Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 1923–1942
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

# Transient adaptivity applied to two-phase incompressible flows

Gaëtan Compère [a,c,*], Emilie Marchandise [a,c], 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, rue d'Egmont 5, 1000 Bruxelles, Belgium

Received 31 January 2007; received in revised form 3 August 2007; accepted 1 October 2007
Available online 12 October 2007

**Abstract**

An anisotropic adaptation process is applied to a three-dimensional incompressible two-phase flow solver. The solver uses a level set/finite element method on unstructured tetrahedral meshes. We show how the level set function can be used to build an anisotropic mesh with good properties. Some computations with a strong transient character and large densities ratios (1/1000) are presented. We show that the efficiency of the computations can be deeply enhanced by mesh adaptations.


*Keywords:* Adaptivity; Anisotropic; Two-phase; Finite elements; Level set

## 1. Introduction

Among all the problems that have to be addressed in the numerical simulations of two-phase flow problems, obtaining an accurate representation of the interface is certainly the most difficult issue. At least two arguments lead to this observation:

- most of the fluid vorticity is concentrated near the interface [1],
- the rapid variation of the fluid properties generates spurious oscillations in the velocities close to the interface [2].

Those issues are even more critical in the presence of large density and/or viscosity jumps across the interface. This is indeed the case in many practical applications, and among those are the ones that involve interfaces between air and water. There, the density ratio is about 1/1000 while the viscosity ratio is about 1/100.

* Corresponding author. Address: Université catholique de Louvain, Department of Civil Engineering, Place du Levant 1, 1348 Louvain-la-Neuve, Belgium.
*E-mail address:* gaetan.compere@uclouvain.be (G. Compère).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.10.002

In previous works [3–5], our team has developed a technique to accurately model three dimensional two-phase flows. A level set [4,6,7] approach was chosen to model the interface.[1] The level set function $\phi(\vec{x},t)$ is an implicit function that is defined on the whole mesh. Its iso-zero $\phi_0 \equiv \phi(\vec{x},t) = 0$ is a surface that evolves in time and that represents the fluid interface. The property of the fluid at one point $\mathbf{x}$ is identified by the sign of the function. The level set function $\phi$ is transported and deformed by the fluid flow. An equation of transport is solved at each time step that allows to move the fluid interface [3]. The level set method is well suited to represent complex flows with dramatic changes in the interface topology. Applications of level set in two-phase flow calculations have been extensively described by Sussman et al. in [7,16,17] and used by [18–21] among others.

This paper starts from the following observation: in two-phase flow problems, the denser is the mesh at the vicinity of the interface, the better is the solution. In other words, an adaptive strategy is required in order to put the effort in the region where the solution is more complex.

Adaptive mesh refinement (AMR) has now been used for two decades in flow problems. The first achievements are those of Berger et al. [22,23] who developed a block-structured grid method to solve compressible [23–25] and incompressible flows problems [26]. Those methods have been more recently applied to two-phase flow problems by Sussman et al. [1,27]. Block-structured grid methods allow to increase the resolution with multiple levels of refinements on blocks located near regions where the scales cannot be resolved by the coarse grid. This technique is quite appropriate to solve problems on uniform grids, as well as the quadtree based methods. The quadtree grids are more general and more efficient than the block-structured grids in the sense that additional points are only added where it is required. Nevertheless, quadtrees require complex data structures which results in complicated algorithms and impacts on the computational time. Results of the coupling of the Euler incompressible equations with the quadtree method have been presented by Popinet [28]. Two-phase flow computations were first performed by Losasso et al. [29], and recently Greaves proposed a coupling with a volume-of-fluid method [30]. Finally, we mention the methods based on the minimization of a *mesh energy* used by Cristini et al. [31–33]. In those methods, a spring-like energy is associated with each edge and is dependent of its length and the local scales of the flow. A more detailed review of the AMR techniques coupled with level set methods and incompressible flows can be found in [34].

In this paper we describe an alternative method to the techniques presented above. Its simplicity allows robust anisotropic mesh adaptations [35,36]. Because our aim is to adapt the mesh in time, we use an adaptive procedure that is based on local mesh modification operators. Starting from an existing mesh, we apply local mesh modification operators until each edge of the resulting mesh has a non-dimensional length that is close to 1. Non-dimensional edge lengths are computed using a non-uniform anisotropic metric field that is based on the position of the interface. An interest of using anisotropic elements is that the high variations of density and viscosity across the interface can be captured in a more efficient way by having more elements along the crossing direction. As a consequence, mesh adaptation can be seen as an alternative to the common solution consisting in smoothing the fluid properties on a thin layer around the interface [17,21]. In the same way, the high pressure gradients generated at the interface in presence of surface tension forces can be evaluated more efficiently with anisotropic elements, which reduces spurious velocities.

The remainder of this paper is organized as follows. In Section 2, the physical model is described as well as the numerical methods used in the flow solver. Section 3 presents the adaptation technique while Section 4 gives numerical examples that show the accuracy and the efficiency of the method.

## 2. Two-phase flow computation

In this section, we give a rapid summary of the different ingredients that are used in the simulations: constitutive equations, numerical methods and coupling between the fluid and the interface solvers. For more details of the recipe, see [4].

[1] Level set and volume of fluid [8–10] can be classified as interface capturing methods. They differ from interface tracking methods [11–15] in the sense that interface tracking methods use an explicit representation of the interface.

### 2.1. Constitutive equations

Our solver computes three-dimensional laminar flows involving two incompressible non-miscible fluids. The fluids are identified by (+) and (−), and their density and viscosity are respectively $(\rho_+, \mu_+)$ and $(\rho_-, \mu_-)$.

The physics of the fluids are given by the two-phase incompressible Navier–Stokes equations:

$$\frac{D\mathbf{u}}{Dt} = -\frac{\nabla p}{\rho(\phi)} + \frac{1}{\rho(\phi)}\frac{1}{Re}\nabla\cdot(2\mu(\phi)S) + \frac{\mathbf{e_g}}{Fr^2} + \frac{\kappa\mathbf{n}}{We}, \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{2}$$

where $\mathbf{u}$ and $p$ are the non-dimensional velocity and static pressure of the fluids, $\rho(\phi)$ and $\mu(\phi)$ are the non-dimensional density and dynamic viscosity, $S = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}})$ is the deformation rate tensor, $\mathbf{e_g}$ is the direction in which the gravity ($\mathbf{g}$) acts, $\kappa$ is the curvature of the interface and $Re$, $Fr$ and $We$ are the numbers of Reynolds, Froude and Weber defined as

$$Re = \frac{\rho_{\mathrm{R}} U_{\mathrm{R}} L_{\mathrm{R}}}{\mu_{\mathrm{R}}}, \quad Fr = \frac{U_{\mathrm{R}}}{\sqrt{g L_{\mathrm{R}}}}, \quad We = \frac{\rho_{\mathrm{R}} U_{\mathrm{R}}^2 L_{\mathrm{R}}}{\sigma_{\mathrm{R}}}, \tag{3}$$

where the subscript R denotes the reference value and $\sigma$ is the surface tension. The solutions in both phases are obtained simultaneously.

The interface is captured using a level set method. This method implies a function $\phi$ named the *level set* whose value is 0 on the interface ($\phi_0$), positive in one fluid and negative in the other. Complex configurations like sloshing problems with bubbles, separations and merging can then be represented.

This function is advected using a tracer equation that can be written in a conservative form:

$$\partial_t\phi + \nabla\cdot(\mathbf{u}\phi) = 0. \tag{4}$$

Further details about the level set method in two-phase flow problems can be found in [4,16].

### 2.2. Numerical method

We use a finite element method for computing the fluid flow (Eqs. (1) and (2)). Both velocity and pressure are approximated using a piecewise linear continuous polynomials. As this representation violates the Babuska–Brezzi (BB) condition [37], spurious pressure modes have to be removed from the solution. These oscillations can be avoided using a pressure stabilized Petrov Galerkin (PSPG) method. The inviscid fluxes are discretized in a stable manner using an upwind finite volume stabilization [38,39].

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 [40] is used to solve at each time step the system of nonlinear equations. The iterative solution of the large sparse linear systems that arises at each Newton iteration is solved by the GMRES method preconditioned by the RAS [41] algorithm.

The interface equation (4) is discretized using discontinuous finite elements. The solution is represented using piecewise discontinuous polynomials ($N^p$) of order $p$. The discontinuity of this numerical approximation is not to be confused with the discontinuous nature of the solution of two-phase flows. In other methods like in the ghost fluid method [42] the physical discontinuity is exactly fitted by a discontinuity in the model and a specific treatment of the interface is designed to handle the jump conditions. This specific treatment leads to the advantage of very small spurious velocities. Here, the interface is represented by the iso-zero of a continuous level set function which is *discretized* in a discontinuous way that is not related to the physical discontinuities. The surface tension forces are computed according to the *continuum surface force* method of Brackbill et al. [43]. Further details about the computation of the surface tension forces can be found in [5]. The discontinuity of the physical properties of the fluid (density and viscosity) are handled by an exact discontinuous integration over the elements overlapping the interface [44,45].

As Eq. (4) involves $\nabla\phi$ and $\mathbf{u}$, we should use a discretization of the level set for which the gradient is at least in the space of the velocity, i.e. $p \geqslant 2$. Nevertheless, as it is shown for the two-dimensional dam break problem presented in the last section of this paper, the use of first order polynomials is often more efficient if we use an

adaptation procedure. Indeed, for the same computational time, we can build a finer mesh near the interface with a first order discretization than with a second or a third. The difference in terms of accuracy due to the order of discretization will be exceeded by the level of refinement of the mesh.

A Runge–Kutta algorithm of order $p+1$ is employed for the time discretization. More details about the time and space discretization of the level set function can be found in [3].

### 2.3. Coupling of the two solvers

The coupling between the fluid and the interface solvers requires a special attention as the discretization differs. The level set function is discretized by using discontinuous $p$-order elements, while the flow solver uses continuous linear approximations ($N^1$) for the velocity $\mathbf{u}$

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

where $n_p$ is the number of Lagrangian points in each tetrahedral element

$$n_p = (p+1)(p+2)(p+3)/6.$$

For efficiency reasons, the same mesh is used by both solvers.

The algorithm that couples the solvers is summarized as follows. At initial time $t^0$, initialize the level set $\phi$. Then, for each time $t^n$, $n = 1, 2, \ldots$

(1) Solve the Navier–Stokes equations in time $t \in [t,\ t+\Delta t]$ to find $\mathbf{u}(t+\Delta t)$ and $p(t+\Delta t)$.
(2) Project the velocity field onto the degrees of freedom of the level set.
(3) Solve the interface equation in time $t \in [t, t+\Delta t]$ using sub-time steps to find $\phi(t+\Delta t)$.
(4) Project the level set function onto the degrees of freedom of the velocity.
(5) Increment in time $t = t+\Delta t$ and go back to step (1).

## 3. Mesh adaptation

The appropriate way to ensure that a mesh-based numerical analysis procedure produces the most effective solution results is to apply an adaptive solution strategy. Efforts on the development of these techniques have been underway for over 25 years.

In this work, a local approach is used for adapting the mesh. The size field that is used for building the optimized mesh is build up using principally the fluid interface data's.

### 3.1. Adapting the mesh using local mesh modifications

It is only recently that authors have developed adaptive methods for transient problems that are able to be applied to unstructured grids. Transient adaptive simulations require to modify the mesh in time. For that purpose, two approaches are possible. The first one requires to build a new mesh any time the mesh has to be adapted [46–48]. In those global remeshing techniques, the issues related to mesh-to-mesh interpolation are critical. In order to control the mesh interpolation errors, Alauzet et al. [49] were able to reduce the number of remeshings using a fixed point algorithm. A metric field $\mathcal{M}$ is computed using the intersection of metric fields at successive times $t_1, t_2, \ldots, t_n$. The simulation is rewinded at time $t_1$ and the mesh is adapted against metric field $\mathcal{M}$. The adapted mesh is valid for a large time interval and it is therefore adapted less often.

The other way of doing the adaptation in time is to locally modify the mesh [35,36,50,51]. A common belief is that doing local mesh modifications has to be faster than remeshing. This is not usually the case. Global remeshing procedures are in fact usually faster than local mesh modifications techniques. This is the case essentially because global remeshing algorithms converge much faster: they allow to create vertices that are readily at their right locations when local mesh modification procedures iteratively add and remove vertices

in the mesh. For transient adaptive computations, local mesh modifications have determinant advantages that are not linked with their computational efficiency:

- local solution projection procedures can be easily set up that ensure the exact conservation of conservative quantities [35,36],
- the mesh remains unchanged in most of the domain, allowing to adapt the mesh frequently,
- local mesh modifications can be performed in parallel, enabling transient adaptive simulation to run on parallel computers.

Local mesh modification operators all consist in replacing a cavity of elements $\mathcal{C}$ by another one $\mathcal{C}'$. In our adaptation procedure, we have set up a moment when both cavities $\mathcal{C}$ and $\mathcal{C}'$ are simultaneously present. At this point, both fluid and interface solvers are called back so that a local solution projection procedure can be performed. The solution in the new cavity $\mathcal{C}'$ is computed using the information in $\mathcal{C}$. More details about this mesh adaptation procedure can be found in previous papers [35,36,50,53].

As the level set is represented by discontinuous polynomials, projections are made at the elementary level. This is one of the major advantages of the DG method. The fluid solution, i.e. velocities and pressures, uses continuous piecewise linear approximations. Both fields are located at mesh vertices. In our projection algorithm, the only local mesh modification that requires some work is the edge splitting. When a new vertex is inserted in the mesh, the fluid solution is simply taken as the average of the solutions at the two nodes of the initial edge.

### 3.2. The mesh metric field

A mesh metric field is a smooth tensor valued $\mathcal{M}(x,y,z)$ defined over the domain. The metric at a point is a symmetric positive definite tensor. Let us consider a mesh edge $e$ that defines a vector $\mathbf{e}$ that goes from its initial vertex to its final one. The non-dimensional length $L_e$ of $e$ is computed as

$$L_e = \int_e \sqrt{\mathbf{e}^t \mathcal{M}(x,y)\mathbf{e}}\,\mathrm{d}l. \tag{5}$$

The aim of the mesh adaptation procedure is to modify an existing mesh to make it a unit mesh, i.e. a mesh for which every edge is of size $L_e = 1$. The use of a tensor valued metric field allows the construction of anisotropic meshes.

Local mesh modification operators are essentially edge-based operators in the sense that they locally modify mesh cavities composed of all the tetrahedrons. The mesh adaptation algorithm works as follow:

(1) Loop over all edges of the mesh, consider the edge $e$ of size $L_e$
   - Refinement: if $L_e > \sqrt{2}$, $e$ is a long edge. Edge $e$ is therefore split in its middle, according to the metric.
   - Coarsening: if $L_e < 1/\sqrt{2}$, $e$ is a short edge. The cavity surrounding $e$ has therefore to be investigated and a coarsening procedure [54] in which $e$ could be collapsed by merging its two nodes is applied.
   - Shape optimization: the edge $e$ is eliminated using an edge swapping algorithm if a better configuration is obtained after swapping.

(2) Go back to (1) if any short or long edge is still present in the mesh.
(3) Do one more step of shape optimization.

Typically, the algorithm stops when every edge of the domain has a dimensionless size in the interval $L_e \in [1/\sqrt{2}, \sqrt{2}]$. Using this interval for short and long edges ensures that the two new edges created by a bisection will not be short edges. Oscillations between coarsening operations and refinements are therefore prevented.

The metric field is computed at every node of the present mesh using the results of a procedure that is described below.

For every vertex $v$ of the mesh,

- Both the normal vector $\mathbf{n}$ and the normal curvature $\kappa$ of the level set are computed using the classical formulas

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

- Two vectors $\mathbf{t_1}$ and $\mathbf{t_2}$ are found such that $(\mathbf{n}, \mathbf{t_1}, \mathbf{t_2})$ form an orthonormal basis of $R^3$.
- The distance $d$ from the vertex $v$ to the interface is computed. This step is achieved using a fast search tree method, namely the Approximate Nearest Neighbor algorithm (ANN) [55,56].
- Mesh sizes $S_n$, $S_{t1}$, $S_{t2}$ are computed in the three directions $\mathbf{n}$, $\mathbf{t_1}$, $\mathbf{t_2}$ as a function of $d$ and $\kappa$. The kind of parametrization that is used in this work is presented in the remainder of this section (Section 3.3).
- The metric field $\mathcal{M}$ at vertex $v$ is computed by

$$\mathcal{M} = \mathcal{R}^{\mathrm{T}}\mathcal{D}\mathcal{R},$$
$$\mathcal{R} = (\,\mathbf{n},\quad \mathbf{t_1},\quad \mathbf{t_2}\,),\quad \text{and}\quad \mathcal{D} = \begin{pmatrix} S_n^{-2} & 0 & 0 \\ 0 & S_{t1}^{-2} & 0 \\ 0 & 0 & S_{t2}^{-2} \end{pmatrix}. \tag{6}$$

### 3.3. Size field parametrization

Because most of the complex physics involved in two-phase flows is located near the interface, the mesh is refined at the vicinity of the iso-zero of the level set. To be more quantitative, we introduce a distance parameter $2\delta$ that represents the thickness of the refined region $[\phi_0 - \delta, \phi_0 + \delta]$ around the interface i.e. the thickness of the zone that has the maximal mesh refinement. We call this region the *proximity zone*. In our algorithm, we ensure that the interface never leaves the proximity zone. When the interface gets too close to the end of the proximity zone, the mesh is modified. Fig. 1 shows schematically how the interface is kept inside the refined region. The parameter $\delta$ has to be chosen carefully. If $\delta$ is big, the number of nodes of the mesh increases and so goes the time spent in the flow solver. If $\delta$ is small, the number of mesh adaptations increases because the interface leaves the proximity zone quicker. A typical choice for $\delta$ is five times the element size in the proximity zone.

Outside the proximity zone, we build an isotropic mesh of variable size $L_{\mathrm{iso}}(d)$ (see Fig. 2) where:

$$L_{\mathrm{iso}}(d) = \begin{cases} h_0 + s_1(d-\delta) & \text{for } \delta < d \leqslant d_1, \\ h_1 + s_2(d-d_1) & \text{for } d_1 < d \leqslant d_2, \\ h_2 & \text{for } d_2 < d, \end{cases} \tag{7}$$

where $s_1$ and $s_2$ are defined by

$$s_1 = \frac{h_1 - h_0}{d_1 - \delta}, \quad s_2 = \frac{d_2 - d_1}{d_2 - d_1}. \tag{8}$$

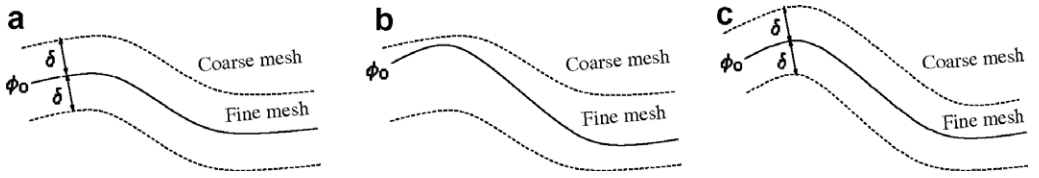



Fig. 1. Schematic view of the proximity zone during a simulation: (a) proximity zone just after a mesh adaptation procedure, (b) new position of the interface close to the boundaries after several iterations of the fluid solver and (c) new position of the proximity zone after the next mesh adaptation procedure.

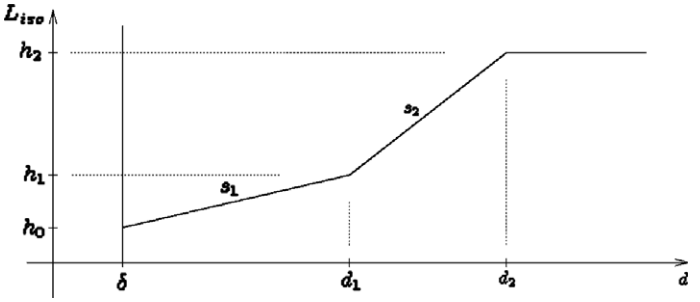



Fig. 2. Distribution of the edge sizes outside the proximity zone.

In this setup, $h_0$, $h_1$, $h_2$ are user-defined mesh sizes: $h_0$ is the smallest size that defines the resolution of the mesh in the proximity zone, $h_1$ is an intermediary size between $\delta < d < d_1$ and $h_2$ is the mesh size in the far field. Following the definition (6) of the metric, we have

$$S_n(d) = S_{t1}(d) = S_{t2}(d) = L_{\text{iso}}(d) \quad \text{for } d > \delta. \tag{9}$$

$L_{\text{iso}}$ is a parameter that is similar to the length of the edges about a sphere defined in [33], except that two slopes are defined and that constant and anisotropic lengths are prescribed in the proximity zone. The length is maintained constant in this zone in order to allow the flow solver to advect the interface with a constant resolution without having to adapt the mesh at each time step.

One of the advantages of the mesh refinement near the interface is the ability to reduce spurious velocity oscillations arising from a strong difference of properties between the fluids. We denote by $R_{\mathbf{n}}$ the mesh size reduction factor in the normal direction to the interface. The reference length $S_n$ in the proximity zone is then

$$S_n = \frac{h_0}{R_{\mathbf{n}}} \quad \text{for } d \leqslant \delta. \tag{10}$$

The ratio $R_{\mathbf{n}}$ is constant in time and space (in the proximity zone). Simulations were performed using isotropic and anisotropic mesh refinements. Fig. 3 shows the difference between results obtained with both isotropic and anisotropic refinements for a *dam break* simulation. More explanations about this simulation are

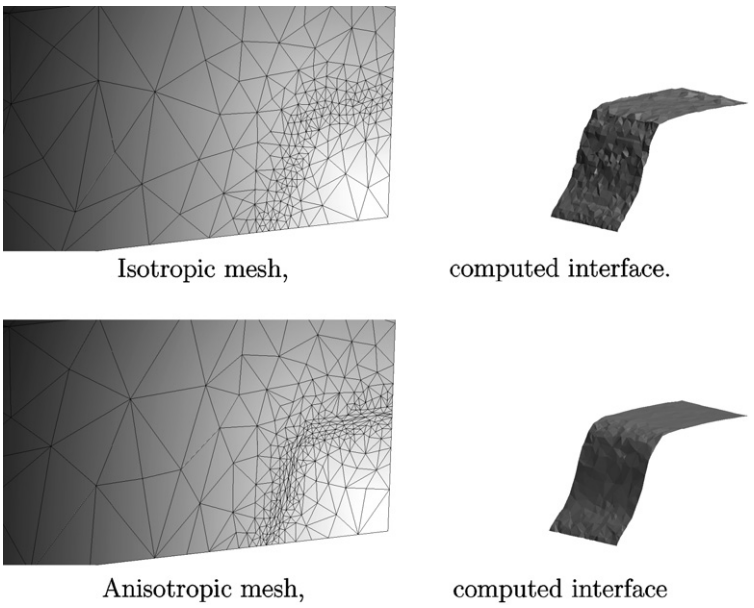



Fig. 3. Comparison of the meshes and the interfaces $\phi_0$ in a *dam break* simulation after a 'small' time for isotropic and anisotropic meshes. The meshes have almost the same number of nodes (about 2850 nodes). For the first mesh, $h_0 = 0.075$, $\delta = 0.1$. For the second, $h_0 = 0.15$, $\delta = 0.05$, $R_{\mathbf{n}} = 4$. The other parameters are identical for the two simulations.

given in the section related to the computational results. During the computation, both meshes keep almost the same number of nodes (about 2850 nodes in Fig. 3). The meshes are adapted according to the following parameters:

- for the first mesh, $h_0 = 0.075$, $\delta = 0.1$,
- for the second mesh, $h_0 = 0.15$, $\delta = 0.05$, $R_{\mathbf{n}} = 4$.

In both cases, the roughness of the interface is due to spurious velocity oscillations and the interface is clearly smoother using meshes that are adapted directionally.

Another advantage of refining the mesh at the interface is the ability to capture smaller scales. This ability can still be improved by taking the curvature $\kappa$ of the interface into account in the definition of the mesh sizes [57]. To this end, we define a reducing ratio $R_{\mathbf{t}}(\kappa)$ as follows:

$$R_{\mathbf{t}}(\kappa) = \min\left(R_{\mathbf{t},\max}, 1 + \frac{\kappa}{\kappa_{\max}}(R_{\mathbf{t},\max} - 1)\right), \tag{11}$$

where $R_{\mathbf{t},\max}$ and $\kappa_{\max}$ are arbitrary constants. This ratio is applied in the tangential directions $\mathbf{t_1}$, $\mathbf{t_2}$ to the interface:

$$S_{t1} = S_{t2} = \frac{h_0}{R_{\mathbf{t}}(\kappa)} \quad \text{for } d \leqslant \delta. \tag{12}$$

Fig. 4 summarizes the use of the ratios $R_{\mathbf{n}}$ and $R_{\mathbf{t}}(\kappa)$. Finally, the mesh refinement parameters are defined as:

- 'near' the interface: $\delta$, $h_0$, $R_{\mathbf{n}}$, $R_{\mathbf{t},\max}$ and $\kappa_{\max}$,
- 'far' from the interface: $h_1$, $h_2$, $d_1$ and $d_2$.

## 4. Computational results

Our two-phase flow solver has already been validated using fixed meshes [4]. The two first tests check that the spurious velocities are reduced when a finer refinement is applied to the mesh. They also show that the anisotropic refinements exhibit smaller spurious currents than the isotropic ones. The next test checks that the mesh adaptation procedure at least does not deteriorate the solution of the fixed mesh solver. More complex simulations are then presented to show that the method can be applied to complex and highly transient flows.

In order to make comparisons between adaptive and non-adaptive computations in terms of computational time, we have to choose an edge length $h_{na}$ for the non-adaptive mesh such that a similar accuracy is obtained with both approaches. As the non-adaptive meshes are isotropic, we define a criterion for the meshes to give similar results. We make the assumption that the accuracy at the interface $\phi_0$ only depends of the size of the edges in the normal direction to the interface, and we obtain $h_{na}$ as follows:

$$h_{na} \approx \frac{h_0}{R_{\mathbf{n}}}. \tag{13}$$

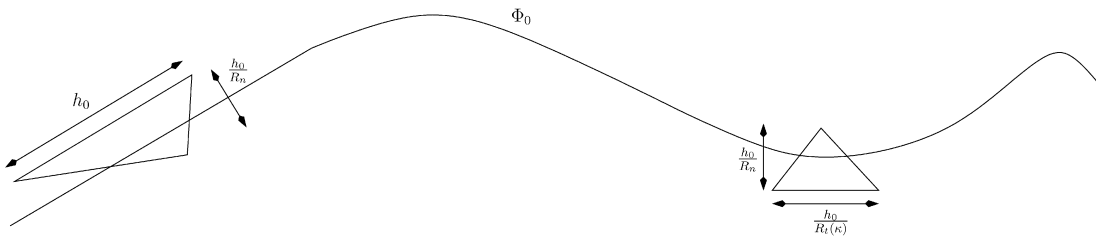



Fig. 4. Schematic view of the anisotropic aspect ratios $R_{\mathbf{n}}$ and $R_{\mathbf{t}}(\kappa)$ used near the interface.

Two effects are not taken into account with this criterion.

- First, with the adaptive technique developed in this paper, the mesh is coarse far from the interface whether it is homogeneous on the whole domain for the fixed mesh. The modeling of the flow can then be slightly different.
- Secondly, the positive effects of anisotropicity on the parasitic velocities and the roughness of the interface discussed in Section 3.3 only acts in adaptive computations.

Unless it is specified, the fluid characteristics are the following for each test ($l$ and $g$ subscribes stand for *liquid* and *gas*):

- density $\rho_l = 1000\ \mathrm{kg\ m^{-3}}$, $\rho_g = 1\ \mathrm{kg\ m^{-3}}$,
- dynamic viscosity $\mu_l = 10^{-3}$ Pa s, $\mu_g = 10^{-5}$ Pa s,

and the level set function is discretized using piecewise linear approximations. The density and viscosity ratios are close to those of air and water.

The tolerance for the GMRES solver has been prescribed to $10^{-6}$. All the computations have been performed on a Dual Core AMD Opteron Processor 265 (1800 MHz).

### 4.1. Parasitic velocities: free surface at rest

As a consequence of the high density ratio of the two fluids, spurious velocity oscillations may occur at the interface. A way to highlight these velocities is to look on how our scheme is able to resolve the hydrostatic equilibrium. Spurious velocities will hopefully decrease with the size of the elements near the interface. We consider a plane rectangular free surface in a 3D box. The two fluids are at rest at initial time $t = 0$ and no perturbation is introduced in the shape of the flat interface. The pressure is initialized to the hydrostatic equilibrium value. The velocity $\mathbf{u}$ is supposed to remain zero and the equation of motion (1) becomes

$$\frac{\nabla p}{\rho(\phi)} = \frac{\mathbf{e_g}}{Fr^2}. \tag{14}$$

If elements cross the interface, the exact solution does not belong to the finite element space and spurious velocities are generated.

The setup of this test is the following. The domain has a width of $2 \times 2$ and a height of 1 and is equally divided into air and water. We characterize the parasitic velocities by the infinity norm $|\mathbf{U_{par}}|_\infty$ of the velocities on the domain after that it has reached a stable value

$$|\mathbf{U_{par}}|_\infty = \max\left(\sqrt{\mathbf{u_{par}^2} + \mathbf{v_{par}^2} + \mathbf{w_{par}^2}}\right).$$

Table 1 shows the maximum velocities obtained for different sizes of the elements. Results have been obtained for isotropic and anisotropic meshes. For the anisotropic refinements the ratio $R_\mathbf{n}$ is 4 and the edge length $h_0$ is multiplied by $2^{\frac{1}{3}}$ in order to have the same mean volume for the elements in the isotropic and anisotropic computations.

Table 1
Stationary case: maximum parasitic velocities $|\mathbf{U_{par}}|_\infty$ with isotropic and anisotropic meshes

| $h_0^{\mathrm{iso}}$ | Isotropic | Anisotropic ($R_\mathbf{n} = 4$) |
|---|---|---|
| 0.1 | $2.76 \times 10^{-4}$ | $1.64 \times 10^{-4}$ |
| 0.01 | $1.39 \times 10^{-5}$ | $6.14 \times 10^{-6}$ |
| 0.001 | $2.38 \times 10^{-7}$ | $1.22 \times 10^{-7}$ |

For the anisotropic computations, the ratio $R_\mathbf{n}$ is 4 and the edge length $h_0^{\mathrm{aniso}} = 2^{\frac{1}{3}} h_0^{\mathrm{iso}}$.

We note that anisotropic meshes need more iterations to reach the tolerance of the GMRES solver. Table 2 shows the number of iterations needed per time step for each computation.

### 4.2. Parasitic velocities: 3D static droplet

In the case of a spherical droplet with no gravity effects, a stationary equilibrium occurs at the interface between the pressure forces and the surface tension forces and gives rise to a pressure jump at the vicinity of the interface. However, since the pressure discontinuity and the discontinuous surface tension forces are not approximated in exactly the same way, this equilibrium may be perturbed and spurious velocities may occur near the droplet interface. Several authors have highlighted this phenomena, like Popinet and Zaleski [58] and Francois et al. [59].

The setup is the same as in [59,60] for the 3D bubble. The domain is a cube of side size $C = 8$. The bubble has a diameter of $d = 4.0$ and is located in the center of the domain. There is no viscosity effect and the density ratio is set to 0.1 between the two fluids, the bubble having the highest density. A surface tension of $\gamma = 73$ is prescribed.

Here we show how the mesh refinement can be used to decrease the parasitic currents. Three levels of mesh refinements are presented, each one with an isotropic and an anisotropic refinement. The number of elements is kept approximately constant between corresponding isotropic and anisotropic computations by applying a factor of $3^{1/3}$ to $h_0 (h_0^{\text{aniso}} = 3^{1/3} h_0^{\text{iso}})$ since a ratio $R_{\mathbf{n}}$ of 3 is used. Table 3 shows the maximum velocity $U^{\max}$ at $t = \Delta t$ and $t = 50\Delta t$, with $\Delta t = 0.001$. The results are compared with the best results of Francois et al. in [59] and Williams et al. in [60] obtained with $h = \frac{C}{40}$. We can observe that the velocities decrease with the size of the elements, which is a common result in the literature. An interesting result is that the anisotropic computations lead to smaller spurious velocities.

The number of iterations required by the GMRES solver at each time step increases slightly when an aspect ratio is applied, as shown in Table 4.

### 4.3. 2D dam break

We consider a column of water maintained by a wall (a dam) that is impulsively removed at $t = 0$. This test is the basis of many works which try to predict the effects of a dam break. This test case has been widely

Table 2
Stationary test: number of GMRES iteration performed per time step for the isotropic and anisotropic ($R_{\mathbf{n}} = 4$) computations

| $h_0^{\text{iso}}$ | Isotropic | Anisotropic |
|---|---|---|
| 0.1 | 12 | 14 |
| 0.01 | 19 | 22 |
| 0.001 | 44 | 56 |

Table 3
Stationary spherical bubble case: maximum velocity $U^{\max}$ with several levels of refinement

| $h_0^{\text{iso}}$ | $t$ | Isotropic | Anisotropic | Francois et al. [59] | Williams et al. [60] |
|---|---|---|---|---|---|
| $C/20$ | $t = \Delta t$ | $9.19 \times 10^{-2}$ | $7.34 \times 10^{-2}$ | | |
| $C/20$ | $t = 50\Delta t$ | $9.26 \times 10^{-1}$ | $4.36 \times 10^{-1}$ | | |
| $C/40$ | $t = \Delta t$ | $6.80 \times 10^{-2}$ | $5.69 \times 10^{-2}$ | $4.02 \times 10^{-3}$ | $8.55 \times 10^{-2}$ |
| $C/40$ | $t = 50\Delta t$ | $3.77 \times 10^{-1}$ | $1.88 \times 10^{-1}$ | $4.02 \times 10^{-2}$ | $3.86 \times 10^{-1}$ |
| $C/80$ | $t = \Delta t$ | $5.73 \times 10^{-2}$ | $4.53 \times 10^{-2}$ | | |
| $C/80$ | $t = 50\Delta t$ | $2.59 \times 10^{-1}$ | $1.31 \times 10^{-1}$ | | |

The time step is $\Delta t = 10^{-3}$. The results are obtained for isotropic refinements and anisotropic refinements with a ratio of $R_{\mathbf{n}} = 3$ and the same numbers of nodes as in the corresponding isotropic meshes. The results are compared with the best results of Francois et al. in [59] and Williams et al. in [60] obtained with $h = \frac{C}{40}$.

Table 4
Stationary spherical bubble case: number of GMRES iterations required for three levels of refinements in isotropic and anisotropic computations

| $h_0^{iso}$ | Isotropic computation | Anisotropic computation |
|---|---|---|
| $C/20$ | 12 | 14 |
| $C/40$ | 18 | 21 |
| $C/80$ | 22 | 25 |

The mean volume of the elements in the dense zone is the same between the corresponding isotropic and anisotropic computations.

studied in the literature using experimental, theoretical and numerical approaches. As this test involves recombinations and strong deformations of the interface as well as a great unsteady character, it is also a good test case to validate our adaptive model.

The length of the domain is 6 and its height is 4. The water column is initially at the extreme left of the domain. The height and the width of the water column are 1. Slip conditions are applied on the bottom and the side walls, neglecting possible boundary layer effects. A zero-pressure condition is imposed on the upper boundary.

Two adaptive computations have been performed. Linear and quadratic elements have been used to discretize the level set function. Table 5 shows the two sets of parameters. The effects of surface tension are neglected because this flow is essentially governed by the gravity. With these parameters, the number of nodes ranges from 1380 to 3151 for the computation 1, from 1013 to 1848 for computation 2. Fig. 5 shows the interface and the mesh of the first computation at different times.

The history of the dimensionless horizontal displacement of the water front is shown in Fig. 6. The time is non-dimensionalized by $t = \sqrt{h_l/g}$ where $h_l$ is the height of the water column. The experimental results from Martin and Moyce [61] and the numerical results from Marchandise and Remacle [4] with a non-adaptive mesh are added to the diagram. The latter computation has been performed on a domain of height 1.5, with an unstructured mesh of 10,218 nodes and a mean edge length of about 0.04.

One can see that these results are in good agreement with the non-adaptive computations and the experiment from Martin and Moyce, except for the height of water in the second simulation, for which the number of nodes is not sufficient to avoid significant mass losses.

For the first computation, the computational time to reach the non-dimensional time 3 was 48 min, with about 24% spent in the adaptation process (involving projections). A non-adaptive computation with an edge length $h_{na} = 0.033$ has been performed with the same domain and fluids setup. The mesh was composed of 46,204 nodes and 230,418 elements. The time 3 was reached after 15.4 h.

The second computation was a bit slower than the first, with 85 min spent to reach time 3 (13.6% in adaptation process).

In order to highlight the efficiency of the anisotropic refinements compared to the isotropic refinements, the evolution of the front position computed with isotropic and anisotropic coarse refinements are presented in Fig. 7. Two levels of accuracy are proposed, each one being tested with the values 1 and 4 for the ratio $R_n$. The size $h_0$ is chosen in such a manner that the number of nodes is roughly the same for the corresponding isotropic and anisotropic computations. The parameters of the meshes are mentioned in Table 6. In order to emphasize the differences between the results, the meshes are relatively coarse which leads to big errors.

Table 5
2D dam break test: parameters of the two first computations

| | $p$ | $h_0$ | $h_1$ | $h_2$ | $\delta$ | $d_1$ | $d_2$ | $R_{\mathbf{n}}$ | $R_{\mathbf{t},\max}$ | $\kappa_{\max}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Computation 1 | 1 | 0.1 | 0.2 | 0.4 | 0.1 | 0.3 | 0.6 | 3.0 | 2.0 | 10.0 |
| Computation 2 | 2 | 0.12 | 0.3 | 0.6 | 0.08 | 0.3 | 0.6 | 3.0 | 2.0 | 10.0 |

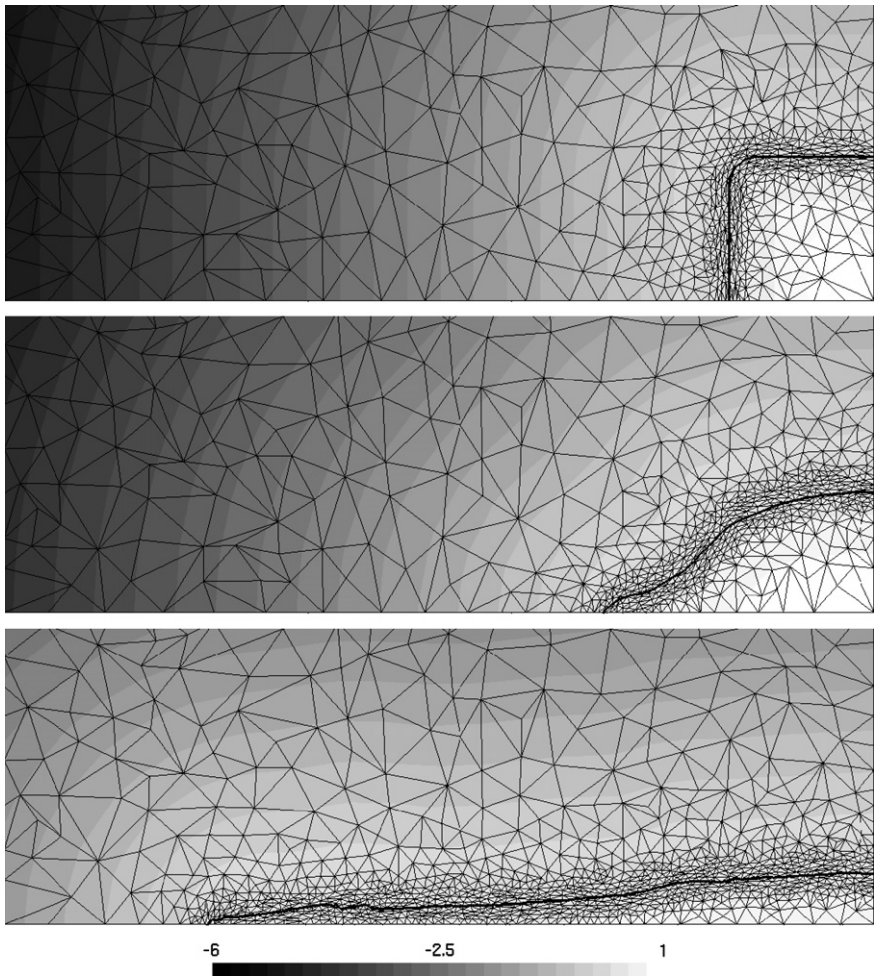
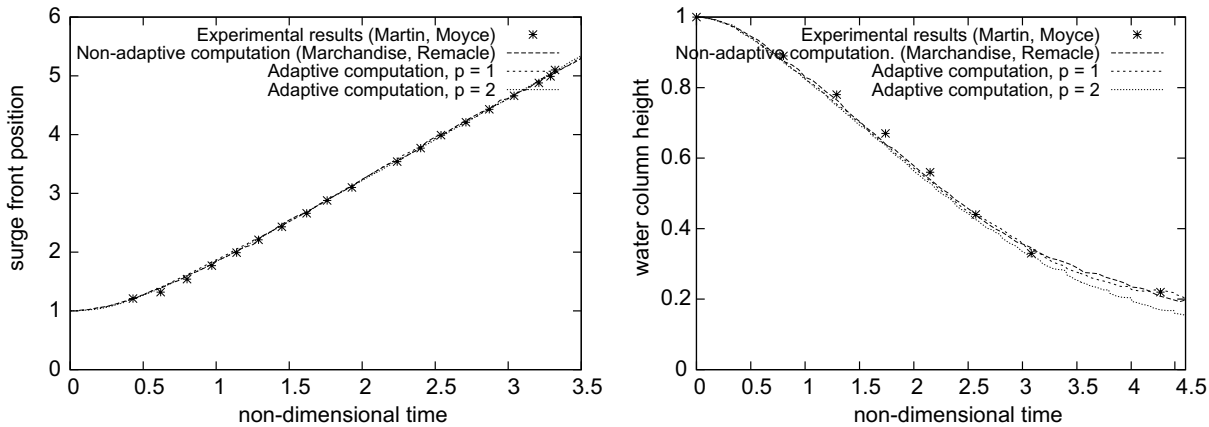
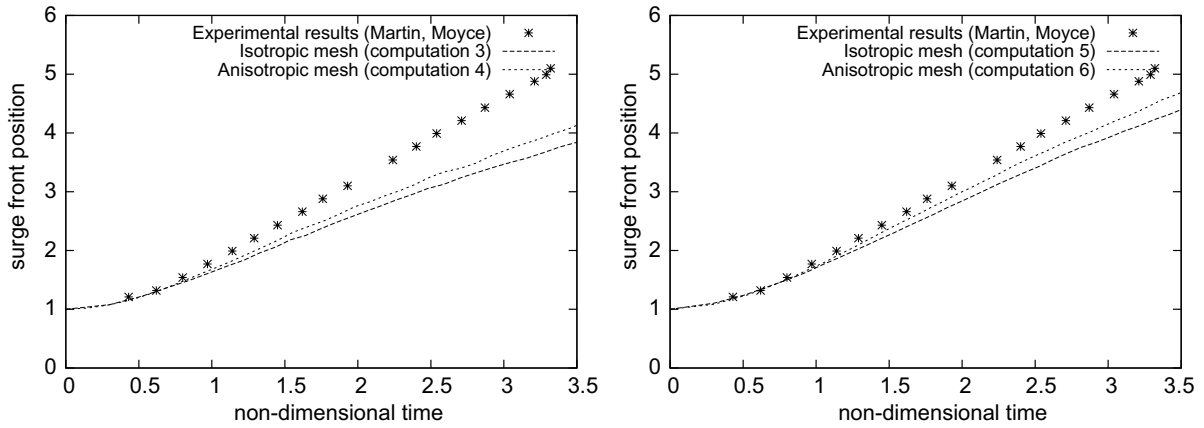



Fig. 5. Dam break problem: visualization of the meshes and the level set fields at non-dimensional times 0, 1 and 3 for the first computation. The upper part of the domain is not represented.




(*a*) Surge front position (*b*) Remaining water column height

Fig. 6. Dam break problem: comparison between experimental results, non-adaptive method and present method.




(a) Computations 3 and 4 (b) Computations 5 and 6

Fig. 7. Dam break problem: comparison between isotropic and anisotropic coarse refinements.

Table 6
2D dam break test: parameters of the computations 3 to 6

| | $p$ | $h_0$ | $h_1$ | $h_2$ | $\delta$ | $d_1$ | $d_2$ | $R_{\mathbf{n}}$ | $R_{\mathbf{t},\max}$ | $\kappa_{\max}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Computation 3 | 1 | 0.15 | 0.4 | 0.4 | 0.2 | 0.3 | 0.6 | 1.0 | 1.0 | 10.0 |
| Computation 4 | 1 | 0.3 | 0.4 | 0.4 | 0.2 | 0.3 | 0.6 | 4.0 | 2.0 | 10.0 |
| Computation 5 | 1 | 0.1 | 0.4 | 0.4 | 0.2 | 0.3 | 0.6 | 1.0 | 1.0 | 10.0 |
| Computation 6 | 1 | 0.2 | 0.4 | 0.4 | 0.2 | 0.3 | 0.6 | 4.0 | 2.0 | 10.0 |

### 4.4. 3D dam break with a square pile

This test case is still a dam break, but the geometry of the problem is 3D. A column of water collapses under gravity and the path of the wave crosses a square pile, leading to the development of a complex three-dimensional highly transient flow. The complexity of the shape of the interface and its variability is space and time makes the adaptive approach highly competitive.

The geometrical setting is the following. The height and the length of the initial water mass are 1. Its width is 1.6. The distance to the pile is 1.35. The pile has a square base of 0.3 and is shifted of 0.1 to the left. The back wall is located 3 far from the water. Slipping conditions are applied on all the walls. A zero-pressure condition is applied on the upper face of the domain, which is located 5 above the bottom wall.

Two computations were performed, with the parameters of Table 7, a Reynolds number of 40,000, a Froude number of 0.3193 and no surface tension. Note that the Reynolds number is purely indicative because boundary layers are not captured in our simulation.

With these settings, the number of nodes ranges from 5202 to 29,350 for the first computation and from 3361 to 16,185 for the second. The computational times were, respectively, 33.9 h (21.4% in the adaptation process) and 51.0 h (12.6% adapt.) hours to reach the non-dimensional time 2.5. In order to build fixed meshes

Table 7
3D dam break case: parameters of the computations

| | $p$ | $h_0$ | $h_1$ | $h_2$ | $\delta$ | $d_1$ | $d_2$ | $R_{\mathbf{n}}$ | $R_{\mathbf{t},\max}$ | $\kappa_{\max}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Computation 1 | 1 | 0.15 | 0.4 | 1.0 | 0.045 | 0.4 | 0.6 | 4.0 | 3.0 | 10.0 |
| Computation 2 | 2 | 0.2 | 0.4 | 1.0 | 0.06 | 0.4 | 0.6 | 4.0 | 3.0 | 10.0 |

with the same accuracy according to criterion (13), the number of nodes would have raised to a number of about 600,000 in the first simulation, and to about 250,000 in the second simulation. Fig. 8 shows the interface with the meshes of some boundaries at different times, while Fig. 9 represents a cut in the tetrahedral mesh at time 1.59.

### 4.5. Axisymmetric bubbles coalescence

In this test case, two air bubbles are immersed in a volume of water. The first one has a bigger radius and is initially located above the second. The bubbles rise under the effect of gravity, but the second rises faster because of the depression created by the first. Eventually, the bubbles merge into a single non-spherical bubble. The effects of surface tension are not taken into account.

The computational domain has a length of 1 in $x$- and $y$-directions, and 2.3 in $z$. The radius are respectively 0.15 and 0.10 for the first (above) and the second bubble. The bubbles are at a distance of 0.05. The gravity

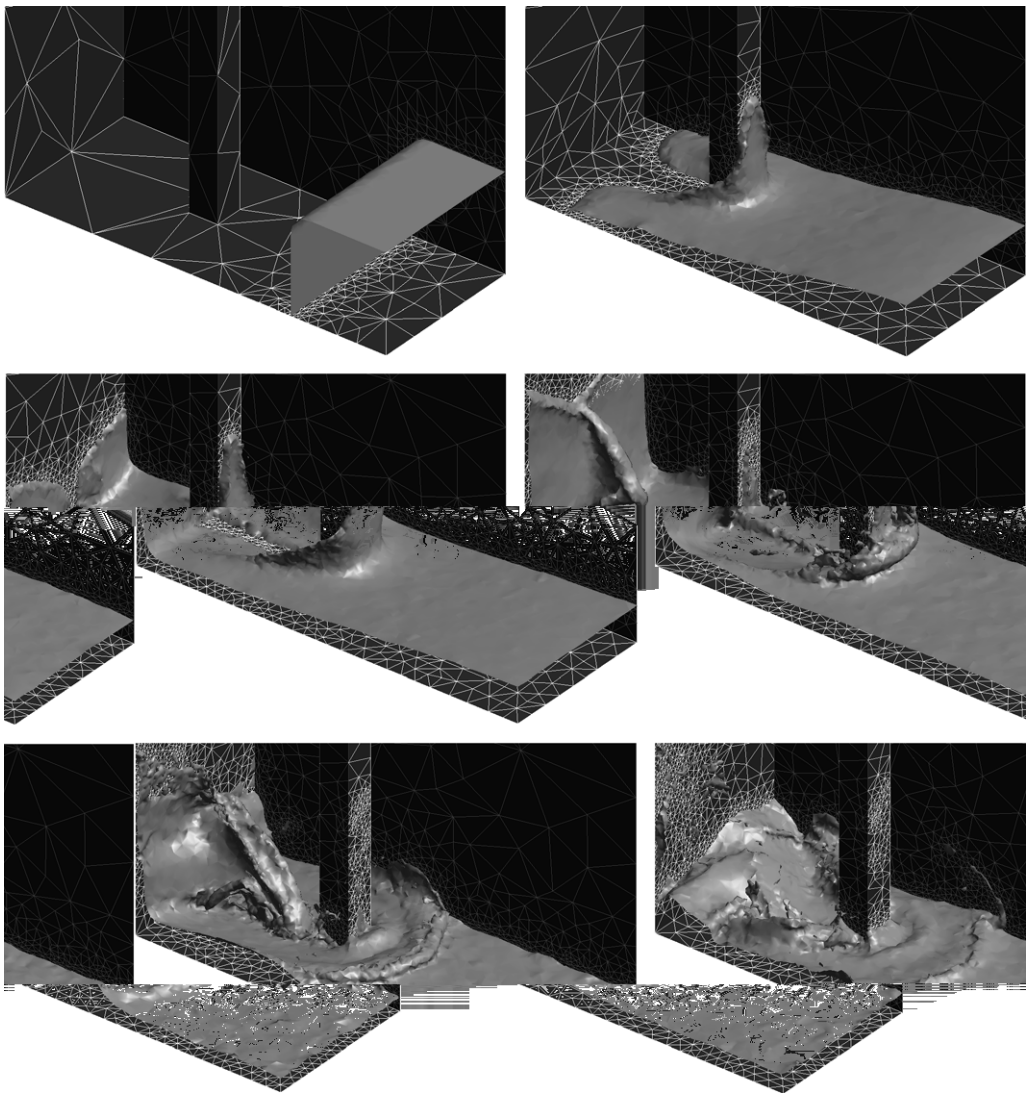


Fig. 8. 3D dam break case: interface and surface meshes at times 0, 0.88, 1.17, 1.59, 2.0 and 2.39.

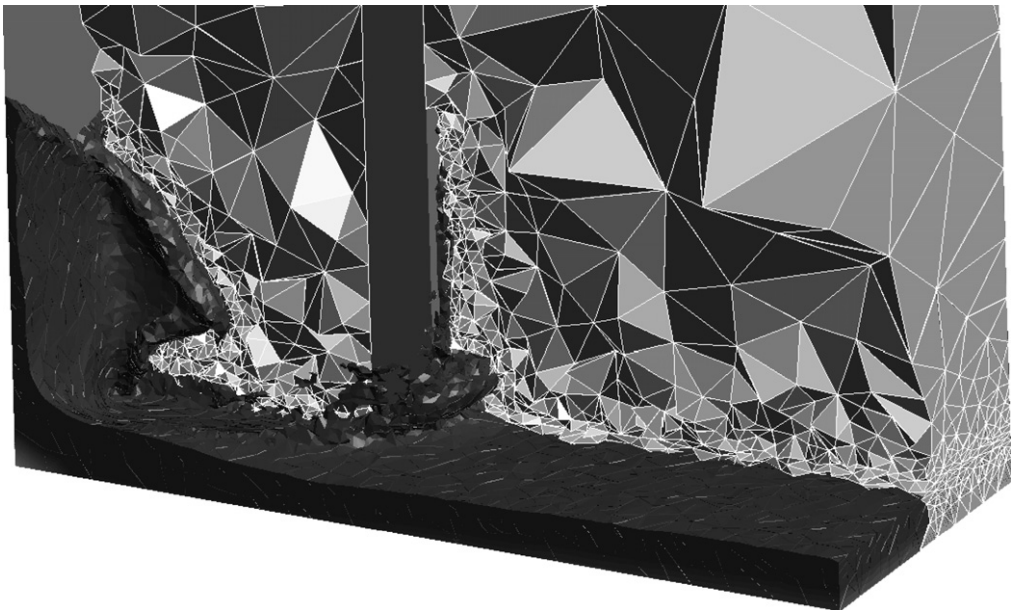


Fig. 9. 3D dam break case: volume of water and tetrahedral mesh at time 1.59.

acts in the $z$-direction. Slipping conditions are applied on all the boundaries. The Reynolds number and the Froude number are the following: $Re = 200$ and $Fr = 1$.

First order elements are used to represent the level set function. The meshes have the characteristics given by Table 8. The meshes produced have a number of nodes ranging from 16,480 to 24,922.

Fig. 10 shows the interface and the meshes at different times. We were able to obtain very smooth interfaces thanks to the mesh adaptation.

It took 17.7 h to run this computation. About 19% of the time was spent in the adaptation process. A non-adaptive computation with a similar accuracy, according to (13), would have needed a mesh size $h_{na} = 0.01$. With this very small mesh size, the mesh would have contained more than 2,300,000 nodes.

### 4.6. Oblique bubbles coalescence

For this test, two spherical bubbles of the same radius are immersed in the fluid with a shift between their horizontal positions. Both phases are initially at rest.

The domain is a parallelipipedron of size $[0,4]\times[0,4]\times[0,8]$. The center of the bubbles are located at positions $(2,2,2.15)$ and $(2.5,2,1)$. Both bubbles have an initial radius $R = 0.5$. The following non-dimensional values are set: $Re = 18.8$, $We = 50$, $Fr = 1$, $\rho_1/\rho_2 = 1/20$ and $\mu_1/\mu_2 = 1/26$. The computation has been made with quadratic shape functions for the level set.

Table 9 summarizes the meshes parameters that have been chosen.

Fig. 11 shows the evolution of the bubbles. The results compare well with those obtained by Marchandise et al. [5] without mesh adaptation and a grid of $30\times30\times60$. They also compare well with those obtained by

Table 8
Axisymmetric bubbles coalescence: parameters of the meshes

| $h_0$ | $h_1$ | $h_2$ | $\delta$ | $d_1$ | $d_2$ | $R_{\mathbf{n}}$ | $R_{\mathbf{t},\max}$ | $\kappa_{\max}$ |
|---|---|---|---|---|---|---|---|---|
| 0.03 | 0.15 | 0.40 | 0.02 | 0.02 | 0.15 | 3.0 | 2.0 | 0.05 |

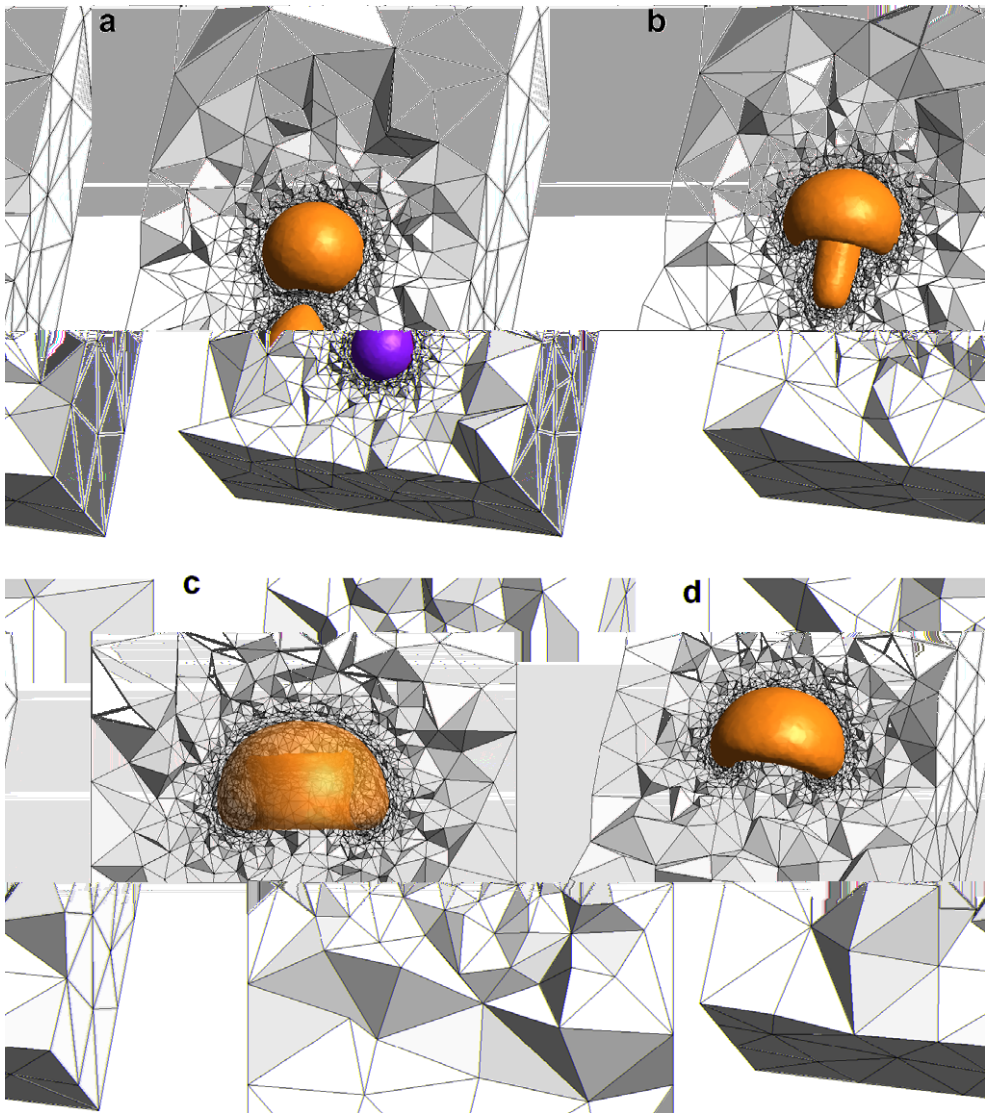



Fig. 10. Axisymmetric bubbles coalescence. $Re = 200$, $Fr = 1$, $\rho_1/\rho_2 = 1/1000$ and $\mu_1/\mu_2 = 1/100$. Interface and meshes at non-dimensional times (a) 0.2, (b) 0.56 and (c and d) 1.

Table 9
Oblique bubbles coalescence: parameters of the meshes

| $h_0$ | $h_1$ | $h_2$ | $\delta$ | $d_1$ | $d_2$ | $R_{\mathbf{n}}$ | $R_{\mathbf{t},\max}$ | $\kappa_{\max}$ |
|---|---|---|---|---|---|---|---|---|
| 0.4 | 0.4 | 2.0 | 0.15 | 0.3 | 2.0 | 4.0 | 3.0 | 0.2 |

Sussman and Puckett [62] with a $64 \times 64 \times 128$ grid, those of de Sousa et al. [63] and the experimental results obtained by Narayanan and Goossens [64]. Whereas 54,000 nodes were necessary in [5], all meshes produced with our computation comprised less than 6000 nodes, leading to a computation of 7 h (about 12% spent for mesh adaptation) to reach $t = 3$.

We can see from Fig. 11 that the anisotropic refined mesh combined with quadratic elements for the level set were able to represent the smallest scales of the interface with a very good accuracy.

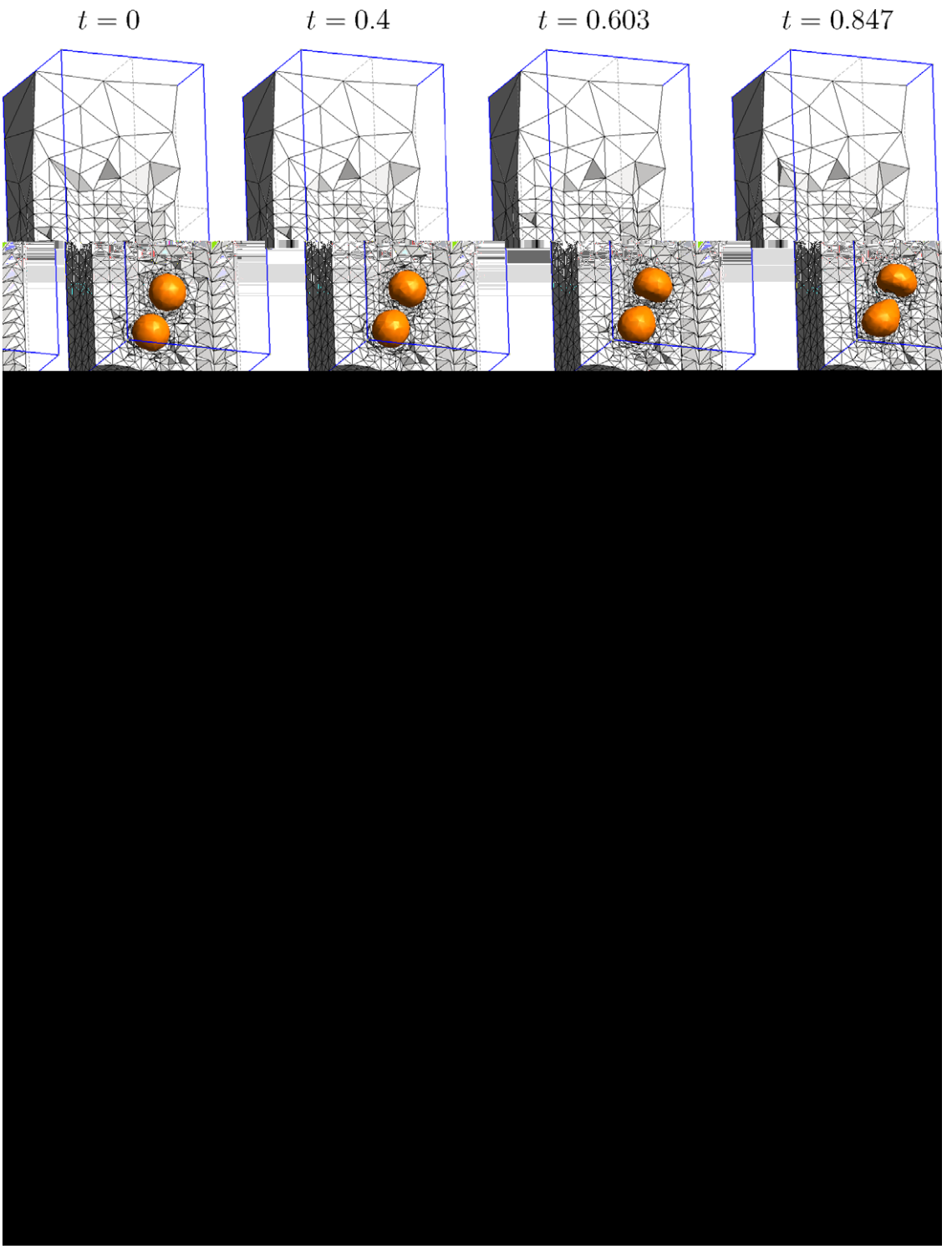



Fig. 11. Oblique bubbles coalescence. $Re = 18.8$, $We = 50$, $Fr = 1$, $\rho_1/\rho_2 = 1/20$ and $\mu_1/\mu_2 = 1/26$. Position of the interface at different non-dimensional times ranging from 0 to 2.89. A view of the mesh is superimposed to the view of the interface.

## 5. Conclusion

An adaptive procedure to improve the efficiency of the numerical method based on the model described by Marchandise and Remacle in [4] have been presented.

This model was already able to simulate highly transient flows with high ratios of densities and viscosities, with the advantages that it simply uses first order elements to represent pressure and velocity variables, and that it relies on a level set method for the interface, which is very flexible. Numerous applications are also in the scope of the model as it is able to represent the effects of surface tension in an accurate way.

The CPU time to perform complex simulations has been dramatically reduced by the adaptive improvement, without undermining the accuracy. This enhancement allows us to handle more complex computations in which either small and large scales are solved.

In the future works, a parallel implementation will be carried out. The fluid solver is already able to perform parallel computations. We are now focusing on the parallelization of the adaptive process.

## References

[1] M. Sussman, S. Almgren, B. Bell, P. Colella, L. Howell, M. Welcome, An adaptive level set approach for incompressible two-phase flows, Journal of Computational Physics 148 (1999) 81–124.
[2] Y. Andrillon, Simulation d'ecoulements a surface libre par une methode de capture d'interface en formulation totalement couplee, Ph.D. thesis, Ecole centrale de Nantes et Universite de Nantes, 2004.
[3] 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.
[4] E. Marchandise, J.-F. Remacle, A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows, Journal of Computational Physics 219 (2006) 780–800.
[5] E. Marchandise, P. Geuzaine, N. Chevaugeon, J.-F. Remacle, A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics, Journal of Computational Physics 225 (2007) 949–974.
[6] 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.
[7] 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.
[8] C. Hirt, B. Nichols, Volume-of-fluid method (VOF) for the dynamics of free boundaries, Journal of Computational Physics 39 (1981) 201–225.
[9] J. Pilliot, E. Puckett, Second order accurate volume-of-fluid algorithms for tracking material interfaces, Journal of Computational Physics 199 (2004) 465–502.
[10] 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 (2003) 110–136.
[11] C. Hirt, A. Amsden, J. Cook, An Arbitrary Lagrangian–Eulerian computing method for all flow speeds, Journal of Computational Physics 14 (1974) 227–253.
[12] 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.
[13] J. Donea, Arbitrary Lagrangian–Eulerian finite element methods, Computational Methods for Transient Analysis 1 (1983) 473–516.
[14] W. Dettmer, P. Saksono, D. Peric, On a finite element formulation for incompressible Newtonian fluid flows on moving domains in the presence of surface tension, Computer Methods in Applied Mechanics and Engineering 19 (2003) 659–668.
[15] F. Harlow, J. Welch, Volume tracking methods for interfacial flow calculations, Physics of Fluids 8 (1965) 21–82.
[16] 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.
[17] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-fluid flows, Computers and Fluids 27 (1998) 663–680.
[18] J. Chessa, T. Belytschko, An extended finite element method for two-phase fluids, Journal of Applied Mechanics (Transactions of the ASME) 70 (2003) 10–17.
[19] J. Chessa, T. Belytschko, An enriched finite element method and level sets for axisymmetric two-phase flow with surface tension, International Journal for Numerical Methods in Engineering 58 (2003) 2041–2064.
[20] C. Norman, M. Miksis, Non-linear dynamics of thin films and fluid interfaces, Physica D: Nonlinear Phenomena 209 (2005) 191–204.
[21] 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.
[22] M. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, Journal of Computational Physics 53 (1984) 484–512.
[23] M. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, Journal of Computational Physics 82 (1989) 64–84.

[24] R. Pember, J. Bell, P. Colella, W. Crutchfield, M. Welcome, An adaptive cartesian grid method for unsteady compressible flow in irregular regions, Journal of Computational Physics 120 (1995) 278–304.
[25] D. Martin, P. Colella, A cell-centered adaptive projection method for the incompressible Euler equations, Journal of Computational Physics 163 (2000) 271–312.
[26] A. Roma, C. Peskin, M. Berger, An adaptive version of the immersed boundary method, Journal of Computational Physics 153 (1999) 509–534.
[27] M. Sussman, A. parallelized, adaptive algorithm for multiphase flows in general geometries, Computers and Structures 83 (2005) 435–444.
[28] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries, Journal of Computational Physics 190 (2003) 572–600.
[29] F. Losasso, F. Gibou, R. Fedkiw, Simulating water and smoke with an octree data structure, ACM Transactions on Graphics (SIGGRAPH Proc) 23 (2004) 457–462.
[30] D. Greaves, A quadtree adaptive method for simulating fluid flows with moving interfaces, Journal of Computational Physics 194 (2004) 35–56.
[31] V. Cristini, J. Blawzdziewicz, M. Loewenberg, An adaptive mesh algorithm for evolving surfaces: simulations of drop breakup and coalescence, Journal of Computational Physics 168 (2001) 445–463.
[32] X. Yang, A. James, J. Lowengrub, X. Zheng, V. Cristini, An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids, Journal of Computational Physics 217 (2006) 364–394.
[33] A. Anderson, X. Zheng, V. Cristini, Adaptive unstructured volume remeshing part i: The method, Journal of Computational Physics 208 (2005) 616–625.
[34] F. Losasso, R. Fedkiw, S. Osher, Spatially adaptive techniques for level set methods and incompressible flows, Computers and Fluids 35 (2006) 995–1010.
[35] J.-F. Remacle, X. Li, M. Shephard, J. Flaherty, Anisotropic adaptive simulation of transient flows using Discontinuous Galerkin methods, International Journal for Numerical Methods in Engineering 62 (2004) 899–923.
[36] J.-F. Remacle, S. Soares Frazao, X. Li, M. Shephard, Adaptive discontinuous Galerkin method for the shallow water equations, International Journal for Numerical Methods in Fluids 52 (2006) 903–923.
[37] I. Babuska, The finite element method with Lagrangian multipliers, Numerische Mathematik 20 (1971) 179–192.
[38] T. 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).
[39] V. Selmin, L. Formaggia, Unified construction of finite element and finite volume discretizations for compressible flows, International Journal for Numerical Methods in Fluids 39 (1996) 1–32.
[40] P. Geuzaine, Newton–Krylov strategy for compressible turbulent flows on unstructured meshes, AIAA Journal 39 (2001) 528–531.
[41] 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.
[42] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), Journal of Computational Physics 152 (1999) 457–492.
[43] J. Brackbill, C. Kothe, D. Zemach, A continuum method for modeling surface tension, Journal of Computational Physics 2 (1992) 335–354.
[44] A. Smoliansky, Numerical modeling of two-fluid interfacial flows, Ph.D. thesis, University of Jyvaskyla, Finland, 2001.
[45] A.-K. Tornberg, Interface tracking methods with applications to multiphase flows, Ph.D. thesis, Royal Institute of Technology (KTH), Finland, 2000.
[46] R. Almeida, P. Feijoo, A. Galeao, C. Padra, R. Silva, Adaptive finite element computational fluid dynamics using an anisotropic error estimator, Computer Methods in Applied Mechanics and Engineering 182 (3-4) (2002) 379–400.
[47] H. Borouchaki, P.L. George, F. Hecht, P. Laug, E. Saltel, Delaunay mesh generation governed by metric specifications – part i: Algorithms and part ii: Applications, Finite Elements in Analysis and Design 25 (1997) 61–83, 85–109.
[48] P. Saramito, N. Roquet, An adaptive finite element method for viscoplastic fluid flow in pipes, International Journal for Numerical Methods in Engineering 190 (2001) 5391–5412.
[49] F. Alauzet, P.L. George, B. Mohammadi, P. Frey, H. Borouchaki, Transient fixed point-based unstructured mesh adaptation, International Journal for Numerical Methods in Fluids 23 (2003) 729–745.
[50] X. Li, Mesh modification procedures for general 3-D non-manifold domains, Ph.D. thesis, Rensselear Polytechnic Institute, 2003.
[51] C. Dobrzynski, P.J. Frey, B. Mohammadi, O. Pironneau, Fast and accurate simulations of air-cooled structures, Computer Methods in Applied Mechanics and Engineering 195 (2006) 3168–3180.
[53] P.-E. Bernard, N. Chevaugeon, V. Legat, E. Deleersnijder, J.-F. Remacle, High-order *h*-adaptative discontinuous Galerkin methods for ocean modelling, Ocean Dynamics 57 (2007) 109–121.
[54] X. Li, M. Shephard, M. Beall, 3D anisotropic mesh adaptation by mesh modification, Computer Methods in Applied Mechanics and Engineering 194 (2005) 4915–4950.
[55] M. Mount, N. Arya, R. Netanyahu, R. Silverman, An optimal algorithm for Approximate Nearest Neighbor searching, Journal of the ACM 45 (1998) 891–923.
[56] ANN library. URL http://www.cs.umd.edu/~mount/ANN/.
[57] X. Zheng, J. Lowengrub, A. Anderson, V. Cristini, Adaptive unstructured volume remeshing ii: Application to two- and three-dimensional level-set simulations of multiphase flow, Journal of Computational Physics 208 (2005) 626–650.

[58] 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.
[59] M. Francois, S. Cummins, E. Dendy, D. 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 (2006) 141–173.
[60] M. Williams, D. Kothe, E. Puckett, Convergence and accuracy of continuum surface tension models, in: W. Shyy, R. Narayanan (Eds.), Fluid Dynamics at Interface, Cambridge University Press, Cambridge, 1999, pp. 294–305.
[61] M. Quecedo, M. Pastor, M. Herreros, An experimental study of the collapse of liquid columns on a rigid horizontal plane, Philosophical Transactions A 244 (1952) 312–324.
[62] 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.
[63] F. de Sousa, N. Mangiavacchi, L. Nonato, A. Castel, M. Tome, V. Ferreira, J. 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.
[64] N.K.S. Narayanan, H. Goossens, Coalescence of two bubbles rising in line at low reynolds number, Chemical Engineering Science 29 (1974) 2071–2082.