Conservative and adaptive level-set method for the simulation of two-fluid flows

C Bahbah, M Khalloufi, A Larcher, Y Mesri, T Coupez, R Valette, E Hachem

▶ **To cite this version:**

C Bahbah, M Khalloufi, A Larcher, Y Mesri, T Coupez, et al.. Conservative and adaptive level-set method for the simulation of two-fluid flows. Computers and Fluids, inPress. hal-02154945

HAL Id: hal-02154945
https://hal.science/hal-02154945

Submitted on 13 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Conservative and adaptive level-set method for the simulation of two-fluid flows

C. Bahbah[a,*], M. Khalloufi[a], A. Larcher[a], Y. Mesri[a], T. Coupez[a], R. Valette[a], E. Hachem[a]

[a] *MINES ParisTech, PSL - Research University, CEMEF - Centre for material forming, CNRS UMR 7635, CS 10207 rue Claude Daunesse, 06904 Sophia-Antipolis Cedex, France.*

Abstract

In this paper, we present a high fidelity conservative and adaptive level-set method for the simulation of two-fluid flows. A new level-set method is designed that associates both re-initialization and convection steps in an implicit manner. Thus the new obtained convection-reaction problem is solved using a stabilized finite element method. The accuracy, the time scheme and the mass conservation are thoroughly analyzed. Anisotropic meshing with conservative interpolation is implemented and tested on several benchmarks including splashes, sloshing and complex bubble dynamics.

Keywords: Convective reactive level-set, Scaled heaviside function, Anisotropic multi criteria mesh adaptation, Conservative interpolation

1. Introduction

Two-fluid flows is the term given for two fluids with different properties. Two-fluid flows appear in a wide variety of natural processes and industrial applications such as geophysical flows, water waves, drop impacts, micro-fluidics, bio mechanics and many others. These applications typically involve immiscible fluids that are separated by thin layers known as the interface. The interface is the region across which the fluids properties as well as some of the flow variables are subjected to variations and deformations. It is a sharp front where density and viscosity change abruptly. The challenge in the numerical simulation of two-fluid flows is how to represent the interface and to model its kinematics.

To follow moving interfaces, mesh-based methods still hold a prevalent position [1]. They can be divided into two categories : interface tracking and interface capturing depending on whether the interface is explicitly described or defined implicitly by an auxiliary function. For instance, front tracking methods rely on considering marker particles to explicitly keep track of the interface [2–8]. These methods provide very accurate interface evolution; however changes in topology need to be handled explicitly as well, which adds to the complexity of those methods. On the contrary, implicit representation treats changes in topology in a straightforward manner. In the context of two-fluid flows, the volume of fluid (VOF) [9–14] method allows mass conservation by construction, but its discontinuous description of the interface makes the computation of the curvature and the normal to the interface difficult. Although the level-set method [15–18] does not have the same conservation properties as the volume of fluid method, it offers many advantages. To start, the implicit representation of the interface allows to accurately compute changes in topology when breakup or coalescence occur. Moreover, since we do not need to explicitly reconstruct the interface, the method is suitable when a calculation of geometric features such as normal or curvature is needed. However, the

* Corresponding author, Tel. +33 (0) 4 93 67 89 39

Email addresses: chahrazade.bahbah@mines-paristech.fr (C. Bahbah), mehdi.khalloufi@mines-paristech.fr (M. Khalloufi), youssef.mesri@mines-paristech.fr (Y. Mesri), thierry.coupez@mines-paristech.fr (T. Coupez), rudy.valette@mines-paristech.fr (R. Valette), elie.hachem@mines-paristech.fr (E. Hachem)

biggest disadvantage of this method is the fact that due to numerical dissipation in simulations, the mass of each phase is not conserved which is crucial when modeling realistic two-fluid flows with high density ratio.

To address this issue, many approaches have been introduced. For instance, Sussman and al. [19, 20] proposed to combine the level-set method with a VOF method in order to conserve mass when the interface is convected. Russo and Smereka [21] completed this work by introducing a subcell-fix that consists in using the location of the interface to compute the fluxes with second-order accuracy for cells containing the interface. This approach was later extended to fourth-order accuracy by [22]. In a different approach, the use of massless particles to complement the level-set representation was introduced in [23, 24]: it is called hybrid particle level-set method and consists in correcting the level-set function when the particles cross the interface. Others have taken advantage of the fact that the mass conservation error reduces when using adaptive mesh refinement, either by solving the level-set equations on an auxiliary high-resolution equidistant cartesian grid [25] or refining the mesh in the regions close to the interface [26, 27]. On the other hand, other approaches aimed at improving level-set approaches by working on the re-initialization step. Indeed, in many cases, an additional equation is added and solved to keep the property of the distance function,[28, 29]. Up to now, many discussions remain concerning the frequency for solving this equation and its applicability in fast dynamics.

In this paper, we design a new level-set method that associates both re-initialization and convection steps in an implicit manner. Thus the new obtained convection-reaction problem is solved using an adequate stabilized finite element method. Finally, a global conservative method for anisotropic mesh adaptation is introduced. It combines an a posteriori error estimator to minimize the interpolation error of the finite element solution followed by an interpolation with restrictions method that conserves physical properties.

The paper is outlined as follows: we start in section 2 with a description of the new definition of the level-set method. Then, we present the mesh adaptation technique with details on the edge-based error estimation. In section 4, the multicriteria conservative interpolation method is presented. In section 5, we introduce the VMS incompressible Navier-Stokes solver. Section 6 provides the numerical test cases showing the efficiency and the accuracy of the proposed method. Finally, section 7 is dedicated to the conclusion and future work.

2. Convective reactive level-set

Level-set methods were first developed and used in the context of computer graphics and image restoration [30] and were extended to the case of two-fluid flows by Sussman and al [18]. In this work, we introduce a new convected-reactive level-set method that combines both convection and re-initialization steps.

First, let us introduce the traditional level-set function as a signed distance function that enables the localization of the interface between two fluids. Let us define Ω the whole computational domain, Ω_l the liquid domain and Ω_g the gas domain and let us denote by $\Gamma = \Omega_l \cap \Omega_g$ the interface between them. The level-set function represents the shortest distance to the interface Γ, it is defined at each node x of the domain Ω as follows:

$$\alpha(x) = \begin{cases} -\operatorname{dist}(x,\Gamma) & \text{if} \quad x \in \Omega_l \\ 0 & \text{if} \quad x \in \Gamma \\ \operatorname{dist}(x,\Gamma) & \text{if} \quad x \in \Omega_g \end{cases} \tag{1}$$

To follow the evolution of the level-set function, an advection problem is solved :

$$\frac{\partial \alpha}{\partial t} + v \cdot \nabla \alpha = 0 \tag{2}$$

with v, the velocity coming from the Navier-Stokes equations. The level-set as a distance function verifies $\|\nabla \alpha\| = 1$. However, when the interface is convected by a given velocity, it loses its distance property and needs to be re-initialized to recover it, this is done solving an Hamilton-Jacobi problem [18] :

$$\frac{\partial \alpha}{\partial \tau} + s(\alpha)(\|\nabla \alpha\| - 1) = 0 \tag{3}$$

with τ a fictitious "time step" and $s(\alpha)$ the sign function of α. The first step towards a more conservative level-set method is to filter the level-set function close to the interface, thus reducing computational cost and ensuring mass conservation. Let us denote by $\tilde{\alpha}$, the new filtered level-set function :

$$\tilde{\alpha} = \frac{1}{1 + e^{-\frac{\alpha}{\varepsilon}}} \tag{4}$$

with ε the thickness of the truncation. We compute the derivative of the new filtered level-set function $\tilde{\alpha}$ with respect to the distance function :

$$\nabla\tilde{\alpha} = \nabla\alpha \frac{e^{-\frac{\alpha}{\varepsilon}}}{\varepsilon(1 + e^{-\frac{\alpha}{\varepsilon}})^2} \tag{5}$$

The truncated level-set function now verifies the following property :

$$\|\nabla\alpha\| = 1 \longrightarrow \|\nabla\tilde{\alpha}\| = \frac{1}{\varepsilon}(1 - \tilde{\alpha})\tilde{\alpha} \tag{6}$$

This new restriction must be taken into account in the re-initialization equation :

$$\frac{\partial\alpha}{\partial\tau} = s(\alpha)(1 - \|\nabla\alpha\|) \longrightarrow \frac{\partial\tilde{\alpha}}{\partial\tau} = s(\tilde{\alpha})\left(\frac{1}{\varepsilon}(1 - \tilde{\alpha})\tilde{\alpha} - \|\nabla\tilde{\alpha}\|\right) \tag{7}$$

Let us introduce by U, the re-initialization velocity. The re-initialization equation now reads :

$$U = s(\tilde{\alpha})\frac{\nabla\tilde{\alpha}}{\|\nabla\tilde{\alpha}\|} \longrightarrow \frac{\partial\tilde{\alpha}}{\partial\tau} + U \cdot \nabla\tilde{\alpha} = s(\tilde{\alpha})\frac{1}{\varepsilon}(1 - \tilde{\alpha})\tilde{\alpha} \tag{8}$$

Therefore, the final one-equation model is obtained by including the re-initialization step as an additional convection term, and written as follows:

$$\frac{\partial\tilde{\alpha}}{\partial t} + (v + \lambda U) \cdot \nabla\tilde{\alpha} = s(\tilde{\alpha})\frac{\lambda}{\varepsilon}(1 - \tilde{\alpha})\tilde{\alpha} \tag{9}$$

where λ is constant proportional to a velocity. We apply then an implicit temporal discretization, the one-equation model becomes:

$$\frac{3\tilde{\alpha}^{n+1} - 4\tilde{\alpha}^n + \tilde{\alpha}^{n-1}}{2\Delta t} + (v^{n+1} + \lambda U^n) \cdot \nabla\tilde{\alpha}^{n+1} - s(\tilde{\alpha})\frac{\lambda}{\varepsilon}(1 - \tilde{\alpha}^n)\tilde{\alpha}^{n+1} = 0 \tag{10}$$

Finally, we introduce a stabilized finite element variational formulation to deal with the new obtained convection-reaction equation (10):

$$\begin{aligned}\left(\frac{3\tilde{\alpha}^{n+1} - 4\tilde{\alpha}^n + \tilde{\alpha}^{n-1}}{2\Delta t}, \omega_h\right)_\Omega + ([v^{n+1} + \lambda U^n] \cdot \nabla\tilde{\alpha}^{n+1}, \omega_h)_\Omega - \left(s(\tilde{\alpha})\frac{\lambda}{\varepsilon}(1 - \tilde{\alpha}^n)\tilde{\alpha}^{n+1}, \omega_h\right)_\Omega \\ + \sum_K (\mathcal{R}(\tilde{\alpha}^{n+1}), \tau^n[v^{n+1} + \lambda U^n] \cdot \nabla\omega_h)_K + \sum_K (\mathcal{R}(\tilde{\alpha}^{n+1}), \tau^n \mid s(\tilde{\alpha})\frac{\lambda}{\varepsilon}(1 - \tilde{\alpha}^n)\tilde{\alpha}^{n+1} \mid \cdot\omega_h)_K = 0,\end{aligned} \tag{11}$$

with $\mathcal{R}$, the residual coming from equation (10). To distribute in space the respective material properties such as the density ρ and the dynamic viscosity μ, we refer to the following mixing laws:

$$\rho = \rho_g \mathcal{H}_{scaled}(\tilde{\alpha}) + \rho_l(1 - \mathcal{H}_{scaled}(\tilde{\alpha})) \tag{12}$$

$$\mu = \mu_g \mathcal{H}_{scaled}(\tilde{\alpha}) + \mu_l(1 - \mathcal{H}_{scaled}(\tilde{\alpha})) \tag{13}$$

$$\tag{14}$$

where $\mathcal{H}_{scaled}$ is a non-symmetrical smoothed Heaviside function that is used generally to reduce spurious oscillations based on density scaling as shown in [31]:

$$\mathcal{H}_{scaled}(\tilde{\alpha}) = \begin{cases} 0 & \text{if} \quad \tilde{\alpha} < -\varepsilon \\ \frac{1}{2}\left(\frac{1}{2} + \frac{\tilde{\alpha}}{\varepsilon} + \frac{\tilde{\alpha}^2}{2\varepsilon^2} - \frac{1}{4\pi^2}\left(\cos\left(\frac{2\pi\tilde{\alpha}}{\varepsilon}\right) - 1\right) + \frac{\varepsilon + \tilde{\alpha}}{\varepsilon\pi}\sin\left(\frac{\pi\tilde{\alpha}}{\varepsilon}\right)\right) & \text{if} \quad \tilde{\alpha} \leq \varepsilon \\ 1 & \text{if} \quad \tilde{\alpha} > \varepsilon \end{cases} \tag{15}$$

3. Navier-Stokes

Flow motion of incompressible fluids is described by the Navier-Stokes equations given by :

$$\begin{cases} \rho\,(\partial_t v + v\cdot\nabla v) - \nabla\cdot\sigma = f \\ \nabla\cdot v = 0 \end{cases} \tag{16}$$

where $t\in[0,T]$ is the time, $v(x,t)$ the velocity, $p(x,t)$ the pressure and ρ the density. The Cauchy stress tensor for a Newtonian fluid is given by:

$$\sigma = 2\mu\ \epsilon(v) - p\ I_d, \tag{17}$$

with I_d the d-dimensional identity tensor, μ the dynamic viscosity and ϵ the strain-rate tensor defined as :

$$\epsilon(v) = \frac{1}{2}(\nabla v + \nabla^T v), \tag{18}$$

The stabilized finite element scheme for the Navier-Stokes equations is derived from a variational multiscale point of view. Both the velocity and the pressure spaces are enriched which cures the spurious oscillations in the convection-dominated regime and deals with the pressure instability. Particular attention is paid to the determination of the stabilization parameters in the presence of mesh anisotropy, through the use of a directional element size.

To fix notation, let $\Omega\subset\mathbb{R}^d$ be the fluid domain, where d is the space dimension, and $\partial\Omega$ its boundary. We briefly described the main steps to derive this formulation. The basic idea is to consider that the velocity and the pressure can be split into two components, a coarse one and a fine one, corresponding to different scales or levels of resolution. First, we solve the fine scales in an approximate manner and then we replace their effect into the large-scale equation. We present here only an outline of the method, and the reader is referred to [32] for extensive details about the formulation.

Let us split the velocity and the pressure fields into resolvable coarse-scale and unresolved fine-scale components: $v = v_h + v'$ and $p = p_h + p'$. The same decomposition can be applied to the weighting functions: $w = w_h + w'$ and $q = q_h + q'$. Subscript h is used hereafter to denote the finite element (coarse) component, whereas the prime is used for the so called scale (fine) component of the unknowns. The enrichment of the functional spaces is performed as follows: $V = V_h \oplus V'$, $V_0 = V_{h,0}\oplus V_0'$ and $Q = Q_h\oplus Q'$. Thus, the finite element approximation for the time-dependent Navier-Stokes problem reads:

$$\begin{cases} \text{Find}(v,p)\in V\times Q \text{ such that:} \\ \rho\left(\partial_t(v_h+v'),(w_h+w')\right) + \rho\left((v_h+v')\cdot\nabla(v_h+v'),(w_h+w')\right) \\ \qquad\qquad + \left(2\mu\varepsilon(v_h+v'):\varepsilon(w_h+w')\right) \\ \qquad\qquad - \left((p_h+p'),\nabla\cdot(w_h+w')\right) = \left(f,(w_h+w')\right), \quad \forall w\in V_0 \\ \left(\nabla\cdot(v_h+v'),(q_h+q')\right)_\Omega = 0, \quad \forall q\in Q. \end{cases} \tag{19}$$

To derive the stabilized formulation, we split Equations (19) into a large-scale and a fine-scale problem. The fine-scale problem is defined on element interiors. Under several assumptions about the time-dependency and the non-linearity of the momentum equation of the subscale system detailed in [32], the fine-scale solutions v' and p' written in terms of the time-dependent large-scale variables using residual-based terms that are derived consistently. Consequently, we can use static condensation, that consists in substituting directly v' and p' into the large-scale problem. This gives rise to additional terms in the Finite Element formulation, that are tuned by a local stabilizing parameter. These terms are responsible for the enhanced stability compared to the standard Galerkin formulation. The large-scale system finally reads:

$$\begin{cases} \rho\left(\partial_t v_h, w_h\right)_\Omega + \left(\rho v_h\cdot\nabla v_h, w_h\right)_\Omega \\ \qquad - \sum_{K\in\mathcal{T}_h}\left(\tau_1\mathcal{R}_{\text{M}}, \rho v_h\nabla w_h\right)_K + \left(2\mu\varepsilon(v_h):\varepsilon(w_h)\right)_\Omega \\ \qquad - \left(p_h,\nabla\cdot w_h\right)_\Omega - \sum_{K\in\mathcal{T}_h}\left(\tau_2\mathcal{R}_{\text{C}},\nabla\cdot w_h\right)_K = \left(f,w_h\right)_\Omega, \quad \forall w_h\in V_{h,0} \\ \left(\nabla\cdot v_h, q_h\right)_\Omega - \sum_{K\in\mathcal{T}_h}\left(\tau_1\mathcal{R}_{\text{M}},\nabla q_h\right)_K = 0, \quad \forall q_h\in Q_h \end{cases} \tag{20}$$

where $(\cdot,\cdot)_\Omega$ represents the scalar product on the whole domain omega while $(\cdot,\cdot)_K$ is the scalar product on Element K. The quantities τ_1 and τ_2 are stabilization parameters defined hereafter. The momentum residual $\mathcal{R}_{\text{M}}$ and the continuity residual $\mathcal{R}_{\text{C}}$ are expressed as:

$$\begin{aligned} \mathcal{R}_{\text{M}} &= f - \rho\partial_t v_h - \rho v_h \cdot \nabla v_h - \nabla p_h \\ \mathcal{R}_{\text{C}} &= -\nabla \cdot v_h \end{aligned} \tag{21}$$

Compared to the standard Galerkin method, the proposed stable formulation involves additional integrals that are evaluated element-wise. These additional terms represent the stabilizing effect of the sub-grid scales and are introduced in a consistent way in the Galerkin formulation. They make it possible to avoid instabilities caused by both dominant convection terms and incompatible approximation spaces. All of these terms are controlled by the stabilization parameters τ_1 and τ_2, for which we adopt the definition proposed in [33]:

$$\tau_1 = \left[\left(\frac{2\rho \|v_h\|_K}{h_K} \right)^2 + \left(\frac{4\mu}{h_K^2} \right)^2 \right]^{-\frac{1}{2}}, \tag{22}$$

$$\tau_2 = \left[\left(\frac{\mu}{\rho} \right)^2 + \left(\frac{c_2 \|v_h\|_K}{c_1 h_K} \right)^2 \right]^{\frac{1}{2}} \tag{23}$$

where h_K is the characteristic length of the element and c_1 and c_2 are algorithmic constants. We take them as $c_1 = 4$ and $c_2 = 2$ for linear elements [33].

Equations (21) are discretized in time by a semi-implicit scheme. The convective term, the viscous term and the pressure term in the momentum equation, as well the divergence term in the continuity equation, are integrated implicitly through a backward Euler scheme. All other contributions (i.e. the source term and the stabilization terms) are integrated explicitly by a forward Euler scheme.

4. Anisotropic mesh adaptation for two-fluid flows

Anisotropic mesh adaptation is a powerful tool to improve the quality and the efficiency of finite elements methods. It provides a way of controlling the accuracy of the numerical solution by modifying the domain discretization according to the size and directional constraint, in that sense discontinuities or gradients of the solution are highly directional and can be captured with a good accuracy using anisotropic mesh adaptation. The algorithm is built in order to compute a mesh and a numerical solution. At each stage, a numerical solution is computed on the current mesh and an estimation of the interpolation error is evaluated. An edge based error estimator combined to a gradient recovery procedure is defined. Once the optimal metric has been obtained, the mesh generation and adaptation procedure based on a topological representation described in [34], can be used to generate a new mesh.

4.1. Edge-based error estimation

Let u_h be a P1 finite element approximation obtained by applying the Lagrange interpolation operator to a regular function $u \in C^2(\Omega)$. At each vertex i of the mesh, we have $U^i = u(x^i) = u_h(x^i)$ (where x^i are the coordinates of the vertex i). Let $\Gamma(i)$ be the "patch" associated to a vertex x^i of the mesh defined as the set of nodes which share one edge with x^i , and let us denote by x^{ij} the edge connecting x^i to x^j as in figure 1.

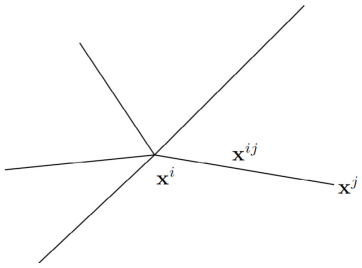


Figure 1: Patch associated with node x^i

The gradient $\nabla u_h \cdot x^{ij}$ on the edge x_{ij} is continuous, therefore we can write :

$$U^j = U^i + \nabla u^h \cdot x^{ij} \tag{24}$$

which leads to :

$$\nabla u_h \cdot x^{ij} = U^j - U^i \tag{25}$$

In the work of [34], an a posteriori error estimate based on the length distribution tensor approach and the associated edge based error analysis is defined :

$$|| \nabla u^h \cdot x^{ij} - \nabla u(x^i) \cdot x^{ij} || \leq \max_{y \in |x^i, x^j|} | x^{ij} \cdot H_u(y) \cdot x^{ij} | \tag{26}$$

with H_u the hessian of u. We want to compute the recovered gradient g^i of u_h at the node x^i :

$$\nabla g_h \cdot x^{ij} = g^j - g^i \tag{27}$$

The projection of the Hessian based on the gradient at the extremities of the edge is obtained as follows :

$$(\nabla g_h \cdot x^{ij}) \cdot x^{ij} = (g^j - g^i) \cdot x^{ij} \tag{28}$$

$$(H_u \cdot x^{ij}) \cdot x^{ij} = g^{ij} \cdot x^{ij} \tag{29}$$

with $g^{ij} = g^j - g^i$. We denote the error along the edges using the following expression :

$$e^{ij} = | g^{ij} \cdot x^{ij} | \tag{30}$$

This error sampling is the exact interpolation error along the edge and enables us to evaluate the global L_1 error. The equation (30) can be evaluated only when the gradient of u is known and continuous at the nodes of the mesh, thus a recovery procedure has to be considered.

4.2. Gradient recovery procedure

The recovery gradient operator is defined by a local optimization problem :

$$G^i = arg \min_{G} \left(\sum_{j \in \Gamma(i)} | (G - \nabla u_h) \cdot x^{ij} |^2 \right) \tag{31}$$

Denoting by $\otimes$, the tensor product between two vectors, let us introduce X^i the length distribution tensor at node i :

$$X^i = \frac{1}{| \Gamma(i) |} \left(\sum_{j \in \Gamma(i)} x^{ij} \otimes x^{ij} \right) \tag{32}$$

whose purpose is to give an average representation of the distribution of edges in the patch. Let us express the recovered gradient G^i in terms of the length distribution tensor :

$$G^i = (X^i)^{-1} \sum_{j \in \Gamma(i)} U^{ij} x^{ij}, \tag{33}$$

with $U^{ij} = U^j - U^i$. Now, the estimated error can be written as :

$$e_{ij} = G^{ij} \cdot x^{ij} \tag{34}$$

4.3. Metric construction

It is necessary to take into account the neighborhood of the node so that the best averaging representation is a metric defined at each node [35]. The metric takes the following expression :

$$\tilde{M^i} = (\tilde{X^i})^{-1} \tag{35}$$

where

$$\tilde{X^i} = \frac{1}{\mid \Gamma(i) \mid} \left(\sum_{j \in \Gamma(i)} s^{ij} \otimes s^{ij} \right) \tag{36}$$

The stretching factor s^{ij} of the edge ij is chosen so that the total number of nodes in the mesh is kept fixed (cite) :

$$s^{ij} = \left(\frac{e_{ij}}{e(N)} \right) \tag{37}$$

with $e(N)$ given in [36].

4.4. Mesh Adaptation Criterion

In two-fluid flows applications, the material interface between the different fluids needs to be modeled accurately. The common way to adapt a mesh to several variables, such as the velocity and the level-set function, is to compute the metrics corresponding to each of them and then to produce a unique metric by an operation known as the intersection of metrics. In this work, we simplify this operation and we use one metric that accounts for different variables. Therefore, based on the theory proposed in the previous section, it is possible to extend the definition to account for several sources of error [36]. In this work, the adaptivity accounts for the velocity, its magnitude and also the level-set function by defining the following vector of sources of error :

$$v(x^i) = \left\{ \frac{V^i}{\mid V^i \mid}, \frac{\mid V^i \mid}{\max_j \mid V^j \mid}, \frac{\tilde{\alpha}}{\max(\tilde{\alpha})} \right\} \tag{38}$$

Because all fields are normalized (the velocity components v_x, v_y and v_z by the local velocity norm, the velocity norm and the level-set function by their respective global maximum), a field that is much larger in absolute value does not dominate the error estimator, and the variations of all variables are fairly taken into account.

5. Multicriteria conservative interpolation

The method implemented in this work is based on the on the work of [37]. The main idea is to interpolate the solutions on the newly adapted mesh using linear interpolation and to impose the conservation of global physical quantities. Here, the method is extended for multicriteria conservation. Let us consider a donor mesh $\mathcal{T}_1$ where a discrete solution u_1 is computed, and a target mesh $\mathcal{T}_2$ where a solution u_2 is computed by a linear interpolation of u_1. To do so, we use tree search algorithms in order to locate the position of the nodes of the target mesh $\mathcal{T}_2$ in the donor mesh $\mathcal{T}_1$.

5.1. Conservation

We apply a series of constraints to the interpolated field u_2 through the Lagrange multipliers. For that, $\tilde{u}_2$ must satisfy the following conditions :

- $\tilde{u}_2$ must be the nearest to u_2 in the L_2-norm
- $\tilde{u}_2$ must respect some physical properties of u_1, solution on the donor mesh

We are interested here in ensuring conservation of linear momentum and mass, to do so we must apply the following constraints :

$$\begin{aligned} \text{Minimize :} \quad & \int_{\Omega_2} \mid \tilde{u}_2 - u_2 \mid^2 \\ \text{Under the constraints :} \quad & \int_{\Omega_2} \tilde{u}_2 = \int_{\Omega_1} u_1 \\ & \int_{\Omega_2} \nabla \cdot \tilde{u}_2 = \int_{\Omega_1} \nabla \cdot u_1 \end{aligned}$$

We express the solutions in terms of the shape functions of each finite element partitions :

$$u_n = \sum_i N_n^i U_n^i, \quad n = 1, 2$$

We introduce the Lagrange multipliers of the constraints. The Lagrangian is :

$$\mathcal{L}\left(\tilde{U}_2, \lambda\right) = \int_{\Omega_2} \mid N_2\tilde{U}_2 - N_2U_2 \mid^2 - \lambda_1 \left(R_{2,1}^t\tilde{U}_2 - R_{1,1}^tU_1\right) - \lambda_2 \left(R_{2,2}^t\tilde{U}_2 - R_{1,2}^tU_1\right) \tag{39}$$

With :

$$R_{1,1} = \int_{\Omega_1} N_1 \tag{40}$$

$$R_{2,1} = \int_{\Omega_1} N_2 \tag{41}$$

$$R_{1,2} = \int_{\Omega_1} \partial_j N_1 \quad j = 1, ..., d \tag{42}$$

$$R_{2,2} = \int_{\Omega_1} \partial_j N_2 \quad j = 1, ..., d \tag{43}$$

To find the optimal point of the Lagrangian, we solve the following system :

$$\begin{bmatrix} M_2 & -R_{2,1} & -R_{2,2} \\ R_{2,1}^t & 0 & 0 \\ R_{2,2}^t & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{U}_2 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} M_2\tilde{U}_2 \\ R_{1,1}^tU_1 \\ R_{1,2}^tU_1 \end{bmatrix} \tag{44}$$

with $M_2 = \int_{\Omega_2} N_2^t N_2$ the mass matrix associated with the target mesh, $R_1^tU_1$ and $R_2^tU_2$ are scalars that represent the global quantities one wishes to conserve (here, linear momentum and mass). Instead of solving the system using iterative methods, which is very costly, the mass matrix M_2 is approximated by a diagonal matrix which reduces considerably the computational cost.

To sum up, for each stage, we compute a numerical solution u_1 on a first mesh $\mathcal{T}_1$. We adapt the mesh with respect to the solution and we obtain a new mesh $\mathcal{T}_2$ and an interpolated solution u_2. Adapting the mesh, will allow minimizing the interpolation error as explained above. The interpolated solution is corrected using the conservative algorithm, we note the new solution $\tilde{u}_2$. In the context of dynamic mesh adaptation, this procedure is repeated till convergence is reached.

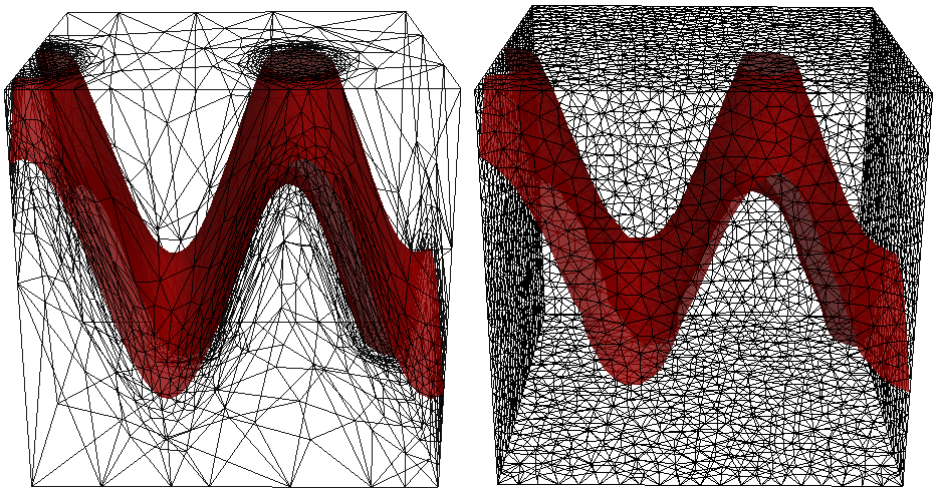


Figure 2: Left : 3D Anisotropic mesh - Right : 3D uniform mesh

	Linear interpolation	**Conservative interpolation**
$\mathcal{T}_1 \rightarrow \mathcal{T}_2 \rightarrow \mathcal{T}_1$	0.0932863	0.0776079
$2\times\mathcal{T}_1 \rightarrow \mathcal{T}_2 \rightarrow \mathcal{T}_1$	0.125586	0.093279
$3\times\mathcal{T}_1 \rightarrow \mathcal{T}_2 \rightarrow \mathcal{T}_1$	0.156789	0.109485
$4\times\mathcal{T}_1 \rightarrow \mathcal{T}_2 \rightarrow \mathcal{T}_1$	0.18643	0.125575
$5\times\mathcal{T}_1 \rightarrow \mathcal{T}_2 \rightarrow \mathcal{T}_1$	0.214496	0.141359

Table 1: Error for the transfer from anisotropic to isotropic

6. Numerical examples

In order to validate the proposed conservative numerical framework, four time-dependent numerical test cases will be presented in this section. The results obtained with the proposed approach are compared with either analytic solutions or with those obtained by other approaches that can be found in the literature.

6.1. 3D analytical test case

To test the effectiveness of our numerical framework, we propose to start in this section with a 3D analytical test case. To this end, we define the following perturbation in a cubic domain $[-0.5, 0.5]^3$ as shown in figure 2:

$$f(x, y, z) = tanh(20(x + 0.3sin(-10y) - 0.3sin(-5(z - 0.1)))) \tag{45}$$

We want to emphasize a complex 3D situation using (i) anisotropic meshes with stretched elements and (ii) dynamic mesh adaptation. Therefore, we interpolate the level set function back and forth from one initial anisotropic mesh $\mathcal{T}_1$ of 31799 elements to an isotropic mesh $\mathcal{T}_2$ of 29414 tetrahedrons (see figure 2).

Interpolating back and forth from one mesh to another allows us to analyze the accumulated error after each interpolation step. This is crucial for mesh adaptation where a large number of interpolation steps are needed. To this end, the transfer from $\mathcal{T}_1^i \rightarrow \mathcal{T}_2^i \rightarrow \mathcal{T}_1^i$ is performed 5 times leading to a total of 10 interpolation steps. To this end, we analyzed the error when the solution is re-interpolated back to the donor mesh by computing the gap in the L_2-norm between u_1^i and u_2^i. Table 1 gives details on the error for each number of interpolation. As expected, the error remains fairly larger when using the linear interpolation. Note that similar conclusions were underlined in the literature, mainly in [37, 38].

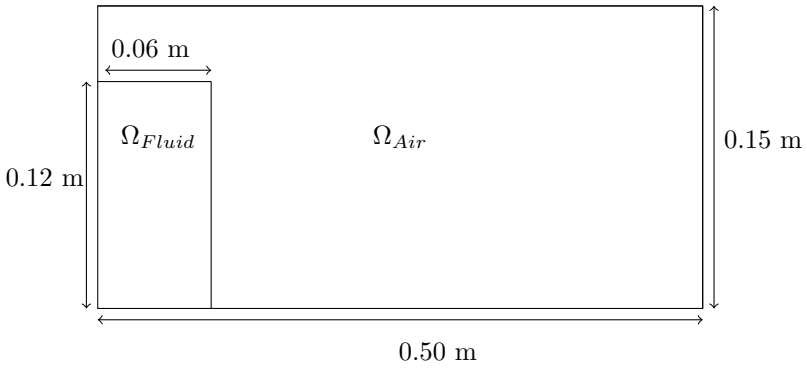


Figure 3: Set up of the 2D dam break

Re = 20	Re = 100	Re = 1000
We = 2000	We = 2000	We = 2000
U = 0.003 $m.s^{-1}$	U = 0.02 $m.s^{-1}$	U = 0.2 $m.s^{-1}$
$\sigma = 1,35 \times 10^{-8} N.m^{-1}$	$\sigma = 1,2 \times 10^{-6} N.m^{-1}$	$\sigma = 1,2 \times 10^{-4} N.m^{-1}$

Table 2: Physical parametres used for the different simulations of the 2D droplet splashing on thin liquid film

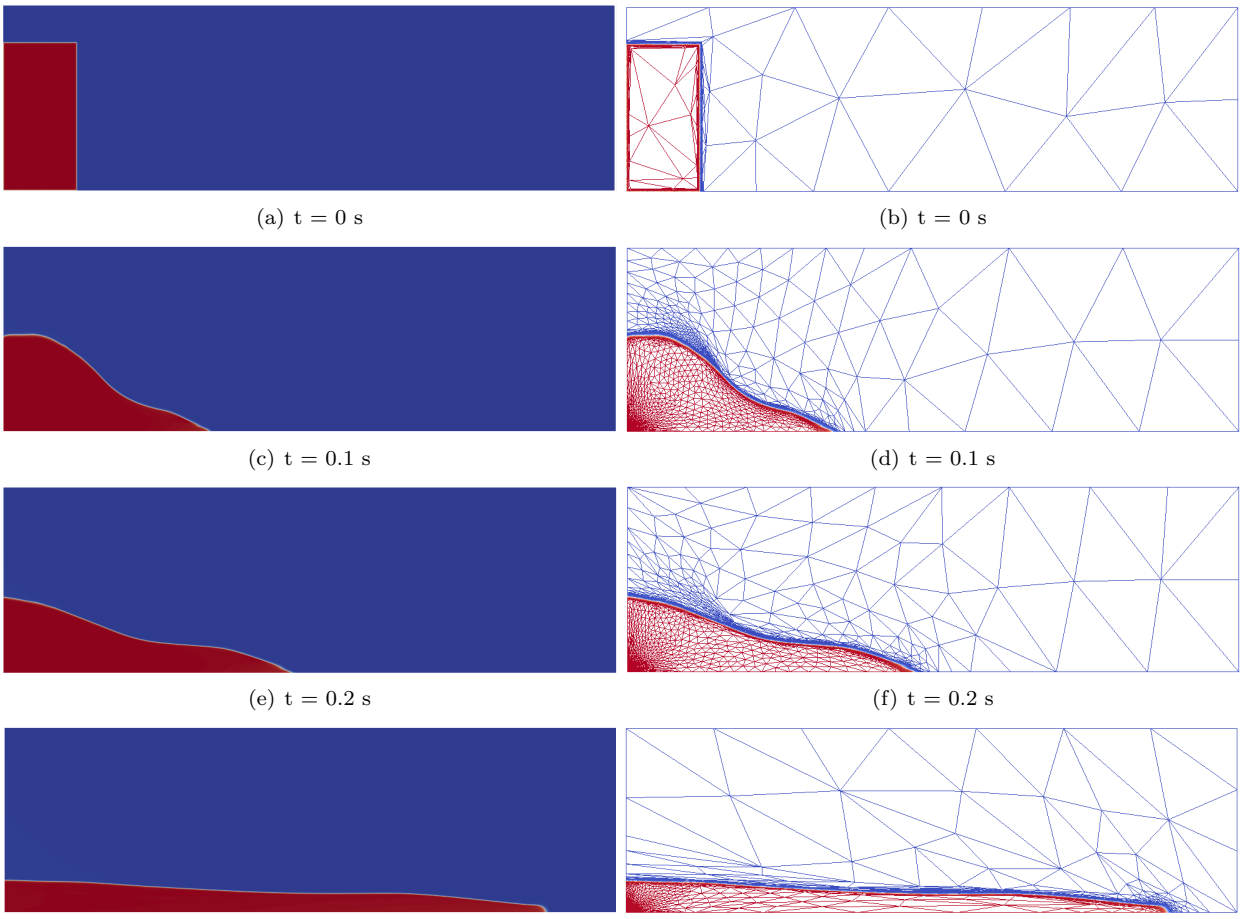
6.2. 2D dam break

The dam-break benchmark represents the study of the collapse of a fluid column in an air cavity [39]. The set up of the dam break is presented in figure 3. In this simulation, a column of water of size l = 0.06 m and h = 0.12 m is located in the left side of a tank (l = 0.50 m and h = 0.15 m). Water has a density of 1000 $kg.m^{-3}$ and the air has a constant density of 1 $kg.m^{-3}$. A free slip boundary condition is prescribed on the vertical walls and a no-slip condition is prescribed on the top and the bottom of the domain. Figure 4 illustrates the evolution of the free surface of the breaking dam and highlights the stretching of the elements near the interface. The kinematics of the fall has been studied and compared to experimental data [39] and other numerical works [40–42]. Figure 5 shows the evolution of the non-dimensional front position versus the non-dimensional time, and in figure 6, the height of the liquid column is plotted. To present the numerical results, the same non-dimensional units have been used in order to allow a direct comparison with the original publication of [39]. The obtained results are in good agreement with both experimental and numerical benchmarks.

6.3. 2D droplet splashing on thin liquid film at different Reynold numbers

In this section, the problem of a droplet splashing on a thin liquid film with large density ratio and high Reynolds numbers is investigated [43, 44]. The set up of the problem is shown in figure 7 : a liquid droplet of diameter $d = 6 \times 10^{-3}$ m is moving downward with a velocity u to a thin liquid film of height $h = 8 \times 10^{-4}$ m in an ambient vapor field. The dimensions of the computational domain are $[0, 2.5d] \times [0, 10d]$. We denote by $(\rho_{Fluid}, \mu_{Fluid})$ and (ρ_{Air}, μ_{Air}) the density and the viscosity inside the fluid domain and the air around it. This is a very challenging test case because the two fluids have large density and viscosity differences. We set $\rho_{Fluid}/\rho_{Air} = 10^3$ and $\mu_{Fluid}/\mu_{Air} = 10^2$. Three simulations at different Reynolds numbers have been carried out; the parameters used for each one are depicted in table 2.

Figures 8-10 show the time evolution of the droplet and the thin liquid film after the instant of impact. The interface of the droplet and the thin liquid film are perfectly represented by the dynamic mesh adaptation with multi criteria. We can clearly see the impact of using different Reynolds numbers. When the Reynolds number is low (figure 8), the impact does not result in splashing, the droplet only spreads gently on the surface, it is called the deposition process. As the Reynolds number is increased to 100 (figure 9), clear splashing phenomenon is observed : two liquid fingers at the end of the rim of the splashing are generated



(a) t = 0 s (b) t = 0 s

(c) t = 0.1 s (d) t = 0.1 s

(e) t = 0.2 s (f) t = 0.2 s

(g) t = 0.3 s (h) t = 0.3 s

Figure 4: Column fall evolution and refined meshes

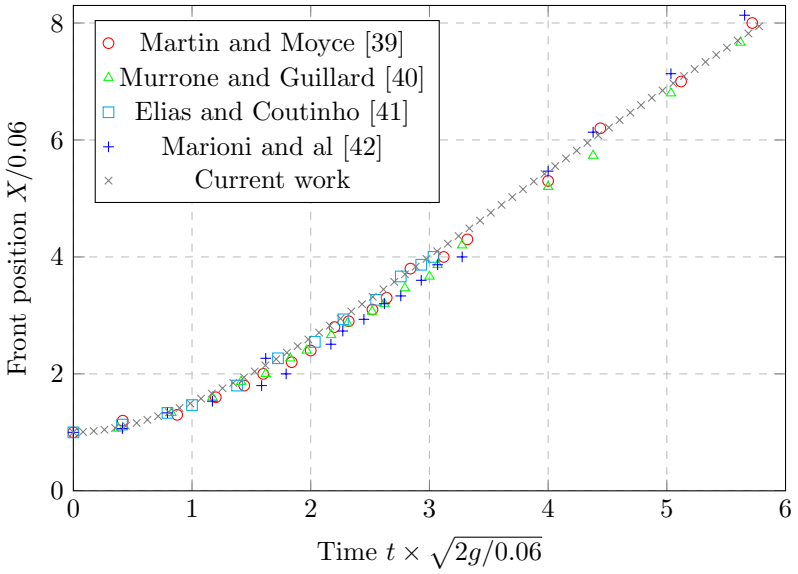


Figure 5: Non-dimensional front position evolution

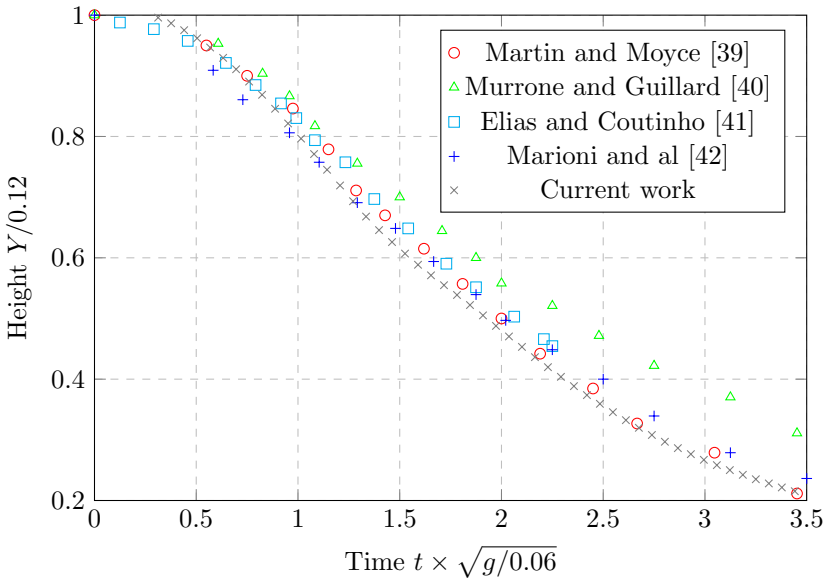


Figure 6: Non-dimensional column height evolution

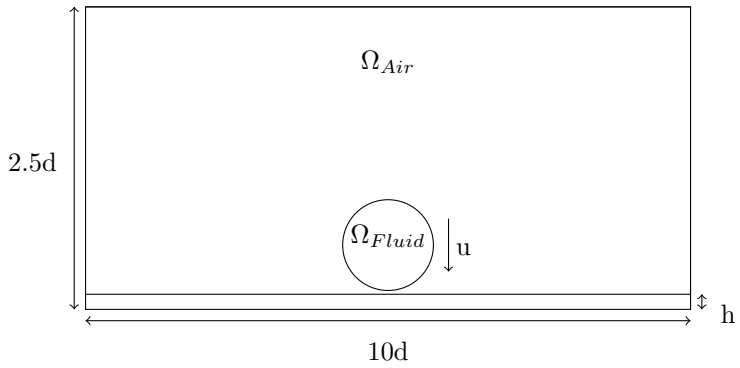
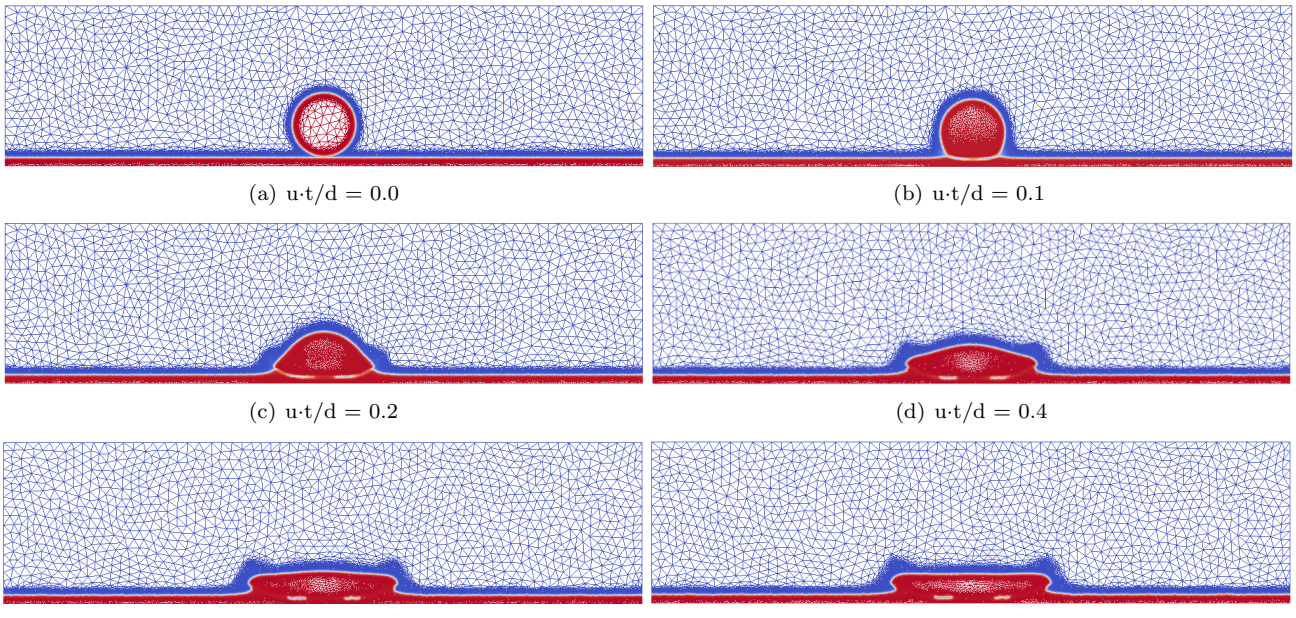


Figure 7: Set up of the 2D droplet splashing on thin liquid film

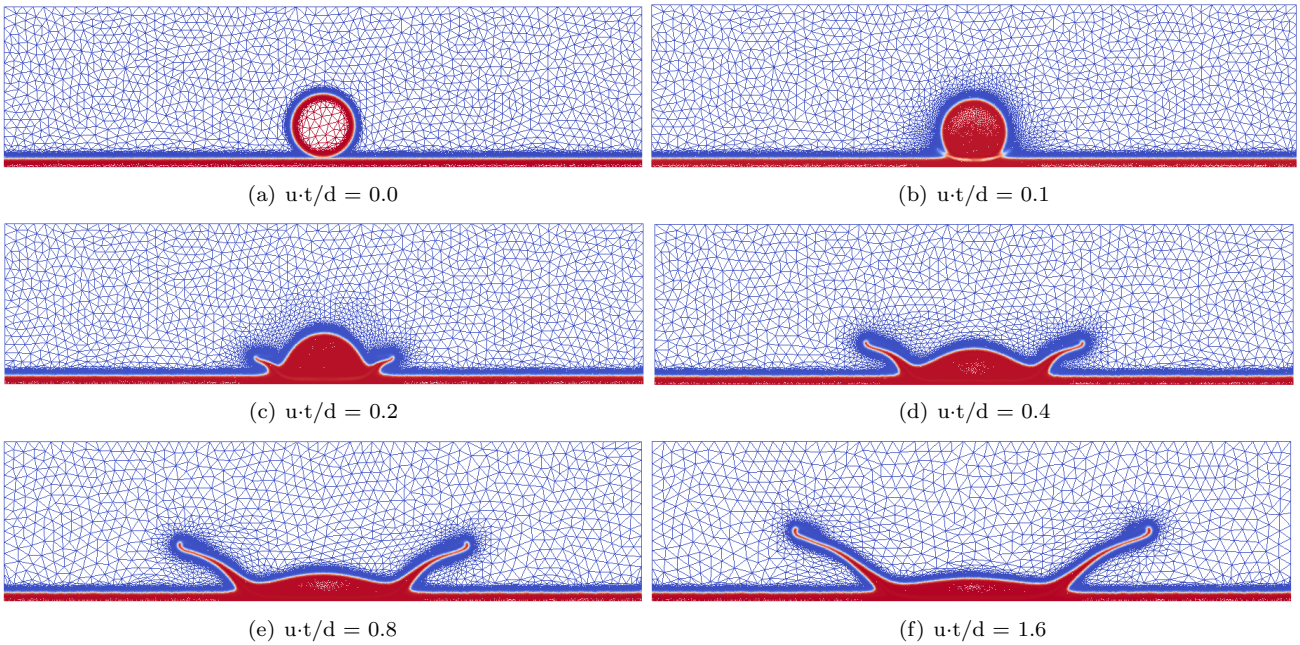


(a) u·t/d = 0.0 (b) u·t/d = 0.1

(c) u·t/d = 0.2 (d) u·t/d = 0.4

(e) u·t/d = 0.8 (f) u·t/d = 1.6

Figure 8: Time evolution of droplet splashing on a thin film at Re = 20, We = 2000

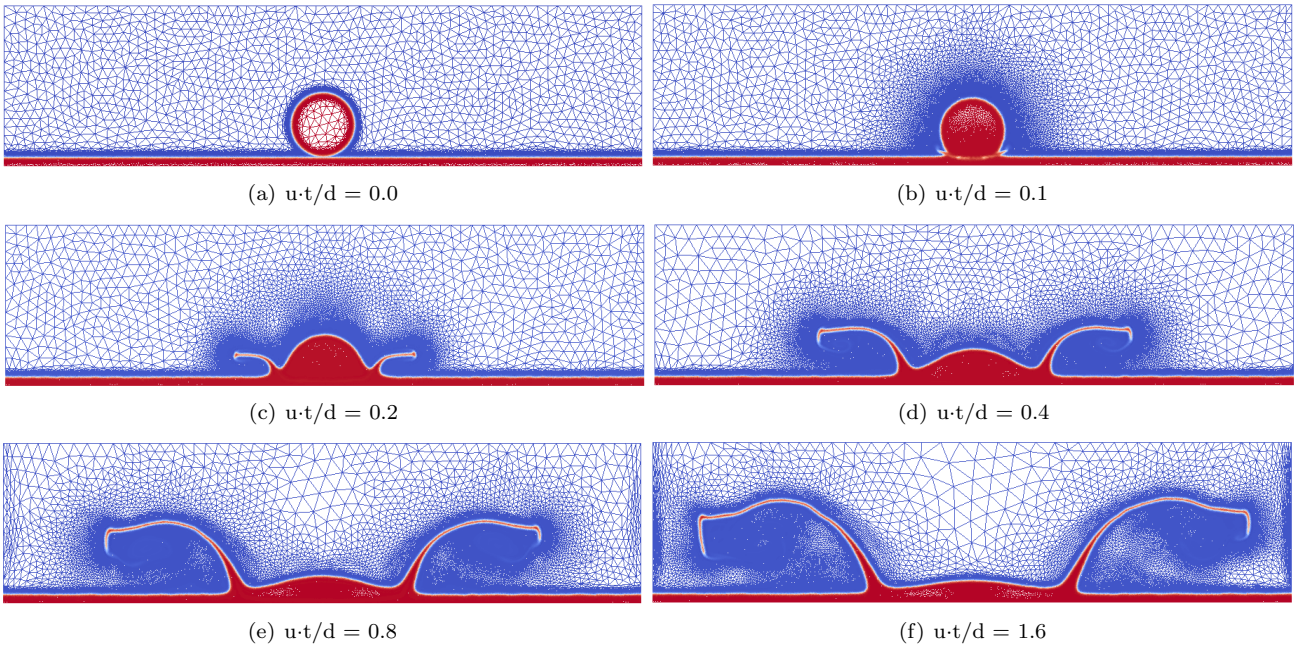


(a) u·t/d = 0.0 (b) u·t/d = 0.1

(c) u·t/d = 0.2 (d) u·t/d = 0.4

(e) u·t/d = 0.8 (f) u·t/d = 1.6

Figure 9: Time evolution of droplet splashing on a thin film at Re = 100, We = 2000



(a) u·t/d = 0.0 (b) u·t/d = 0.1

(c) u·t/d = 0.2 (d) u·t/d = 0.4

(e) u·t/d = 0.8 (f) u·t/d = 1.6

Figure 10: Time evolution of droplet splashing on a thin film at Re = 1000, We = 2000

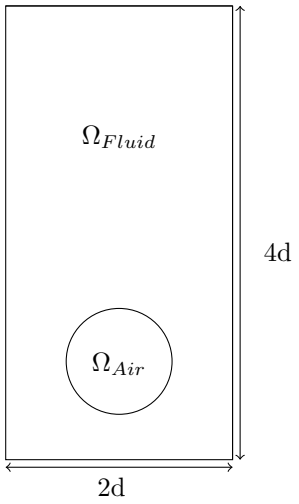


Figure 11: Set up of the 2D rising bubble

Adaptation Frequency	**Linear Interpolation**	**Conservative Interpolation**
1	29.04 %	6.68 %
1/5	19.70 %	4.77 %

Table 3: Mass variation for the 2D rising bubble with 5000 elements

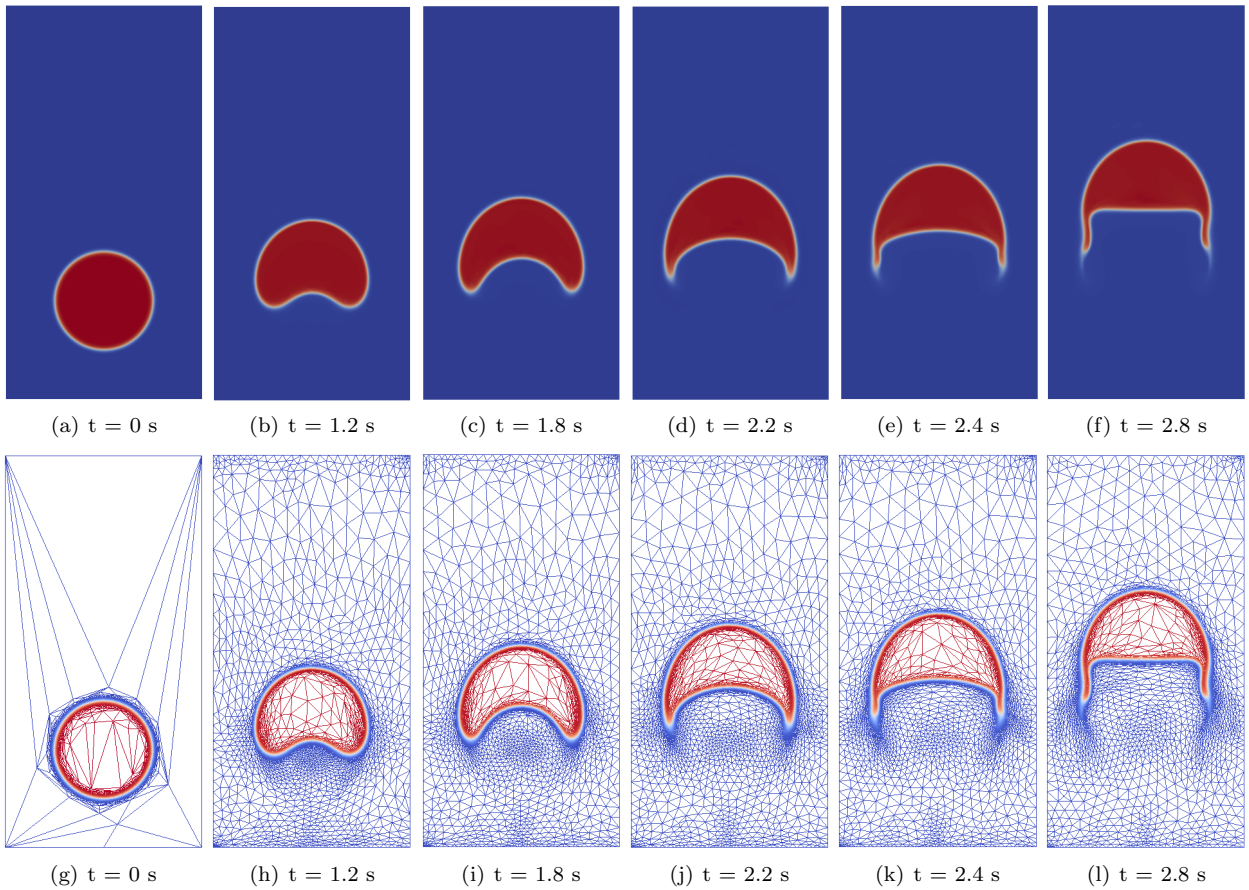
after the impact of the droplet. Similar observations were reported in the work of [45]. As the Reynolds number increases to 1000 (figure 10), the height and the speed of the splashing is higher and we can also note that the thickness of the fingers becomes smaller, which forms a thin film in the direction perpendicular to the plane. It is important to mention that in this work the simulation of the whole domain was carried out, unlike in the work of [46] where they consider half of the computational domain. Still the results reflect the same symmetry assumed in their work.

6.4. 2D rising bubble

To validate the numerical framework, we conducted a 2D simulation of a single rising bubble under buoyancy force [47]. This test case has already been investigated by our research team in [48], thus in this paper we only focus on the mesh adaptation and the conservative interpolation. A rising bubble of diameter $d = 0.5\ m$ is in water with a density ratio of 1000 and a viscosity ratio of 100. We use a rectangular domain of sizes $[0, 2d] \times [0, 4d]$, the set sup is shown in figure 11. A free slip boundary condition is prescribed on the vertical walls and a no-slip condition is prescribed on the top and the bottom of the cavity. We performed two simulations, using adaptive anisotropic meshing, under the constraint of a fixed number of elements. The criteria chosen for the adaptive meshing is again both the velocity (in direction and magnitude) and the level-set function. Figure 12 shows the evolution of the shape of the bubble as well as the associated anisotropic meshes associated at different time steps. The high surface tension coefficient maintains the bubble as an ellipsoid. In this section, we want to emphasize the benefits in terms of conservation of our numerical framework and this even when dealing with both dynamic and anisotropic meshes. Four simulations have been carried out, two are performed using about 5000 elements and the others were performed using about 10 000 elements, while changing the adaptation frequency (we adapt every time step or every 5 steps). The accumulation of mass loss when using linear interpolation and conservative interpolation are depicted in tables 3-4. As expected, the tables highlight a significant difference between both approaches, confirming again the important role of the proposed global conservative method.

Adaptation Frequency	Linear Interpolation	Conservative Interpolation
1	19.05 %	3.86 %
1/5	16.82 %	2.37 %

Table 4: Mass variation for the 2D rising bubble with 10 000 elements



(a) t = 0 s (b) t = 1.2 s (c) t = 1.8 s (d) t = 2.2 s (e) t = 2.4 s (f) t = 2.8 s

(g) t = 0 s (h) t = 1.2 s (i) t = 1.8 s (j) t = 2.2 s (k) t = 2.4 s (l) t = 2.8 s

Figure 12: 2D rising bubble evolution and refined meshes

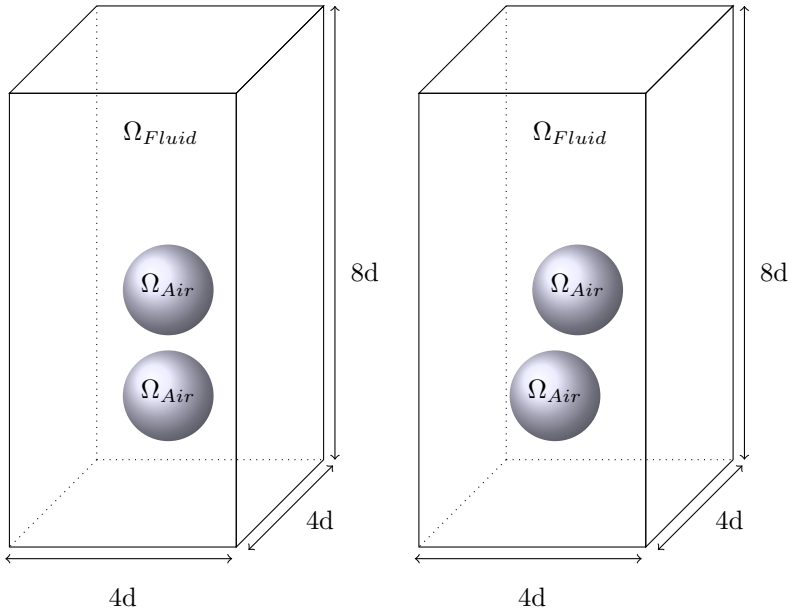


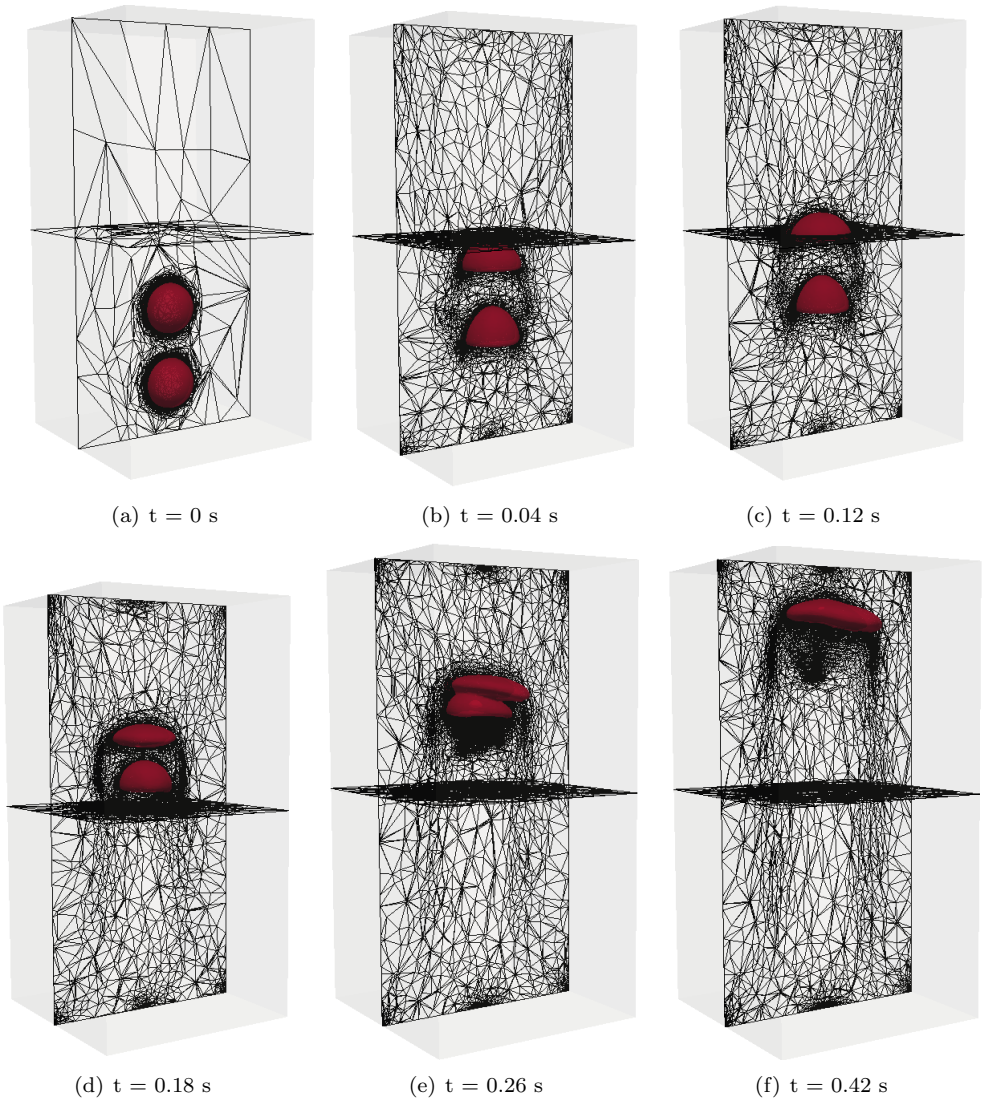
Figure 13: Set up of the 3D simulations of the axisymetric and non-axisymetric merging of two bubbles

6.5. Fully 3D simulations of the axisymetric and non-axisymetric merging of two bubbles

To test high efficiency of the method for two-fluid flows involving complex topological changes of the interface, we consider the merging of two air bubbles with first co-axial and then oblique coalescence. This test case was studied by [49, 50] and the results suggested that due to surface tension a merging will occur. We consider two gas bubbles with a diameter $d = 0.01\ m$ that are initially set in a quiescent liquid. The density and viscosity ratios (gas to liquid) are respectively 0.001 and 0.01. The dimensionless parameters we use for this problem are the Eotvos number $Eo = 16$ and the Morton number $M = 2 \times 10^{-4}$. The dimensions of the computational domain are $[-2d, 2d] \times [0, 8d] \times [-2d, 2d]$. For the co-axial case, the center of the upper bubble is at $[0, 2.5d, 0]$ and the lower bubble is at $[0, d, 0]$. As for the non-axisymetric case, the lower bubble is shifted to the position $[0.8d, d, 0]$. We use free-slip boundary conditions on all sides of the domain. It can easily be observed in figures 14 and 15 that the two bubbles gradually merge together and form a larger bubble under capillary force. For both cases, the computed evolution of the bubble shape is well captured by the dynamic mesh adaptation. The interfaces of the two bubbles are very well captured and the boundary layers as well as the detachments are automatically detected. Our results agree with the experimental photographs in [49] and the numerical results of [51]. We can see that the evolution of the lower bubble is completely different from the leading one. The tailing bubble catches the leading bubble later to form one single bubble. For the non-axisymetric case shown in figure 15, the dynamics are similar to the previous one except we can note that the flow field is clearly asymmetrical.

7. Conclusion

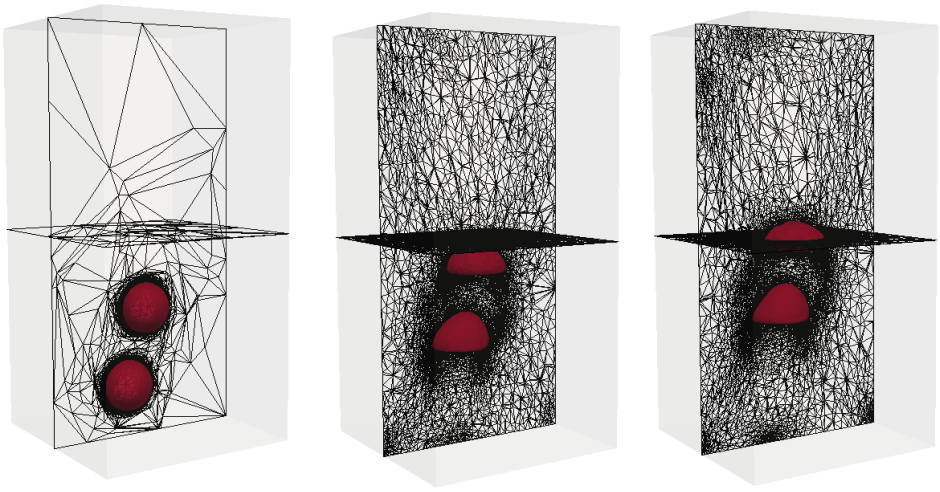
We proposed in this paper, a new convective-reactive level-set method coupled with a conservative mesh adaptation technique. Using a new definition of the level-set function that describes implicitly the interface between the two-fluid flows, combined with an edge based error estimator for anisotropic mesh adaptation and conservative interpolation, it results in an accurate and conservative framework that allows simulating liquid-gas flows. All the details of the new numerical framework were presented in this paper. Several time-dependent test cases with two-fluid flows were presented and analyzed. The numerical results and the comparisons with the literature show that the new framework is able to exhibit good accuracy and



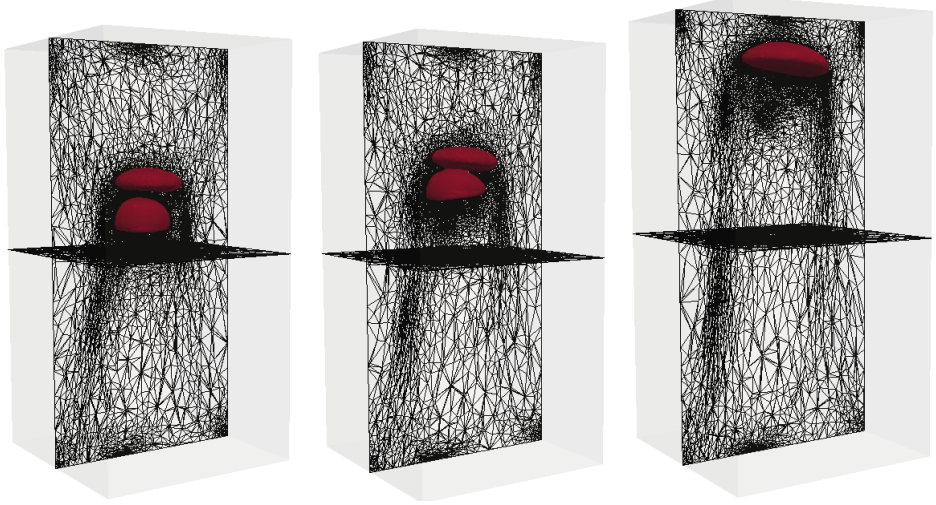
(a) t = 0 s (b) t = 0.04 s (c) t = 0.12 s

(d) t = 0.18 s (e) t = 0.26 s (f) t = 0.42 s

Figure 14: Time evolution of the bubble shape for the co-axial coalescence



(a) t = 0 s (b) t = 0.04 s (c) t = 0.12 s



(d) t = 0.18 s (e) t = 0.26 s (f) t = 0.42 s

Figure 15: Time evolution of the bubble shape for the oblique coalescence

conservation properties on anisotropic meshes with highly stretched elements. Further investigations will focus on the study of other three dimensional simulations using high performance computing.

Acknowledgments

The author gratefully acknowledges support from the company Montupet involved in the project.

References

[1] S. Elgeti, H. Sauerland, Deforming fluid domains within the finite element method: five mesh-based tracking methods in comparison, Archives of Computational Methods in Engineering 23 (2) (2016) 323–361.
[2] J. Glimm, J. W. Grove, X. L. Li, K.-m. Shyue, Y. Zeng, Q. Zhang, Three-dimensional front tracking, SIAM Journal on Scientific Computing 19 (3) (1998) 703–727.
[3] J. Glimm, J. W. Grove, X. Li, N. Zhao, Simple front tracking, Contemporary Mathematics 238 (2) (1999) 133–149.
[4] D. Juric, G. Tryggvason, A front-tracking method for dendritic solidification, Journal of computational physics 123 (1) (1996) 127–148.
[5] D. Juric, G. Tryggvason, Computations of boiling flows, International Journal of Multiphase Flow 24 (3) (1998) 387–410.
[6] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, Journal of Computational Physics 169 (2) (2001) 708–759.
[7] S. O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, Journal of computational physics 100 (1) (1992) 25–37.
[8] H. Udaykumar, W. Shyy, M. Rao, Elafint: a mixed eulerian–lagrangian method for fluid flows with complex and moving boundaries, International journal for numerical methods in fluids 22 (8) (1996) 691–712.
[9] D. J. Benson, Computational methods in lagrangian and eulerian hydrocodes, Computer methods in Applied mechanics and Engineering 99 (2-3) (1992) 235–394.
[10] D. J. Benson, Volume of fluid interface reconstruction methods for multi-material problems, Applied Mechanics Reviews 55 (2) (2002) 151–165.
[11] C. W. Hirt, B. D. Nichols, Volume of fluid (vof) method for the dynamics of free boundaries, Journal of computational physics 39 (1) (1981) 201–225.
[12] J. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, Journal of computational physics 100 (2) (1992) 335–354.
[13] N. V. Deshpande, Fluid mechanics of bubble growth and collapse in a thermal ink jet printhead, in: Hard Copy Output, Vol. 1079, International Society for Optics and Photonics, 1989, pp. 273–282.
[14] E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, W. J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, Journal of computational physics 130 (2) (1997) 269–282.
[15] S. Osher, R. P. Fedkiw, Level set methods: an overview and some recent results, Journal of Computational physics 169 (2) (2001) 463–502.
[16] J. A. Sethian, Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, Vol. 3, Cambridge university press, 1999.
[17] J. A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annual review of fluid mechanics 35 (1) (2003) 341–372.
[18] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational physics 114 (1) (1994) 146–159.
[19] M. Sussman, E. G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, Journal of computational physics 162 (2) (2000) 301–337.
[20] M. Sussman, A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles, Journal of Computational Physics 187 (1) (2003) 110–136.
[21] G. Russo, P. Smereka, A remark on computing distance functions, Journal of Computational Physics 163 (1) (2000) 51–67.
[22] A. du Chéné, C. Min, F. Gibou, Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes, Journal of Scientific Computing 35 (2-3) (2008) 114–131.
[23] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, Journal of Computational physics 183 (1) (2002) 83–116.
[24] P. Trontin, S. Vincent, J.-L. Estivalezes, J.-P. Caltagirone, A subgrid computation of the curvature by a particle/level-set method. application to a front-tracking/ghost-fluid method for incompressible flows, Journal of Computational Physics 231 (20) (2012) 6990–7010.
[25] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, Journal of computational physics 227 (4) (2008) 2674–2706.
[26] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive cartesian grids, Journal of Computational Physics 225 (1) (2007) 300–321.
[27] F. Gibou, R. Fedkiw, S. Osher, A review of level-set methods and some recent applications, Journal of Computational Physics.
[28] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, Journal of computational physics 210 (1) (2005) 225–246.

[29] O. Desjardins, H. Pitsch, A spectrally refined interface approach for simulating multiphase flows, Journal of computational physics 228 (5) (2009) 1658–1677.
[30] S. Osher, R. Fedkiw, Level set methods and dynamic implicit surfaces, Vol. 153, Springer Science & Business Media, 2006.
[31] K. Yokoi, A density-scaled continuum surface force model within a balanced force formulation, Journal of Computational Physics 278 (2014) 221–228.
[32] E. Hachem, B. Rivaux, T. Kloczko, H. Digonnet, T. Coupez, Stabilized finite element method for incompressible flows with high reynolds number, Journal of computational physics 229 (23) (2010) 8643–8665.
[33] R. Codina, Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods, Computer methods in applied mechanics and engineering 190 (13-14) (2000) 1579–1599.
[34] T. Coupez, Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing, Journal of computational physics 230 (7) (2011) 2391–2405.
[35] T. Coupez, Génération de maillage et adaptation de maillage par optimisation locale, Revue européenne des éléments finis 9 (4) (2000) 403–423.
[36] T. Coupez, E. Hachem, Solution of high-reynolds incompressible flow with stabilized finite element and adaptive anisotropic meshing, Computer methods in applied mechanics and engineering 267 (2013) 65–85.
[37] C. Bahbah, Y. Mesri, E. Hachem, Interpolation with restrictions in an anisotropic adaptive finite element framework, Finite Elements in Analysis and Design 142 (2018) 30–41.
[38] F. Alauzet, M. Mehrenberger, P1-conservative solution interpolation on unstructured triangular meshes, International Journal for Numerical Methods in Engineering 84 (13) (2010) 1552–1588.
[39] J. C. Martin, W. J. Moyce, W. G. Penney, A. Price, C. Thornhill, Part iv. an experimental study of the collapse of liquid columns on a rigid horizontal plane, Phil. Trans. R. Soc. Lond. A 244 (882) (1952) 312–324.
[40] A. Murrone, H. Guillard, A five equation reduced model for compressible two phase flow problems, Journal of Computational Physics 202 (2) (2005) 664–698.
[41] R. N. Elias, A. L. Coutinho, Stabilized edge-based finite element simulation of free-surface flows, International Journal for Numerical Methods in Fluids 54 (6-8) (2007) 965–993.
[42] L. Marioni, M. Khalloufi, F. Bay, E. Hachem, Two-fluid flow under the constraint of external magnetic field: Revisiting the dam-break benchmark, International Journal of Numerical Methods for Heat & Fluid Flow 27 (11) (2017) 2565–2581.
[43] C. Josserand, S. Zaleski, Droplet splashing on a thin liquid film, Physics of fluids 15 (6) (2003) 1650–1657.
[44] Q. Li, K. Luo, X. Li, Lattice boltzmann modeling of multiphase flows at large density ratio with an improved pseudopotential model, Physical Review E 87 (5) (2013) 053301.
[45] Y. Wang, C. Shu, H. Huang, C. Teo, Multiphase lattice boltzmann flux solver for incompressible multiphase flows with large density ratio, Journal of Computational Physics 280 (2015) 404–423.
[46] H. Yuan, Z. Chen, C. Shu, Y. Wang, X. Niu, S. Shu, A free energy-based surface tension force model for simulation of multiphase flows by level-set method, Journal of Computational Physics 345 (2017) 404–426.
[47] S.-R. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics, International Journal for Numerical Methods in Fluids 60 (11) (2009) 1259–1288.
[48] M. Khalloufi, Y. Mesri, R. Valette, E. Massoni, E. Hachem, High fidelity anisotropic adaptive variational multiscale method for multiphase flows with surface tension, Computer Methods in Applied Mechanics and Engineering 307 (2016) 44–67.
[49] G. Brereton, D. Korotney, I. Sahin, G. Tryggvason, Dynamics of bubbles and vortices near a free surface, New York: ASME, AMD 119.
[50] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, M. L. Welcome, An adaptive level set approach for incompressible two-phase flows, Journal of Computational Physics 148 (1) (1999) 81–124.
[51] Z. Xie, D. Pavlidis, P. Salinas, J. R. Percival, C. C. Pain, O. K. Matar, A balanced-force control volume finite element method for interfacial flows with surface tension using adaptive anisotropic unstructured meshes, Computers & Fluids 138 (2016) 38–50.