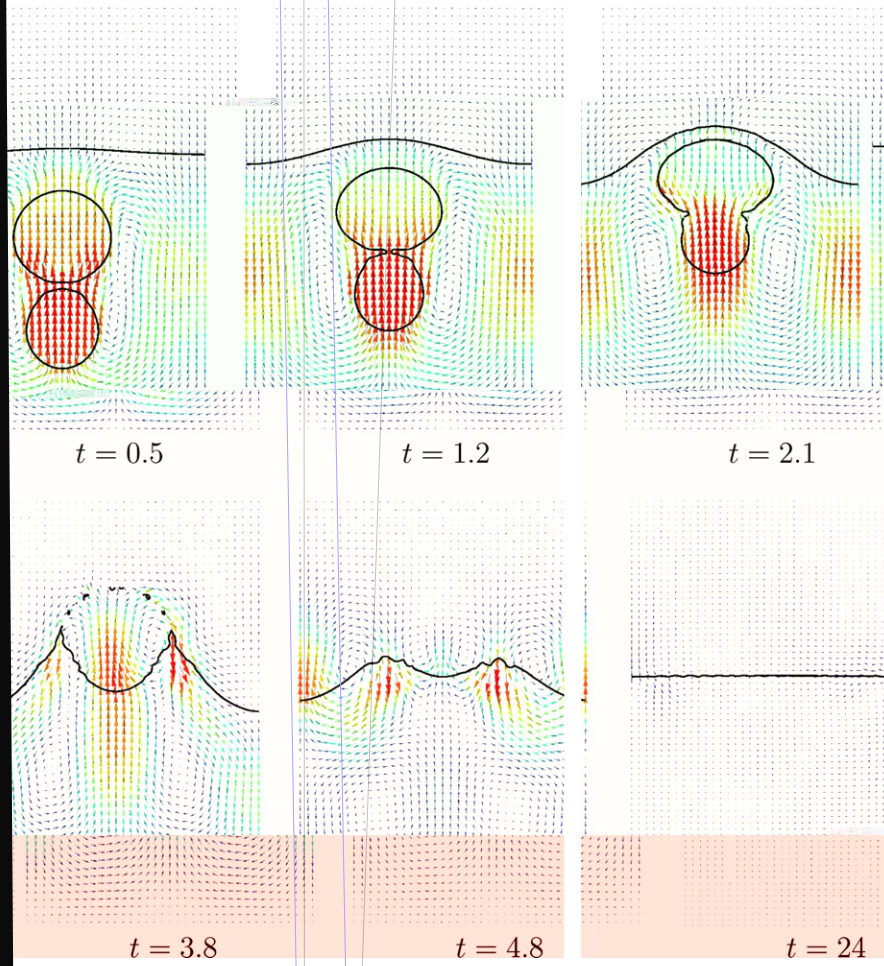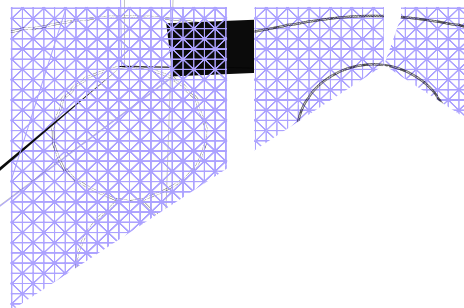
Fig. 16. Merging of two rising bubbles towards a free surface. $Fr = 1$, $Re = 10$, $We = 10$, $\rho_1/\rho_2 = 100$ and $\mu_1/\mu_2 = 2$. Position of the bubbles and velocity fields at different non-dimensional times.

$P1$ elements $P2$ elements

Fig. 17. The exact time of merging depend on the numerical diffusion.

This example demonstrates the convergence of the proposed scheme under mesh refinement and we note that the convergence is achieved on fairly refined grids.

5.5. Bubble merging

We study the rise of two buoyant bubbles in two and three-dimensions. The dynamics of the bubbles depends mainly on the initial distance between them and on the magnitude of the surface tension (Fig. 16). We first consider a rectangular domain $[-1.5, 1.5] \times [0, 6]$ with two circular bubbles inside and a free surface above those (see Fig. 16). The center of the first bubble is $(0, 1)$ and its radius is equal to $R = 0.4$, the center of the second bubble is $(0, 2)$ and its radius is $R = 0.5$. Thus the bubbles have a common axis of symmetry and the initial distance between them equals 1/5 of the largest bubble. The density and viscosity ratio are respectively: $\rho_1/\rho_2 = 100$ and $\mu_1/\mu_2 = 2$ and the Reynolds and Weber number are respectively $Re = 10$, $We = 10$. The
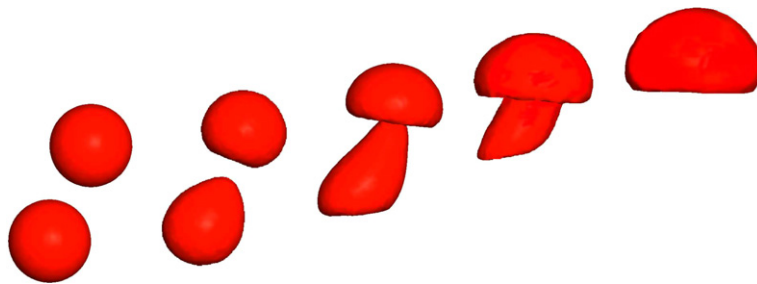

Fig. 18. Transient solution of bubble coalescence when the bubbles are not aligned. $Fr = 1$, $Re = 18.8$, $We = 50$, $\rho_1/\rho_2 = 26$, $\mu_1/\mu_2 = 20$. From left to right the non-dimensional time is $t^* = 0$, $t = 0.5$, $t = 1$, $t = 1.5$, $t = 2$.
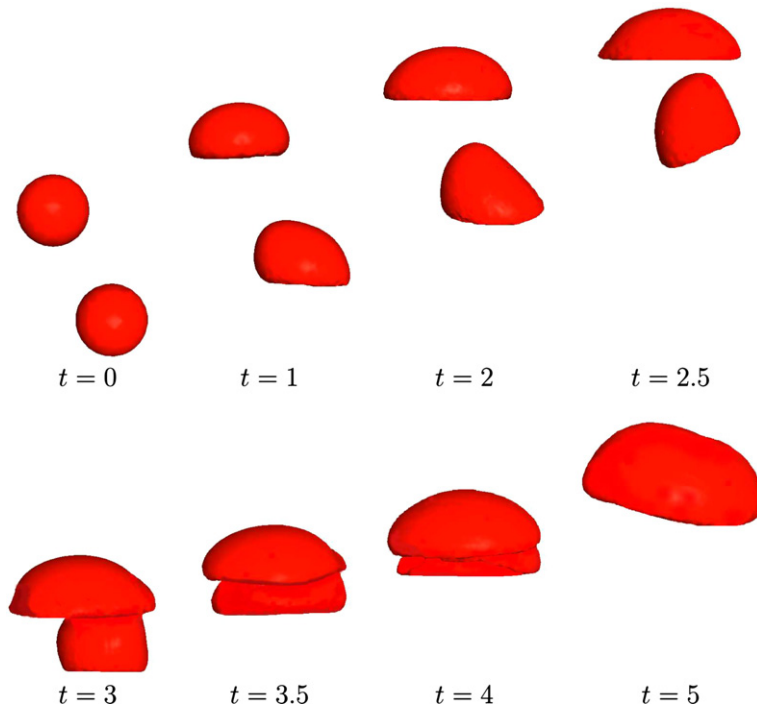

Fig. 19. Transient solution of bubble coalescence when the bubbles are not aligned. $Fr = 1$, $Eo = 16$, $Mo = 0.0002$, $\rho_1/\rho_2 = 100$, $\mu_1/\mu_2 = 100$. Bubble shapes at non-dimensional times.

non-dimensional time step is $\Delta t = 0.01$, which is about 200 times the one used by Tornberg et al. [17] for the same simulation.

Fig. 16 illustrates the merger process. The small bubble travels in the wake of the upper larger bubble and rises faster than that. When the small bubble reaches the larger bubble, the two bubbles merge. In the process, two opposite vortices are created in the wake of the larger bubble. The merged bubble gets more deformed as it gets closer to the surface and it pushes the surface upwards. Finally the filament between the two interfaces gets so thin that the bubble merges with the surface. Subsequently, the two fluid filaments will be smoothened out by local high velocities. Finally we reach a steady state where the two fluids are at rest and separated by a flat surface.

The exact time at which merging will occur in a simulation for a fixed set of physical parameters is however affected by the resolution, and by the amount of diffusion present in the calculations. This amount of numerical dissipation depends upon the order of polynomials p chosen to represent the level set:

$$E_{\text{diss}} = \mathcal{O}(hk)^{2p+2}, \tag{44}$$

where h is the mesh size and k is the dimensionless wave number.

Fig. 17 shows at a fixed non-dimensional time $t = 0.1$ that the bubbles have merged or not according to the use of polynomials of order $p = 1$ or $p = 2$ for the level set.

This is however a process converging as the resolution is increased and the numerical diffusion is decreased.

The second computation involves the oblique coalescence of two gas bubbles in an initially quiet liquid where the bubbles are initially spherical with their centers not aligned on a vertical line. The computational
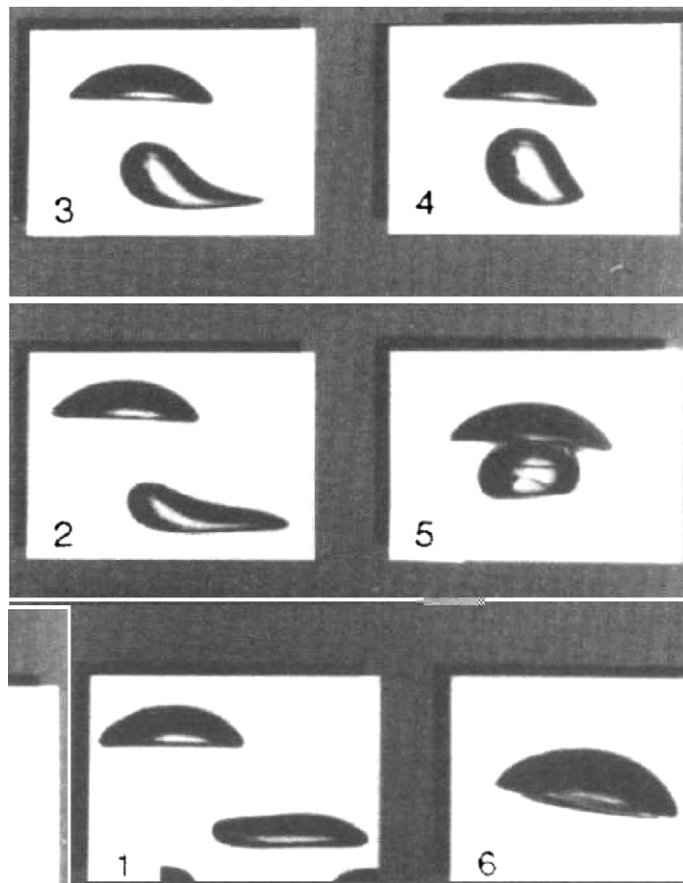

Fig. 20. Experimental photographs by Brereton [50] of the oblique coalescence of bubbles. $Fr = 1$, $Eo = 16$, $Mo = 0.0002$, $\rho_1/\rho_2 = 100$, $\mu_1/\mu_2 = 100$. The non-dimensional time between subsequent photographs is $t^* = 1.2$.

domain is a $[0,4] \times [0,4] \times [0,8]$ box and the two bubbles of radius $R = 0.5$ are located respectively at $(2,2,2.15)$ and $(2.5,2,1)$. We set $Re = 18.8$, $We = 50$, $\rho_1/\rho_2 = 26$, $\mu_1/\mu_2 = 20$. The computational mesh is a $30 \times 30 \times 60$ grid made of tetrahedra.

In Fig. 18, the computed evolution of the bubble shape is shown. The bubbles first become cap-like shaped. Then the lower bubble experiences considerable deformation when it enters the wake region of the upper bubble and they finally merge together because the vortex flow created by the upper bubble accelerates the rising speed of the lower bubble.

Our results compare well with those of Sussman et al. (see [47, p. 330]) that use a much finer refined grid ($64 \times 64 \times 128$ grid). They also compare well with those of Sousa et al. [48] and the experimental results obtained by Narayanan [49].

For the last simulation, we take the same non-dimensional parameters as those used by [45]: $Eo = 16$, $Mo = 0.0002$ ($Re = 67$, $We = 16$), $\rho_1/\rho_2 = 100$, $\mu_1/\mu_2 = 100$. We compare the computed evolution of the bubble shapes with experimental photographs of the observed bubble shapes just before and after the coalescence process [50].

In Figs. 19 and 20, the evolution of the bubble shape is shown together with photographs of the experimentally observed bubble shapes just before and after the coalescence process. Again, we observe that in the beginning the leading bubble behaves like an isolated bubble and that the trailing bubble catches up with the leading bubble and experiences great shape deformation. The shape of the bubbles are in good agreement with the experimental observations of [50] reproduced in Fig. 20 and with the numerical computations of [45,51].

6. Conclusion

In this paper, we have presented a novel numerical method for the simulation of three-dimensional bubble flow problems. The method that has been described extensively in [1,2] combines a quadrature free discontinuous Galerkin method (DGM) for the level set equation with a pressure stabilized finite element method for the Navier Stokes equations. Two main characteristics of our method are:

- fully implicit solution of the Navier Stokes Solver (use of larger time steps);
- high order DGM (usually order $p = 2$) that exhibits very small mass losses, even on fairly coarse grids.

The scope of this paper was the accurate representation of surface tension effects that have to be taken into account when computing bubble dynamics.

In the context of the DGM, we have compared two different numerical methods to compute the curvature. We showed that the use of a curvature computed by means of a direct derivation of the level set function leads to inaccurate and oscillatory results, while a least-squares computation of the curvature filters out the high frequencies, exhibits a quadratic convergence and is more robust.

A number of numerical simulations were used to verify the accuracy of the surface tension approximation method (continuum surface force model). The simulation of a static bubble showed good agreement with the analytical solution indicating that the curvature calculation procedure is accurate. Subsequently, the shapes of single gas bubbles rising in quiescent liquids were compared with data taken from the bubble diagram of Grace. The model was also successfully applied in the case where the interface experiences substantial changes, i.e. co-axial and oblique coalescence of gas bubbles rising in a viscous liquid.

References

[1] E. Marchandise, J.-F. Remacle, N. Chevaugeon, A Quadrature free discontinuous Galerkin method for the level set equation, Journal of Computational Physics 212 (2006) 338–357.

[2] E. Marchandise, J.-F. Remacle, A stabilised finite element method using a discontinuous level set approach for solving two phase incompressible flows, Journal of Computational Physics (2006).

[3] C. Hirt, A. Amsden, J. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, Journal of Computational Physics 135 (2) (1997) 203–216.

[4] T. Hughes, W. Liu, T. Zimmermann, Lagrangian–Eulerian finite element formulation for incompressible viscous flows, Computer Methods in Applied Mechanics and Engineering 29 (1981) 239–349.

[5] T. Tezduyar, M. Behr, J. Liou, A new strategy for finite-element computations involving moving boundaries and interfaces – the deforming–spatial–domain/space–time procedure: I. The concept and the preliminary tests, Computer Methods in Applied Mechanics and Engineering 94 (1992) 339–351.

[6] S. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, Journal of Computational Physics 100 (1) (1992) 25–37.

[7] C. Hirt, B. Nichols, Volume of Fluid Method (VOF) for the dynamics of free boundaries, Journal of Computational Physics 39 (1981) 201–225.

[8] J. Pilliot, E. Puckett, Second order accurate Volume-of-Fluid algorithms for tracking material interfaces, Journal of Computational Physics 199 (2004) 465–502.

[9] S. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, Journal of Computational Physics 79 (1988) 12–49.

[10] J.A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Cambridge University Press, 1999.

[11] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational Physics 114 (1994) 146–159.

[12] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-fluid ows, Computers and Fluids 27 (1998) 663–680.

[13] M. Sussman, E. Fatemi, An efficient interface preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, SIAM Journal on Scientific Computing 20 (1999) 1165–1191.

[14] S. Shin, S. Abdel-Khalib, V. Daru, D. Juric, Accurate representation of surface tension using the level contour reconstruction method, Journal of Computational Physics 203 (2005) 493–516.

[15] J.U. Brackbill, C. Kothe, D.B. Zemach, A continuum method for modeling surface tension, Journal of Computational Physics 100 (2) (1992) 335–354.

[16] Y. Renardy, M. Renardy, Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method, Journal of Computational Physics 183 (2002) 400–421.

[17] A.-K. Tornberg, B. Engquist, A finite flow applications, Computing Visualization in Science 3 (2000) 93–101.

[18] S. Nagrath, K. Jansen, R. Lahey, Computation of incompressible bubble dynamics with a stabilized finite element level set method, Computer Methods in Applied Mechanics and Engineering 194 (2005) 4565–4587.

[19] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, International Journal for Numerical Methods in Fluids 30 (1999) 775–793.

[20] I. Ginzburg, G. Wittum, Two-phase flows on interface refined grids modeled with vof, staggered finite volumes, and spline interpolants, Journal of Computational Physics 166 (2001) 302–335.

[21] S.J. Cummins, M.M. Francois, D.B. Kothe, Estimating curvature from volume fractions, Computers and Structures 83 (2005) 425–434.

[22] T. Hughes, L. Franca, G. Hulbert, A new finite element formulation for fluid dynamics: VIII. The Galerkin/least-squares method for advective–diffusive equations, Computer Methods in Applied Mechanics and Engineering 73 (1989) 173–189.

[23] T. Tezduyar, S. Mittal, S. Ray, R. Shih, Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity–pressure elements, Computer Methods in Applied Mechanics and Engineering 95 (1992) 221–242.

[24] Y.C. Chang, T.Y. Hou, B. Merriman, S. Osher, Eulerian capturing methods based on a level set formulation for incompressible fluid interfaces, Journal of Computational Physics 124 (1996) 449–464.

[25] R. Rannacher, Finite element methods for the incompressible Navier–Stokes equations, Tech. rep., Institute of Applied Mathematics, University of Heidelberg, 1999.

[26] A.-K. Tornberg, Interface Tracking Methods with Applications to Multiphase Flows, Ph.D. thesis, Royal Institute of Technology (KTH), Finland, 2000.

[27] J.-F. Remacle, N. Chevaugeon, E. Marchandise, C. Geuzaine, Efficient visualization of high order finite elements, International Journal for Numerical Methods in Engineering (2006).

[28] D.L. Chopp, Some improvements of the fast marching method, SIAM Journal on Scientific Computing 23 (1) (2001) 230–244.

[29] M. Mount, N.S. Arya, R. Netanyahu, A. Wu, An optimal algorithm for Approximate Nearest Neighbor searching, Journal of the ACM 45 (1998) 891–923.

[30] S. Nagrath, K. Jansen, R. Lahey, Computation of incompressible bubble dynamics with a stabilized finite element level set method, Computer Methods in Applied Mechanics and Engineering 194 (42–44) (2005) 4565–4587.

[31] A. Smolianski, Finite-element level-set operator-splitting (felsos) approach for computing two-fluid unsteady flows with free moving interfaces, International Journal for Numerical Methods in Fluids 48 (2005) 231–269.

[32] T. Hughes, L. Franca, A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska–Brezzi condition: a stable Petrov–Galerkin formulation of the stokes problem accommodating equal-order interpolations, Computer Methods in Applied Mechanics and Engineering 59 (1986) 85–99.

[33] T.J. Barth, Aspects of unstructured grids and finite-volume solvers for the Euler and Navier–Stokes equations, Special Course on Unstructured Grid Methods for Advection Dominated Flows, AGARD R-787, 1992, Chapter 4.

[34] T. Tezduyar, Y. Osawa, Finite element stabilization parameters computed from element matrices and vectors, Computer Methods in Applied Mechanics and Engineering 190 (2000) 411–430.

[35] F. Shakib, T. Hughes, Z. Johan, A new finite element formulation for computational fluid dynamics: the compressible Euler and Navier–Stokes equations, Computer Methods in Applied Mechanics and Engineering 89 (1991) 141–219.

[36] P. Geuzaine, Newton–Krylov strategy for compressible turbulent flows on unstructured meshes, AIAA Journal 39 (2001) 528–531.
[37] X. Cai, C. Farhat, M. Sarkis, A minimum overlap restricted additive Schwarz preconditioner and applications in 3D flow simulations, Contemporary Mathematics 218 (1998) 478–484.
[38] D. Kothe, W. Rider, S. Mosso, J. Brock, J. Hochstein, Volume tracking of interfaces having surface tension in two and three dimensions, Tech. Rep. AIAA 96-0859, AIAA, 1996. URL <citeseer.ist.psu.edu/kothe96volume.html>.
[39] M. Kang, R. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, SIAM Journal on Scientific Computing 15 (2000) 323–360.
[40] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, Journal of Computational Physics 131 (2) (1997) 267–279.
[41] T.J. Barth, P. Frederickson, Higher order solution of the euler equations on unstructured grids using quadratic reconstruction, AIAA Paper 90-0013.
[42] M.M. Francois, S.J. Cummins, E.D. dendy, D.B. Kothe, J. Sicilian, M. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, Journal of Computational Physics 213 (1) (2006) 141–173.
[43] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multiphase flows with surfer, Journal of Computational Physics 113 (1) (1994) 134–147.
[44] S. Shin, D. Juric, Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity, Journal of Computational Physics 180 (2002) 427–470.
[45] M. van Sint Annaland, N. Deen, J. Kuipers, Numerical simulation of gas bubbles behaviour using a three-dimensional volume-of-fluid method, Chemical Engineering Science 60 (2005) 2999–3011.
[46] R. Clift, M. Weber, Bubbles, Drops, and Particles, Academic Press, New York, 1978.
[47] M. Sussman, E. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible Two-Phase flows, Journal of Computational Physics 162 (2000) 301–337.
[48] F.S. de Sousa, N. Mangiavacch, L.G. Nonato, A. Castel, M.F. Tome, V.G. Ferreira, J.A. Cuminato, S. McKee, A front-tracking/front-capturing method for the simulation of 3d multi-fluid flows with free surfaces, Journal of Computational Physics 198 (2004) 469–499.
[49] N.K.S. Narayanan, H.J. Groossens, Coalescence of two bubbles rising in line at low Reynolds number, Chemical Engineering Science 29 (1974) 2071–2082.
[50] G. Brereton, D. Korotney, Coaxial and Oblique Coalescence of Two Rising Bubbles, vol. 119, ASME, New York, 1999.
[51] R. Singh, W. Shyy, Three-dimensional adaptive grid method for multiphase flow computations, in: 43rd AIAA Aerospace Sciences Meeting and Exhibit, 2005.