Received: 9 October 2016 | Revised: 3 January 2018 | Accepted: 4 February 2018

DOI: 10.1002/fld.4502

RESEARCH ARTICLE

WILEY

An adaptive numerical scheme for solving incompressible 2-phase and free-surface flows

P. Frey[1,2] | D. Kazerani[1,2] | T.T.M. Ta[1,2]

[1]Sorbonne Universités, Université Pierre et Marie Curie, Laboratoire Jacques-Louis Lions, Paris, France

[2]Centre National de la Recherche Scientifique, Laboratoire Jacques-Louis Lions, Paris, France

Correspondence
P. Frey, Sorbonne Universités, Université Pierre et Marie Curie 06, UMR 7598, Laboratoire Jacques-Louis Lions, 75005 Paris, France; or Centre National de la Recherche Scientifique, UMR 7598, Laboratoire Jacques-Louis Lions, 75005 Paris, France.
Email: pascal.frey@upmc.fr

Dena Kazerani, Sorbonne Universités, Université Pierre et Marie Curie 06, UMR 7598, Laboratoire Jacques-Louis Lions, 75005 Paris, France; or Centre National de la Recherche Scientifique, UMR 7598, Laboratoire Jacques-Louis Lions, 75005 Paris, France.
Email: dena.kazerani@inria.fr

Summary

In this paper, we present a numerical scheme for solving 2-phase or free-surface flows. Here, the interface/free surface is modeled using the level-set formulation, and the underlying mesh is adapted at each iteration of the flow solver. This adaptation allows us to obtain a precise approximation for the interface/free-surface location. In addition, it enables us to solve the time-discretized fluid equation only in the fluid domain in the case of free-surface problems. Fluids here are considered incompressible. Therefore, their motion is described by the incompressible Navier-Stokes equation, which is temporally discretized using the method of characteristics and is solved at each time iteration by a first-order Lagrange-Galerkin method. The level-set function representing the interface/free surface satisfies an advection equation that is also solved using the method of characteristics. The algorithm is completed by some intermediate steps like the construction of a convenient initial level-set function (redistancing) as well as the construction of a convenient flow for the level-set advection equation. Numerical results are presented for both bifluid and free-surface problems.

KEYWORDS

anisotropic mesh adaptation, finite element method, free-surface flows, incompressible Navier-Stokes equations, level-set method, method of characteristics, 2-phase flows

1 | INTRODUCTION

Over the last decades, tremendous progress has been achieved on the development and analysis of numerical methods for 1-phase incompressible Stokes and Navier-Stokes flows, as emphasized by the vast literature on this topic. Open-source and commercial software packages are now readily available and can be used as black-box solvers for a large class of industrial problems. However, challenging topics still require further investigation to reach the same level of maturity. For instance, the work that has been done on numerical methods for incompressible Navier-Stokes equations for 1 fluid is already a good starting point for dealing with 2 fluids (or 2-phase flow) as well as free-surface flow problems. In fact, research on these topics has started in the last few years. However, there are several specific issues relevant to 2-fluid and free-surface flow problems that are not present in 1-phase or 1-fluid incompressible flow problems. In this context, the numerical treatment of the interface between 2 immiscible fluids, as well as the free surface, is certainly a difficult challenge that raises many related yet mostly unresolved problems: the coupling between the fluid dynamics and the interface (or free surface) evolution, the conservation of mass, the treatment of singularities (geometrical and

topological), and the approximation of surface tension forces. Not surprisingly, given the level of difficulty of mathematical analysis, most research results in these topics have been published in the engineering literature (see, nevertheless, the noticeable monograph by Gross and Reusken[1] on 2-phase incompressible flows and the works of Anderson et al[2] or Tryggvason et al[3] for an overview of numerical methods for the simulation of multiphase flows). Let us also mention that in most applications especially oceanography, free-surface flows are modeled by a shallow water approximation of incompressible Navier-Stokes or free-surface Euler equations (see, for instance, the works of Saint-Venant,[4] Green and Naghdi,[5] Gerbeau and Perthame,[6] and Lannes[7]). These models are simpler to analyze. Nevertheless, they are approximative and are valid only in the shallow water regime. In this work, we try to approach the free-surface problem by solving the free-surface Navier-Stokes equation by adopting the algorithm we propose for the 2-fluid problem. In other words, no shallow water model is considered here. Nevertheless, as done in some test cases, the comparison between the numerical simulation of the free-surface Navier-Stokes equation and the analytical solution of shallow water equations is an interesting issue.

Well-posedness results for the general weak formulation of the Navier-Stokes problem for 2-phase flows including the interface condition $V_\Gamma = \mathbf{u} \cdot \mathbf{n}$ (see Section 2) have been analyzed only for special cases, eg, when the 2-fluid domain is unbounded ($\Omega = \mathbb{R}^3$) with $\lim_{|x|\to\infty} \mathbf{u}(x,t) = 0$, when the initial interface $\Gamma(0)$ is a closed manifold,[8] or when it is close to a half-plane,[9] or under suitable conditions on the viscosity.[10] The case of a bounded domain Ω for arbitrary time intervals $[0, T]$, $T > 0$ is treated in the work of Tanaka[11]; it provides a well-posedness result for the Navier-Stokes problem in a weak formulation. Similarly, the well-posedness of the free-surface Navier-Stokes problem has been treated in the works of Allain,[12,13] Tani and Tanaka,[14] Tani,[15] and Xu et al.[16] Most analyses apply to cases with sufficiently smooth data and do not apply when the regularity of the interface drops down, like when bubbles collide, for example. In such cases, curvature is no longer well defined, and weak alternatives have to be considered. These alternatives involve different representations of the interface, which, in turn, induce relevant numerical techniques for the simulation of 2-fluid flows.

1.1 | Interface/free-surface representation

In this section, we briefly recall the 2 most important methods for representing the interface (or the free surface) $\Gamma(t)$. Broadly speaking, these methods can be classified as Lagrangian ordinary differential equation (ODE) techniques and Eulerian partial differential equation techniques. Authors often introduce the terminology *interface tracking* versus *interface capturing* to characterize the treatment of the interface. In the remainder of this paper, we will restrict our numerical investigation to only one of these, ie, the *level set* representation.

1.1.1 | Interface tracking

When the interface regularity is sufficient, then normal, curvature, and immiscibility conditions are well defined. Given a velocity field $\mathbf{u} \in V$, where V is a suitable functional space, the trace $\mathbf{u}|_\Gamma$ is well defined, and the interface evolution can be described in Lagrangian coordinates. Each and every infinitely small particle in the domain Ω is transported (advected) by the flow field $\mathbf{u}(\mathbf{x}, t)$, and we define the characteristic $\mathbf{X}(t) = \mathbf{X}(\mathbf{x_0}, t_0; t)$ as the path of this particle with initial position $\mathbf{x_0}$. This trajectory is described by the following set of ODEs:

$$\begin{cases} \frac{\mathrm{d}\mathbf{X}(\mathbf{x_0}, t_0; t)}{\mathrm{d}t} = \mathbf{u}\left(\mathbf{X}(\mathbf{x_0}, t_0; t), t\right), & t \geq 0 \\ \mathbf{X}(\mathbf{x_0}, t_0; t_0) = \mathbf{x_0}. \end{cases} \tag{1}$$

For $\mathbf{u}(\mathbf{x}, t)$ that is Lipschitz continuous (with respect to $\mathbf{x}$), this system has a unique solution, and the regularity of $\mathbf{X}$ is related to the regularity of $\mathbf{u}$. For $T > t_0$ that is sufficiently close to t_0, there is a one-to-one mapping between $\Omega \times [t_0, T]$ and Ω in the sense that for all $(\mathbf{x}, t) \in \Omega \times [t_0, T]$, there exists a unique $\mathbf{x_0} \in \Omega$ such that $\mathbf{x} = \mathbf{X}(\mathbf{x_0}, t_0; t)$. Following the flow backwards in time starting from $(\mathbf{x}, t)$ yields the following equation:

$$\mathbf{x} = \mathbf{x_0} + \int_{t_0}^{T} \mathbf{u}\left(\mathbf{X}(\mathbf{x_0}, t_0; t)\right) \mathrm{d}t, \tag{2}$$

which represents a transformation from Eulerian to Lagrangian coordinates. The Navier-Stokes problem can be transformed accordingly into a nonstationary problem with a stationary interface $\Gamma(0)$.[11] Likewise, the evolution of the interface

$\Gamma(t)$ can be described by using the Lagrangian coordinates, and $\Gamma(t)$ is simply characterized as the set of $\mathbf{x}$ satisfying (2) for $\mathbf{x_0} \in \Gamma(0)$. This class of method is called *interface tracking*.

Practically, the Navier-Stokes problem is solved on a fixed grid or an unstructured mesh using a Eulerian approach, and a Lagrangian approach is used to solve the evolution of the interface. *Marker points* are equidistributed along the interface $\Gamma(t_0)$ at time t_0 and then advected by the flow field $\mathbf{u}$ over a time period Δt. Their final location marks the position of the interface at time $t_0 + \Delta t$. After several time steps, the equidistribution property is usually lost, and marker points have to be redistributed along the new interface $\Gamma(t_0 + \Delta t)$. Markers are usually connected to define a piecewise affine interpolation of the interface and can coincide with the set of vertices of a triangulation of $\Gamma(t_0)$. In addition, this method requires transfer of information between the interface and the fixed grid once the interface has moved. Obviously, this approach is not very well suited for dealing with topology changes or severe displacements (distortion) of the interface between 2 time steps. More details about this method and its implementation can be found in the works of Unverdi and Tryggvason,[17] Esmaeeli and Tryggvason,[18,19] and Hirt et al.[20]

To partially overcome these problems, hybrid approaches like the arbitrary Lagrangian-Eulerian method have been proposed and revealed to be especially efficient for solving a fluid-structure interaction.[21-24] They consist in solving the interface using a grid or a mesh, which is moved according to the flow velocity $\mathbf{u}$. The mesh velocity in the interior of the domain generally differs from the flow velocity field, in order to avoid strong distortions.

In volume-of-fluid (VOF) methods, the treatment of the interface is based on a weak formulation of the following advection equation:

$$\frac{\partial \chi_1}{\partial t} + \mathbf{u} \cdot \nabla \chi_1 = 0, \tag{3}$$

where $\chi_1(\cdot, t)$ is the characteristic function for the subdomain $\Omega_1(t)$ representing 1 of the 2 fluids if the system is bifluid and the fluid domain if the problem is free-surface. This domain is delimited by the border of the support of this characteristic function, ie, $\partial\Omega_1(t) = \partial \text{supp}(\chi_1(\cdot, t))$. The function χ_1 is discontinuous across the interface, and the transport equation requires a specific treatment. Given an arbitrarily small elementary VOF W and integrating leads to the following equation:

$$\frac{\partial \chi_1}{\partial t} \int_W \chi_1 \mathrm{d}x + \int_{\partial W} \chi_1 \mathbf{u} \cdot \mathbf{n} \mathrm{d}s = 0, \tag{4}$$

which can be interpreted as a weak formulation of (3) corresponding to volume conservation. The method typically involves 2 steps: first, the reconstruction of the interface (approximation of the characteristic function) and, second, the advection of the volume fraction function.

This VOF approach is widely used for the numerical simulation of 2-fluid and free-surface flows, mainly because it enjoys the good mass (volume) conservation property and can handle topology changes without difficulty.[25,26] The main drawbacks are 2-fold: VOF methods are tedious to implement on unstructured meshes and tend to lose accuracy. In addition, a Courant-Friedrichs-Lewy condition must be satisfied, which leads to severe limitations on the time step. Furthermore, obtaining accurate intrinsic geometric properties, such as normals, tangents, curvatures, and, hence, surface tension, reveals difficult in practice. Recent works focus on solving these problems and carry a lot of promises.[27-29]

1.1.2 | Interface capturing

As pointed out, in VOF methods, the discontinuous characteristic function χ_1 across the interface imposes a specific numerical treatment of the transport equation (3). An interesting alternative consists in introducing a continuous auxiliary function. The level-set method introduced by Dervieux and Thomasset[30] and Osher and Sethian[31] suggests the use of the signed distance function to the initial interface as auxiliary function ϕ, ie,

$$\phi(\mathbf{x}, t) \leq 0 \Longleftrightarrow \mathbf{x} \in \Omega_1(t), \quad \phi(\mathbf{x}, t) \geq 0 \Longleftrightarrow \mathbf{x} \in \Omega_2(t), \quad \phi(\mathbf{x}, t) = 0 \Longleftrightarrow \mathbf{x} \in \partial\Omega_1(t) \cap \partial\Omega_2(t).$$

Assuming that the velocity field $\mathbf{u}(\mathbf{x}, t)$ is sufficiently smooth, then for $t > 0$, the level-set values $\phi(\mathbf{x}, t)$ are defined by considering the values constant along characteristics, namely, by writing

$$\phi\left(\mathbf{X}(\mathbf{x}_0, t_0; t), t\right) = \phi(\mathbf{x}_0, 0), \qquad \mathbf{x_0} \in \Omega(0), t \geq 0, \tag{5}$$

and when we differentiate this equation with respect to t, it becomes

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \quad \text{for all } \mathbf{x} \in \Omega, t \geq 0. \tag{6}$$

This equation (similar to (3)) is well defined in its current formulation given that the velocity field $\mathbf{u}$ is Lipschitz continuous with respect to $\mathbf{x}$. Furthermore, the interface (or the free surface) $\Gamma(t)$ can be defined by the values of the auxiliary function ϕ at any time t, ie,

$$\Gamma(t) = \{\mathbf{x} \in \Omega; \quad \phi(\mathbf{x}, t) = 0\}. \tag{7}$$

There is no uniqueness of a solution for a general continuous velocity field $\mathbf{u}$ in this strong formulation. However, the notion of *viscosity solutions* of transport equations with a continuous velocity field eventually applies here, which yields to sub- or supersolutions.[32]

The level set (6) not only is used for the mathematical analysis of well-posedness of 2-fluid flows but also has very attractive numerical features for representing and handling the interface $\Gamma(t)$. We consider the initialization of ϕ with the signed distance function $\phi_0 = d(\mathbf{x}, \Gamma(0))$ to the interface (Equation (7)) at $t = 0$. The velocity field is the result of the Navier-Stokes equation. Moreover, the transport equation (6) is discretized using suitable numerical methods in space and time (see Sections 2 and 3 for more details of our implementation). As the level-set function is continuous, its discretization is more accurate than that of the characteristic function considered in VOF methods. As the iterations in time increase, so does the discrepancy between the numerical solution $\phi_h(\mathbf{x}, t)$ and the signed distance function. A reinitialization of the level-set function is then carried out when, for example, $\|\nabla\phi_h(\mathbf{x}, t)\|_2$ exceeds some given tolerance value.

Due to its simplicity and its ability to efficiently deal with topology changes, the level-set method has been extensively used in engineering applications. Additionally, the extension from 2 dimensions to 3 dimensions can be achieved easily. Like the arbitrary Lagrangian-Eulerian method described above, variants of the level-set method can be considered, in which the interface is explicitly discretized (hence the terminology *interface capturing*) by a mesh, and this mesh is moved with the flow velocity. As will be seen, the intrinsic properties of the interface can be accurately computed using the level-set function.[33]

1.2 | Proposed approach

In this paper, we describe in detail our work and propose a general strategy for solving 2-phase/free-surface flows, which takes advantage of the flexibility of the level-set method for capturing and tracking the evolution of the interfaces, including topological changes, and enjoys an exact and accurate description of the interface using a conforming unstructured mesh. The idea of combining an implicit method for dealing with the domain evolution and an explicit representation of the manifold separating the 2 fluids is obviously not new. The numerical resolution of incompressible Navier-Stokes equations using a Lagrange-Galerkin scheme was first introduced by Benqué et al[34] and studied in the work of Pironneau.[35] Nevertheless, unlike similar approaches that solve the flow problem and the transport equation on Cartesian grids,[27,36-38] our approach relies on an adaptive unstructured mesh to carry out these computations. This allows us to solve the time-discretized Navier-Stokes equation in the fluid domain only, in the case of free-surface problems.

In the work of Bui et al,[39] the authors proposed a former version of this method in 2 dimensions, which required much more computational effort in resolving the problem at each time step and is proved to be difficult getting extended to 3 dimensions. Our method has several things in common with this work: a computational domain Ω is defined and discretized with an unstructured mesh that is modified at each time iteration, in such a way that the interface/free surface $\Gamma(t)$ is explicitly discretized in the mesh. Nevertheless, Navier-Stokes equations and the transport equation are solved using the same mesh of the domain. In other words, our strategy has several assets: on the one hand, no projection is needed between different meshes; on the other hand, the method does not present any theoretical difficulty in terms of extension from the 2-dimensional (2D) case to the 3-dimensional (3D) case. This is a tremendous feature insofar as mesh adaptation is concerned, since mesh generation is known to be more difficult to deal with in 3 dimensions. Most of the difficulty related to meshing is recast as a robust *remeshing* problem.

This paper is organized in 5 sections. The next section briefly presents the mathematical model and the initial, boundary, and interfacial conditions. Section 3 is the central section of this paper and presents numerical tools used for the global algorithm. In particular, 2 variants of the method of characteristics used to resolve the material derivative of the Navier-Stokes problem, corresponding to nonlinear convection, and the advection of the interface, are presented. Velocity extension/regularization, mesh adaptation, and redistancing of the level-set function are other tools described in this section. Then, the general scheme is summarized in Section 4, and several numerical examples and test cases are presented in Section 5.

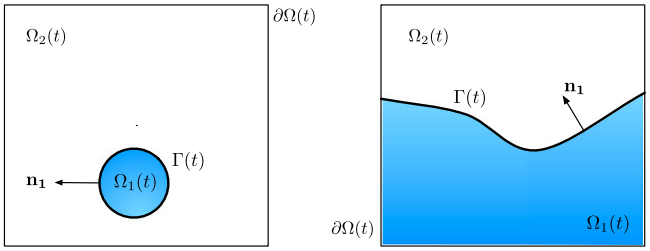

FIGURE 1 Illustration of a computational domain $\Omega(t)$ composed of 2 subdomains $\Omega_1(t)$ and $\Omega_2(t)$. These subdomains are separated by an interface $\Gamma(t)$, which may evolve in time [Colour figure can be viewed at wileyonlinelibrary.com]

2 | A MODEL FOR 2-PHASE/FREE-SURFACE FLOW PROBLEM

In this paper, we are interested in 2-phase or free-surface flows, that is, we consider a bounded open set $\Omega(t) \subset \mathbb{R}^d$, with at least Lipschitz regularity. This set is filled with 2 different immiscible incompressible phases (ie, liquid-liquid or liquid-gas) if the system is bifluid. Each phase has different material properties, density ρ_i, and dynamic viscosity μ_i, $(i = 1, 2)$. We suppose that both phases are chemically uniform, physically distinct, and separable, and we assume isothermal conditions. Therefore, at each time step $t \in [0, T]$, the computational domain $\Omega(t)$ is partitioned into 2 open subdomains $\Omega_1(t)$ and $\Omega_2(t)$. Each subdomain contains, at most, 1 phase such that $\overline{\Omega_1(t)} \cup \overline{\Omega_2(t)} = \bar{\Omega}$, $\Omega_1(t) \cap \Omega_2(t) = \emptyset$. The contact region between the 2 subdomains is called the *interface* and is denoted by $\Gamma(t) = \overline{\Omega_1(t)} \cap \overline{\Omega_2(t)}$ (cf Figure 1). The different phases as well as the interface may move in time. Otherwise, if the problem is free-surface, $\Omega_1(t)$ represents the fluid domain at time t, whereas $\Omega_2(t)$ represents the vacuum. In this case, we simply denote by ρ the density and by μ the dynamic viscosity of the fluid.

2.1 | A Navier-Stokes model for 2-phase flows

In each phase, the conservation laws for mass and momentum hold, and we consider separate Navier-Stokes equations in the 2 domains Ω_i, $i = 1, 2$, ie,

$$\begin{cases} \rho_i \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) = \mathrm{div}\sigma_i + \rho_i \mathbf{f} \\ \mathrm{div}\,\mathbf{u} = 0 \end{cases}, \quad \text{in } \Omega_i \times [0, T] (i = 1, 2), \tag{8}$$

where $\sigma_i = -p\mathbf{I} + 2\mu_i \mathbf{D(u)}$ is the stress tensor and $\mathbf{D(u)} = (\nabla \mathbf{u} + \nabla \mathbf{u}^t)/2$ is the deformation tensor, with $p(x, t)$ as the pressure, $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ as the velocity, and $\mathbf{f}$ as the density of an external volume force exerted on the phases, eg, the gravity. The constants $\mu_i > 0$ and ρ_i denote the dynamic viscosity and the density of the phases in the subdomains Ω_i, $i = 1, 2$, respectively.

The model needs to be endowed with suitable boundary and initial conditions for velocity $\mathbf{u}$ only. The initial condition is $\mathbf{u}(x, 0) = \mathbf{u}_0(\mathbf{x})$, where $\mathbf{u}_0(\mathbf{x})$ is a divergence-free function, which is relevant for the physical problem at hand. Boundary conditions are of 2 types: *natural* and *essential*. We assume that the outer boundary $\partial\Omega(t)$ remains unchanged in time, hence omitting the time dependency, and is partitioned into 3 possibly nonconnected parts $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N \cup \partial\Omega_S$, such that $\partial\Omega_D \cap \partial\Omega_N \cap \partial\Omega_S = \emptyset$. In applications, we consider conventionally as essential conditions the Dirichlet boundary conditions $\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_D(\mathbf{x}, t)$ for $\mathbf{x}$ on $\partial\Omega_D$ for a given function $\mathbf{u}_D$ and use these to prescribe the inflow velocity profile and conditions along the outer boundaries, eg, no-slip condition. On the other boundaries, we introduce natural boundary conditions that allow to prescribe the outflow and slip conditions. Neumann conditions are of the form

$$\sigma \mathbf{n}_{\partial\Omega_N} = \mathbf{u}_N \quad \text{or} \quad \sigma \mathbf{n}_{\partial\Omega_N} = -p_e \mathbf{n}_{\partial\Omega_N}, \tag{9}$$

where p_e denotes an external pressure function and $\mathbf{n}_{\partial\Omega_N}$ is the unit normal pointing outward the boundary $\partial\Omega_N$. Finally, slip conditions are prescribed by a combination of 2 conditions, as follows:

$$\mathbf{u} \cdot \mathbf{n}_{\partial\Omega_S} = 0 \quad \text{and} \quad \alpha \mathbf{u} \cdot \tau_{\partial\Omega_S} + \tau_{\partial\Omega_S} \cdot \sigma \mathbf{n}_{\partial\Omega_S} = 0. \tag{10}$$

Here, $\mathbf{n}_{\partial\Omega_S}$ is the unit outward normal vector to $\partial\Omega_S$, whereas $\tau_{\partial\Omega_S}$ is a unit tangent vector to $\partial\Omega_S$. In addition, the constant α is strictly positive and is called the friction coefficient.

In the weak formulation, the slip boundary conditions are interpreted by the integral term $\int_{\partial\Omega_S}\alpha(\mathbf{u}\cdot\tau_{\partial\Omega_S})\cdot(\mathbf{v}\cdot\tau_{\partial\Omega_S})ds$ on the left-hand side (see Section 3). Slip without friction corresponds to the case $\alpha = 0$, which then makes the previous integral term to vanish. The condition $\mathbf{u}\cdot\mathbf{n}_{\partial\Omega_S} = 0$ must be added to the functional space on which the weak formulation is defined. Nevertheless, from the numerical point of view, this function space can get relaxed from this constraint by using a penalization technique, ie, by adding the term $\int_{\partial\Omega_S}\mathcal{A}^*(\mathbf{u}.\mathbf{n}).(\mathbf{v}.\mathbf{n})ds$ on the left-hand side of the weak form, where $\mathcal{A}^*$ is a penalization coefficient (large about 10^6).

Next, model (8) is endowed with *coupling conditions* on the interface $\Gamma(t)$. We assume that there is no mass transfer and that no phase transition takes place; hence, the velocity is continuous across the interface, ie,

$$[\mathbf{u}]_{\Gamma(t)} = 0, \tag{11}$$

where the notation $[\cdot]_{\Gamma(t)}$ denotes the jump of a function across the interface in the direction of $\mathbf{n}_1$, which denotes the exterior normal to $\Omega_1(t)$ on the interface, ie, $[\cdot]_{\Gamma(t)} = \cdot|_{\Omega_1} - \cdot|_{\Omega_2}$. Due to the momentum conservation law, the balance of the normal stress with the surface tension across $\Gamma(t)$ reads

$$[\sigma]_{\Gamma(t)}\cdot\mathbf{n}_1 = -\gamma\kappa\mathbf{n}_1. \tag{12}$$

The scalar function $\kappa(\mathbf{x})$ is the algebraic mean curvature for $\mathbf{x}$ on $\Gamma(t)$, which is positive if the interface curve/surface bends toward Ω_1 and is negative otherwise. The parameter $\gamma > 0$ is the surface tension coefficient (ie, the magnitude of the surface tension force), which is a material property of the model and is assumed to be constant here for the sake of simplicity.

Finally, the immiscibility assumption leads to the introduction of the normal velocity $\mathbf{u}_\Gamma(t) = \mathbf{u}_{\Gamma(t)}(\mathbf{x}, t)$, which prescribes the *magnitude* of the velocity field of the interface $\Gamma(t)$ in the normal direction. This condition is meant to indicate that the normal velocity of the interface is equal to the normal component of the flow field along $\Gamma(t)$, ie,

$$\mathbf{u}_{\Gamma(t)} = \mathbf{u}\cdot\mathbf{n}_1. \tag{13}$$

Gathering the Navier-Stokes equation (8), the initial and boundary conditions (9)-(12), and the velocity condition on the interface (13) leads to the following model for 2-phase flows in each subdomain Ω_i, $i = 1, 2$:

$$\begin{cases} \rho_i\left(\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = \mathrm{div}\sigma_i + \rho_i\mathbf{f}, & \text{in} \quad \Omega_i(t)\times[0,T] \\ \mathrm{div}\,\mathbf{u} = 0, & \text{in} \quad \Omega_i\times[0,T] \\ \mathbf{u}(\mathbf{x},t) = \mathbf{u}_D(\mathbf{x},t), & \text{on} \quad \partial\Omega_D \\ \sigma\mathbf{n}_{\partial\Omega_N} = -p_e\mathbf{n}_{\partial\Omega_N}, & \text{on} \quad \partial\Omega_N \\ \mathbf{u}\cdot\mathbf{n}_{\partial\Omega_S} = 0 \quad \text{and} \quad \alpha\mathbf{u}\cdot\tau_{\partial\Omega_S} + \tau_{\partial\Omega_S}\cdot\sigma\mathbf{n}_{\partial\Omega_S} = 0, & \text{on} \quad \partial\Omega_S \\ \mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}), & \text{in} \quad \Omega_i \\ [\mathbf{u}]_{\Gamma(t)} = 0, \qquad [\sigma]_{\Gamma(t)}\cdot\mathbf{n}_1 = -\gamma\kappa\mathbf{n}_1, & \text{on} \quad \Gamma(t) \\ \mathbf{u}_{\Gamma(t)} = \mathbf{u}\cdot\mathbf{n}_1, & \text{on} \quad \Gamma(t). \end{cases} \tag{14}$$

Let us mention that by introducing 2 scalar functions μ and ρ defined on the whole computational domain Ω by

$$\mu = \chi_1\mu_1 + \chi_2\mu_2 \quad \text{and} \quad \rho = \chi_1\rho_1 + \chi_2\rho_2,$$

with χ_i as the characteristic function of the domain Ω_i, the first equation of (14) can be written as

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = \mu\Delta\mathbf{u} + \rho\mathbf{f}, \quad \text{in } \Omega_i(t)\times[0,T](i = 1, 2).$$

2.2 | A Navier-Stokes model for free-surface flows

Similarly to the previous subsection, the model we consider for free-surface flows is the incompressible Navier-Stokes with essential and natural boundary conditions. The main difference is that the Navier-Stokes equation is not considered

on the whole computational domain Ω but on the fluid domain $\Omega_1(t)$, which moves in time. This means that system (8) with boundary conditions writes

$$\begin{cases} \rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = \mathrm{div}\sigma + \rho\mathbf{f}, & \text{in} \quad \Omega_1(t)\times[0,T] \\ \mathrm{div}\,\mathbf{u} = 0, & \text{in} \quad \Omega_1(t)\times[0,T] \\ \mathbf{u}(\mathbf{x},t) = \mathbf{u}_D(\mathbf{x},t), & \text{on} \quad \partial\Omega_D \cap \partial\Omega_1(t) \\ \sigma\mathbf{n}_{\partial\Omega_N} = -p_e\mathbf{n}_{\partial\Omega_N}, & \text{on} \quad \partial\Omega_N \cap \partial\Omega_1(t) \\ \mathbf{u}\cdot\mathbf{n}_{\partial\Omega_S} = 0 \quad \text{and} \quad \alpha\mathbf{u}\cdot\tau_{\partial\Omega_S} + \tau_{\partial\Omega_S}\cdot\sigma\mathbf{n}_{\partial\Omega_S} = 0, & \text{on} \quad \partial\Omega_S \cap \partial\Omega_1(t) \\ \mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}), & \text{in} \quad \Omega_1. \end{cases} \tag{15}$$

Since no fluid is considered in part $\Omega_2(t)$, the continuity condition on the interface disappears. In other words, no constraint is considered on the values the velocity takes on the free surface. Nevertheless, the normal stress on the free surface is balanced with the surface tension and, eventually, the atmospheric pressure p^a. Again, the normal velocity of the interface is equal to the normal component of the flow along the free surface. Therefore, the free surface and the kinematic boundary condition are written as

$$\begin{aligned} \sigma\cdot\mathbf{n}_1 &= -\left(\gamma\kappa + p^a\right)\mathbf{n}_1, \quad \text{on} \quad \Gamma(t) \\ \mathbf{u}_{\Gamma(t)} &= \mathbf{u}\cdot\mathbf{n}_1, \quad \text{on} \quad \Gamma(t). \end{aligned}$$

Similarly to the bifluid problem, the position and the evolution of $\Gamma(t)$ are both unknown. However the coupling condition $\mathbf{u}_{\Gamma(t)} = \mathbf{u}\cdot\mathbf{n}_1$ determines its transport. Several approaches have been devised to represent and to approximate the interface. In this article, we consider an *interface capturing* method inspired by the works of Sethian,[33] Chang et al,[40] and Sussman et al.[41] This method is described in the following subsection.

2.3 | Level-set representation of the interface $\Gamma(t)$

The computational domain Ω is implicitly defined by a scalar level-set function $\phi : \Omega \to \mathbb{R}$, such that the following property holds:

$$\forall \mathbf{x}\in\Omega, \quad \begin{cases} \phi(\mathbf{x}) < 0, & \text{in } \Omega_1(t) \\ \phi(\mathbf{x}) = 0, & \text{on } \Gamma(t) \\ \phi(\mathbf{x}) > 0, & \text{in } \Omega_2(t), \end{cases} \tag{16}$$

and we set this level-set function to be the *signed distance function* $d_{\Gamma(t)}$ to the interface $\Gamma(t)$, ie, $\phi(\mathbf{x},t) = d_{\Gamma(t)}(\mathbf{x},\Gamma(t))$ with

$$\forall \mathbf{x}\in\mathbb{R}^d, \quad d_{\Gamma(t)}(\mathbf{x}) = \begin{cases} -d(\mathbf{x},\Gamma(t)), & \text{if } \mathbf{x}\in\Omega_1(t) \\ 0, & \text{if } \mathbf{x}\in\Gamma(t) \\ d(\mathbf{x},\Gamma(t)), & \text{if } \mathbf{x}\in\Omega_2(t), \end{cases} \tag{17}$$

where $d(\cdot,\Gamma(t))$ denotes the usual Euclidean distance function to $\Gamma(t)$. This signed distance function enjoys several properties. In particular, its gradient is of unit norm wherever it is defined, ie, $|\nabla d_{\Gamma(t)}| = 1$ almost everywhere on $\mathbb{R}^d$. It is well known that $d_{\Gamma(t)}$ can be obtained as the steady-state solution of the unsteady *Eikonal equation*[33,42]:

$$\begin{cases} \frac{\partial\phi}{\partial s}(\mathbf{x},t) + \mathrm{sgn}(\psi_0)(|\nabla\phi| - 1) = 0, & \text{for } (\mathbf{x},s)\in\mathbb{R}^d\times(0,\infty) \\ \phi(\mathbf{x},0) = \psi_0(\mathbf{x}), & \text{for } \mathbf{x}\in\mathbb{R}^d \end{cases} \tag{18}$$

with ψ_0 as any level-set function associated with Ω. A numerical scheme to compute the solution of this problem is devised in the work of Dapogny and Frey.[43] This scheme relies on an explicit formula[42] and that has been implemented as an iterative algorithm on unstructured meshes.

2.4 | Level-set formulation of 2-phase/free-surface flows

The evolution of the implicitly defined interface $\Gamma(t)$ is numerically carried out using the level-set method.[31] Suppose that the motion of $\Gamma(t)$ over a time period $[0, T]$ is driven by a velocity field $\tilde{\mathbf{u}} : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}^d$ and follows the advection equation, ie,

$$\frac{\partial \phi}{\partial t}(\mathbf{x}, t) + \tilde{\mathbf{u}}(\mathbf{x}, t) \cdot \nabla \phi(\mathbf{x}, t) = 0, \qquad \text{on } \mathbb{R}^d \times [0, T]. \tag{19}$$

Here, the normal velocity of the interface is equal to the normal component of the velocity of the flow field along $\Gamma(t)$. Following this formulation, the kinematic boundary condition

$$\mathbf{u}_{\Gamma(t)} = \tilde{\mathbf{u}} \cdot \mathbf{n}_1, \qquad \text{on } \Gamma(t),$$

is replaced by an advection equation with Dirichlet boundary conditions on the computational domain, ie,

$$\begin{cases} \frac{\partial \phi}{\partial t} + \tilde{\mathbf{u}} \cdot \nabla \phi = 0, & \text{on } \Omega \times [0, T], \\ \phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}), & \text{on } \Omega(0), \end{cases} \tag{20}$$

where ϕ_0 is a level-set function for the initial interface/free-surface position. As mentioned previously, from the numerical point of view, the level-set function ϕ is adjusted at each iteration of the algorithm in such a way that it remains close to the signed distance function. This step of the algorithm is called redistancing and is done by the algorithm suggested in the work of Dapogny and Frey.[43]

In the next section, we will present different numerical tools that we need to solve our global algorithm.

3 | NUMERICAL TOOLS

In this section, we present the numerical techniques necessary for understanding the global scheme in Section 4. Some of these numerical tools let us advect the interface/free surface, ie, characteristic methods for the resolution of the transport equation, velocity regularization, and redistancing, whereas some others let us solve the time-discretized equation on the fluid domain, ie, characteristic methods for the time discretization of the Navier-Stokes equation, mesh adaptation velocity extension, and finite element methods.

3.1 | Method of characteristics for the advection equation

The method of characteristics[34,44-46] is known to be very efficient in solving advection-diffusion problems, including the Navier-Stokes equations (see the works of Pironneau[35,47,48] for the mathematical and numerical method and its application to fluid mechanics problems).

Here, we consider the Cauchy problem for the advection equation (3.5): *given a velocity field* $\tilde{\mathbf{u}}(\mathbf{x}, t)$ *defined on* Ω, *find a scalar solution* $\phi(\mathbf{x}, t)$ *defined on* $\Omega \times [0, T]$ *to solve Equation* (20). Under proper hypotheses on the regularity and the growth of $\tilde{\mathbf{u}}$ and ϕ_0,[47] the unique C^1 solution ϕ of (20) is

$$\phi(\mathbf{x}, t) = \phi_0\left(\mathbf{X}(\mathbf{x}, t; 0)\right) \tag{21}$$

with $s \mapsto \mathbf{X}(\mathbf{x}, t; s)$ as the *characteristic curve* of $\tilde{\mathbf{u}}$ passing through $\mathbf{x}$ at time t and defined as the solution of the ODE system, ie,

$$\begin{cases} \frac{\mathrm{d}\mathbf{X}(\mathbf{x}, t; s)}{\mathrm{d}s} = \tilde{\mathbf{u}}\left(\mathbf{X}(\mathbf{x}, t; s), t\right), & \text{for } s \in \mathbb{R} \\ \mathbf{X}(\mathbf{x}, t; t) = \mathbf{x}. \end{cases} \tag{22}$$

This curve describes the trajectory of a particle at position $\mathbf{x}$ at time t and transported by the velocity field $\tilde{\mathbf{u}}$. The first equation of (20) implies that $\phi(\mathbf{x}, t)$ is constant along the characteristics $\mathbf{X}(\mathbf{x}, t; s)$. In the numerical setting of this work, we divide the time interval $[0, T]$ into a finite number of subintervals of the form $[t^{n-1}, t^n]$ with $t^n = t^{n-1} + \Delta t$, for a time step Δt. Then, we denote $\phi^0(\mathbf{x}) := \phi_0(\mathbf{x})$ and define ϕ^n, for all integer $n \in [1, \frac{T}{\Delta t}]$, as the solution of

$$\begin{cases} \frac{\partial \phi^n}{\partial t}(\mathbf{x}, t) + \tilde{\mathbf{u}}(\mathbf{x}, t) \cdot \nabla \phi^n(\mathbf{x}, t) = 0, & (\mathbf{x}, t) \in \Omega \times (t^{n-1}, t^n) \\ \phi^n(\mathbf{x}, t^{n-1}) = \phi^{n-1}(\mathbf{x}), & \forall \mathbf{x} \in \Omega. \end{cases} \tag{23}$$

Again, we use the characteristic method to solve (23). In other words, the solution ϕ^n of this equation is approximated by

$$\phi^n(\mathbf{x}) = \phi^{n-1}\left(\mathbf{X}(\mathbf{x}, t^n; t^{n-1})\right), \qquad \text{for all } \mathbf{x} \in \Omega, \tag{24}$$

where $\mathbf{X}(\mathbf{x}, t^n; t^{n-1})$ is the position at the time t^{n-1} of the characteristic line emerging from $\mathbf{x}$ at time t^n. This function is computed by

$$\begin{cases} \frac{\mathrm{d}\mathbf{X}(\mathbf{x},t^n;t)}{\mathrm{d}t} = \tilde{\mathbf{u}}(\mathbf{X}(\mathbf{x}, t^n; t), t), & \text{for all } (\mathbf{x}, t) \in \Omega \times (t^{n-1}, t^n) \\ \mathbf{X}(\mathbf{x}, t^n; t^n) = \mathbf{x}, & \text{for all } x \in \Omega. \end{cases} \tag{25}$$

Let us also note that the first equation of (25) is equivalent to

$$\mathbf{X}(\mathbf{x}, t^n; t) = \mathbf{x} - \int_t^{t^n} \tilde{\mathbf{u}}\left(\mathbf{X}(\mathbf{x}, t^n; t), t\right) \mathrm{d}t. \tag{26}$$

This integral can be approximated by any method used for solving ODEs. We use here a fourth-order Runge-Kutta scheme to solve the equation of characteristics. Then, we use this solution to approximate ϕ^n, by considering the projection of (24) on a suitable Lagrange finite element space (see Appendix A for more details).

Remark 1. It is worth to forecast some difficulties: in the numerical framework, the characteristic curves may go out of the computational domain especially when the computational mesh approximates a computational domain with curvy boundaries. In such cases, we project the final point of the characteristic curve, corresponding to $\mathbf{X}(\mathbf{x}, t^n; t^{n-1})$ for some n, on the mesh boundary. This is done by identifying the edge (or the face) of the boundary such that the barycentric coordinate of the final point with respect to this edge (or face) is strictly negative in the element to whom this edge (or face) belongs.

3.2 | Extension and regularization of the velocity field

Finally, we turn to a problem that has been left out so far, ie, the definition of the velocity field $\tilde{\mathbf{u}}$ that appears in equation (3.5). The implicit embedding associated with the level set implies that the velocity field $\tilde{\mathbf{u}}$ that transports the interface $\Gamma(t)$ needs to be defined and must be meaningful in the whole domain and not only on the interface, in order to advance the neighboring level sets and not only the zero level set.[49] In 2-phase flow simulations, the velocity field has a physical meaning on and away from $\Gamma(t)$. However, the evolution of the interface $\Gamma(t)$ depends only on the flow field in its vicinity. On the other hand, large velocity discrepancies between neighboring points may cause uncontrolled oscillations on the level sets and jeopardize the stability of the numerical algorithm. For this reason, the regularization of the velocity field is a step of our numerical scheme for the 2-flow problem. In addition, if the flow is free-surface, the velocity field is defined only on the subdomain Ω_1 of Ω representing the fluid domain, whereas the transport equation needs to be solved on the whole computational domain. Therefore, an extension of velocity is crucial for the resolution of free-surface flows while using the level-set formulation.

There are many approaches for regularizing the velocity field. The method suggested by Chang et al[40] and Sussman et al[50] builds regularized velocities from the underlying fluid velocities. As mentioned previously, the regularized velocity must be defined away from the interface, and that it smoothly approaches the prescribed interface velocity as the zero level set is approached. To this end, we extract the velocity values at all mesh vertices along the interface. Then, following the works of Burger[51] and de Gournay,[52] the regularized velocity denoted by $\tilde{\mathbf{u}}$ is searched as the unique solution in a suitable Hilbert space V to a Helmholtz problem of the following form:

$$\begin{cases} -a\Delta\tilde{\mathbf{u}} + \tilde{\mathbf{u}} = 0, & \text{on } \Omega \\ \nabla\tilde{\mathbf{u}} \cdot \mathbf{n} = 0, \quad \text{on} \quad \partial\Omega \quad \text{and} \quad \tilde{\mathbf{u}} = \mathbf{u}_{|\Gamma(t)}, & \text{on } \Gamma(t), \end{cases} \tag{27}$$

where $a > 0$ is the regularization length scale. This problem is equivalent to solving the variational problem: *find* $\tilde{\mathbf{u}} \in V$ *such that*

$$a\int_{\Omega(t)} \nabla\tilde{\mathbf{u}} \cdot \nabla\mathbf{v} + \int_{\Omega(t)} \tilde{\mathbf{u}}\mathbf{v} = \int_{\Gamma(t)} \mathbf{u}\mathbf{v}, \qquad \text{for all } \mathbf{v} \in V. \tag{28}$$

The left-hand side of Equation (29) is a coercive bilinear form on V, which is close to I (so that $\tilde{\mathbf{u}}$ is expected to be close to $\mathbf{u}$). Numerically, this weak form is solved using a finite element method.

Let us note that if in the case of free-surface flows, the flow velocity is smooth enough (ie, if it does not represent any large discrepancy between neighboring points), we only need to extend the velocity in the subdomain corresponding to the vacuum. This means that we only need to solve the Helmholtz equation on Ω_2. In other words, we solve

$$\begin{cases} -a\Delta\tilde{\mathbf{u}} + \tilde{\mathbf{u}} = 0, & \text{on } \Omega_2(t) \\ \nabla\tilde{\mathbf{u}} \cdot \mathbf{n} = 0, \quad \text{on} \quad \partial\Omega \cap \partial\Omega_1 \quad \text{and} \quad \tilde{\mathbf{u}} = \mathbf{u}_{|\Gamma(t)}, & \text{on } \Gamma(t). \end{cases} \tag{29}$$

In both cases, the extension regularization parameter a must be small enough such that large values of $\tilde{\mathbf{u}}$ outside of the free surface do not interfere with other values on the other side of the free surface. Indeed, if a is too large, some strictly positive level sets may cross the zero level set during advection. Therefore, the level-set function no longer represents the fluid domain during advection. For this reason, it is important to consider a Helmholtz operator and not only a Laplace operator. The identity part of the Helmholtz operator together with a conveniently small a guarantees a reasonable decrease for the extended velocity flow as the boundary of the domain far from the free surface is approached. On the other hand, a must obviously not be too small, since the solution $\tilde{\mathbf{u}}$ must be smooth enough such that the advection equation can be resolve by the characteristic method. In our numerical tests, we set $a = 0.5$.

3.3 | Resolution of the Navier-Stokes equations on the fluid domain

In this section, we are interested in the numerical resolution of the free-surface incompressible Navier-Stokes equation on the fluid domain at some time t. In other words, we endow the time-discretized problem with essential and natural boundary conditions on the fluid domain. This time discretization is done following the pioneering works.[34,35,53] More precisely, we consider the term

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right),$$

which actually models the *transport* of the *momentum* $\rho\mathbf{u}$ by the velocity field $\mathbf{u}$ with the idea of taking advantage of the *transport nature* of the nonlinear term. Hence, we use here the *backward method of characteristics* for the time discretization of the Navier-Stokes equations. Then, we will use a convenient finite element method for the spatial resolution of the time-discretized equation on the fluid domain.

3.3.1 | Time discretization by the method of characteristics

In this section, we describe the time discretization of the Navier-Stokes equations based on the method of characteristics. This is not the only possible choice. Nevertheless, we use this scheme because it is *unconditionally stable* (see the work of Pironneau[35]). Moreover, this method is already implemented in our framework to solve the advection equation (20). Therefore, we can use the same scheme for the resolution of the advection equation and the time discretization of the Navier-Stokes equation. Moreover, we obviously use the same time step Δt for both advection and Navier-Stokes equations. Indeed, we have the same subdivision $[t^{n-1}, t^n]$, with $t^n = t^{n-1} + \Delta t$, of the time interval $[0, T]$.

The main idea is that the nonlinear convection part of Navier-Stokes equations can be hidden in the Cauchy problem (22), ie, the operator $\frac{\partial}{\partial t} + \mathbf{u}.\nabla$ may be turned into a total derivative (also called the Lagrange derivative) $\frac{\mathrm{d}}{\mathrm{d}t}$. Therefore, the first equation of (14) and (15) is recast into the following form:

$$\rho\frac{\mathrm{d}\mathbf{u}\left(\mathbf{X}(\mathbf{x}, t; s), s\right)}{\mathrm{d}s}|_{s=t} - \mu\Delta\mathbf{u} + \nabla p = \rho\mathbf{f}. \tag{30}$$

Hence, we denote $\mathbf{u}(\mathbf{x}, t^n)$ by $\mathbf{u}^n(\mathbf{x})$ for all n and use the following explicit approximation:

$$\frac{\mathrm{d}\mathbf{u}\left(\mathbf{X}(\mathbf{x}, t^n; s), s\right)}{\mathrm{d}s}|_{s=t^n} \approx \frac{\mathbf{u}\left(\mathbf{X}(\mathbf{x}, t^n; t^n), t^n\right) - \mathbf{u}\left(\mathbf{X}(\mathbf{x}, t^n; t^{n-1}), t^{n-1}\right)}{\Delta t},$$

since

$$\frac{\mathrm{d}\mathbf{u}\left(\mathbf{X}(\mathbf{x}, t^n; s), s\right)}{\mathrm{d}s}|_{s=t^n} = \frac{\mathbf{u}\left(\mathbf{X}(\mathbf{x}, t^n; t^n), t^n\right) - \mathbf{u}\left(\mathbf{X}(\mathbf{x}, t^n; t^{n-1}), t^{n-1}\right)}{\Delta t} + O(\Delta t).$$

This approximation is equivalent, using the latter notation together with definition (22) of the characteristic curve, to

$$\frac{\mathrm{d}\mathbf{u}\left(\mathbf{X}(\mathbf{x}, t^n; s), s\right)}{\mathrm{d}s}|_{s=t^n} \approx \frac{\mathbf{u}^n(\mathbf{x}) - \mathbf{u}^{n-1}\left(\mathbf{X}(\mathbf{x}, t^n; t^{n-1})\right)}{\Delta t}.$$

Therefore, denoting $\mathbf{X}(\mathbf{x}, t^n; t^{n-1})$ by $\mathbf{X}^{n-1}(\mathbf{x})$, the time-discretized Navier-Stokes writes

$$\begin{cases} \rho \frac{\mathbf{u}^n(\mathbf{x}) - \mathbf{u}^{n-1} \circ \mathbf{X}^{n-1}(\mathbf{x})}{\Delta t} - \mu \Delta \mathbf{u}^n(\mathbf{x}) + \nabla p^n(\mathbf{x}) = \rho \mathbf{f}^n, & \text{in } \Omega_i(t^n) \\ \operatorname{div} \mathbf{u}^n(\mathbf{x}) = 0, & \text{in } \Omega_i(t^n) \end{cases} \tag{31}$$

or, equivalently,

$$\begin{cases} \rho \frac{\mathbf{u}^n(\mathbf{x})}{\Delta t} - \mu \Delta \mathbf{u}^n(\mathbf{x}) + \nabla p^n(\mathbf{x}) = \rho \mathbf{f}^n + \rho \frac{\mathbf{u}^{n-1} \circ \mathbf{X}^{n-1}(\mathbf{x})}{\Delta t}, & \text{in } \Omega_i(t^n) \\ \operatorname{div} \mathbf{u}^n(\mathbf{x}) = 0, & \text{in } \Omega_i(t^n). \end{cases} \tag{32}$$

Indeed, $\mathbf{u}^{n-1} \circ \mathbf{X}^{n-1}(\mathbf{x})$ approximates the velocity at the point $\mathbf{X}^{n-1}(\mathbf{x})$ at time t^{n-1}. Here, $i \in \{1, 2\}$ if the flow is bifluid and $i = 1$ if the flow is free-surface. We endow problem (32) with slip boundary and free-surface conditions. Indeed, the boundary condition of (14) writes here for 2-phase flows as*

$$\begin{cases} \mathbf{u}^n = 0, \quad \text{on} \quad \partial\Omega_D(t^n) \\ \sigma^n \mathbf{n}_{\partial\Omega_N(t^n)} = -p_e \mathbf{n}_{\partial\Omega_N(t^n)}, \quad \text{on} \quad \partial\Omega_N(t^n) \\ \mathbf{u}^n \cdot \mathbf{n}_{\partial\Omega_S(t^n)} = 0, \quad \text{on} \quad \partial\Omega_S(t^n) \\ \left[\alpha \mathbf{u}^n + \mu \left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) \mathbf{n}_{\partial\Omega_S(t^n)}\right]_{\tan} = 0, \quad \text{on} \quad \partial\Omega_S(t^n) \\ [\sigma]_{\Gamma(t^n)} \mathbf{n}_1 = -\gamma \kappa \mathbf{n}_1, \quad \text{in} \quad \Gamma(t^n). \end{cases} \tag{33}$$

Likewise, the boundary condition of (15) for free-surface flows writes

$$\begin{cases} \mathbf{u}^n = 0, \quad \text{on} \quad \partial\Omega_D(t^n) \cap \partial\Omega_1(t^n) \\ \sigma^n \mathbf{n}_{\partial\Omega_N(t^n)} = -p_e \mathbf{n}_{\partial\Omega_N(t^n)}, \quad \text{on} \quad \partial\Omega_N(t^n) \cap \partial\Omega_1(t^n) \\ \mathbf{u}^n \cdot \mathbf{n}_{\partial\Omega_S(t^n)} = 0, \quad \text{on} \quad \partial\Omega_S(t^n) \cap \partial\Omega_1(t^n) \\ \left[\alpha \mathbf{u}^n + \mu \left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) \mathbf{n}_{\partial\Omega_S(t^n)}\right]_{\tan} = 0, \quad \text{on} \quad \partial\Omega_S(t^n) \cap \partial\Omega_1(t^n) \\ \left(\mu \left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) - p^n\right) \mathbf{n}_1 = -\left(\gamma \kappa + p^a\right) \mathbf{n}_1, \quad \text{in} \quad \Gamma(t^n). \end{cases} \tag{34}$$

Obviously, at each time step, the Navier-Stokes problem becomes a Stokes problem plus a transport of the previous solution on the characteristic. This problem is resolved in 2 steps.

1. Approximate the characteristic curves $\mathbf{X}^{n-1}(\mathbf{x})$.
2. Solve the resulting generalized Stokes system with the corresponding boundary conditions.

Remark 2.

- The approximation of characteristic curves $\mathbf{X}^{n-1}(\mathbf{x})$ in each time interval $[t^{n-1}, t^n]$ for the Navier-Stokes problem is implemented in the same way as in Section 3.1.
- Similarly to Section 3.1, the characteristic curves may cross some elements or go out of the computational domain. In these cases, we identify the last element as the characteristic curve crossed and project the final point of this curve to the corresponding edge (or face). Let us recall that in the case of free-surface problems, we intend to solve the Navier-Stokes equation only on the fluid domain (Ω_1). However, the flow we consider to solve the characteristic equation is not the flow velocity $\mathbf{u}$ but the extended flow velocity $\tilde{\mathbf{u}}$. This lets us handle situations where the characteristic curves cross the free surface of the fluid but stay in the computational domain D.

3.3.2 | Variational formulation for the bifluid problem

As mentioned previously, we intend to solve the time-discretized Navier-Stokes equation (32) together with boundary conditions (33) by the finite element method for all integer $n \in [0, \frac{T}{\Delta t}]$. Equation (32) is nothing but a generalized Stokes equation. The associated variational formulation is then obtained by the same method as for generalized Stokes problems. In this part, we assume, for the sake of simplicity, that the fluid satisfies everywhere on $\partial\Omega$ the Navier slip boundary

*Let us mention that the curvature κ at the free surface as well as the normal vector $\mathbf{n}_1$ depends on time t^n since they correspond to the interface/free surface Γt^n, which travels by time. Nevertheless, for the sake of simplicity, we do not use any n superscript for these symbols in (32) either in the following section.

conditions, ie, $\partial\Omega = \partial\Omega_S$ and $\partial\Omega_D = \partial\Omega_N = \emptyset$. Indeed, this simplification is considered in this part, since the addition of Dirichlet or Neumann boundary conditions reveals no mathematical or technical difficulty†. Therefore, here, we consider Hilbert spaces for the velocity flow and the pressure, respectively, V_n and Q_n, defined by (see, for instance, the works of Verfürth[54-56] and Dione and Urquiza[57])

$$V_n = \left\{ \mathbf{v} \in \left(\mathbb{H}^1 \left(\Omega(t^n) \right) \right)^d, \mathbf{v} \cdot \mathbf{n}_{\partial\Omega(t^n)} = 0 \quad \text{on} \quad \partial\Omega(t^n) \right\}, \tag{35}$$

$$Q_n = \mathbb{L}^2(\Omega(t^n)). \tag{36}$$

The variational formulation is then given in the following proposition.

Proposition 1. *The variational formulation corresponding to (32) and (33) is as follows.*
Find $(\mathbf{u}^n, p^n)$ in $V_n \times Q_n$ such that, for all $(\mathbf{v}, q) \in V_n \times Q_n$, we have

$$\begin{cases} a(\mathbf{u}^n, \boldsymbol{v}) + b(\mathbf{v}, p^n) = l(\mathbf{v}), \\ b(\mathbf{u}^n, q) = 0. \end{cases} \tag{37}$$

In this formulation, $a(.,.)$ is a continuous bilinear symmetric coercive operator defined on V_n by

$$a(\mathbf{u}, \mathbf{v}) = \frac{\rho}{\Delta t} \int_{\Omega(t^n)} \mathbf{u} \cdot \mathbf{v} + 2\mu \int_{\Omega(t^n)} \mathbf{D}(\mathbf{u}) : \mathbf{D}(\boldsymbol{v}) + \alpha \int_{\partial\Omega(t^n)} [\mathbf{u}]_{\tan} \cdot [\mathbf{v}]_{\tan}. \tag{38}$$

Besides, the bilinear operator $b(.,.)$ is defined on $V_n \times Q_n$ by

$$b(\mathbf{u}, q) = -\int_{\Omega(t^n)} \mathrm{div}\mathbf{u} q. \tag{39}$$

The linear operator $l(.)$ is continuous on V_n and given by

$$l(\mathbf{v}) = \int_{\Omega(t^n)} \rho \left(\mathbf{f} + \frac{\mathbf{u}_\star^{n-1}}{\Delta t} \right) \cdot \mathbf{v} - \int_{\Gamma(t^n)} \gamma\kappa \mathbf{n}_1 \cdot \mathbf{v}.$$

Proof. The proof is based on classical computations presented in Appendix B. □

Remark 3. Let us mention that the operator b satisfies the Babuska-Brezzi inf-sup condition and the variational formulation (37) is well-posed (see, for instance, the work of Brezzi and Fortin[58]).

3.3.3 | Variational formulation for the free-surface problem

In order to obtain the variational formulation for the time-discretized Navier-Stokes free-surface problem, we consider (32) together with boundary conditions (34). Again, we assume that $\partial\Omega_S = \partial\Omega$. The variational formulation for this problem is obtained exactly by the same method as in the previous section for bifluid problems. The only difference is that the Hilbert spaces we consider are defined on the fluid domain Ω_1.

Similarly, integrations are done on Ω_1. After similar computations, we get the following formulation for the problem.

Proposition 2. *The variational formulation corresponding to (32) and (34) is*

$$V_n = \left\{ \boldsymbol{v} \in \left(\mathbb{H}^1 \left(\Omega_1(t^n) \right) \right)^d, \boldsymbol{v} \cdot \mathbf{n}_{\partial\Omega_S(t^n)} = 0 \quad \text{on} \quad \partial\Omega_S(t^n) \right\}, \tag{40}$$

$$Q_n = \mathbb{L}^2 \left(\Omega_1(t^n) \right). \tag{41}$$

Find $(\boldsymbol{u}^n, p^n)$ in $V_n \times Q_n$ such that, for all $(\boldsymbol{v}, q) \in V_n \times Q_n$, we have

$$\begin{cases} a(\boldsymbol{u}^n, \boldsymbol{v}) + b(\boldsymbol{v}, p^n) = l(\boldsymbol{v}), \\ b(\boldsymbol{u}^n, q) = 0. \end{cases} \tag{42}$$

†Nevertheless, if the flow is bifluid and $\partial\Omega_D = \partial\Omega$, the discrete space we consider for the pressure is the set of zero-mean square-integrable function. This space is classically denoted by $Q_n = \mathbf{L}_0^2(\Omega)$.

In this formulation, $a(.,.)$ is a continuous bilinear symmetric coercive operator defined on V_n by

$$a(\boldsymbol{u},\boldsymbol{v}) = \frac{\rho}{\Delta t}\int_{\Omega_1(t^n)} \boldsymbol{u}\cdot\boldsymbol{v} + 2\mu\int_{\Omega_1(t^n)} \boldsymbol{D}(\boldsymbol{u}) : \boldsymbol{D}(\boldsymbol{v}) + \alpha\int_{\partial\Omega_S(t^n)} [\boldsymbol{u}]_{\tan}\cdot[\boldsymbol{v}]_{\tan}. \tag{43}$$

Besides, the bilinear operator $b(.,.)$ is defined on $V_n \times Q_n$ by

$$b(\boldsymbol{u}, q) = -\int_{\Omega_1(t^n)} \mathrm{div}\boldsymbol{u}q. \tag{44}$$

The linear operator $l(.)$ is continuous on V_n and given by

$$l(\boldsymbol{v}) = \int_{\Omega_1(t^n)} \rho\left(\boldsymbol{f} + \frac{\boldsymbol{u}_\star^{n-1}}{\Delta t}\right)\cdot\boldsymbol{v} - \int_{\Gamma(t^n)} \left(p^a + \gamma\kappa\right)\mathbf{n}_1\cdot\boldsymbol{v}.$$

Remark 4. The slip boundary condition is considered here using the penalization technique, ie, instead of the bilinear form a defined on V_n in (37) and (42), we consider a_ϵ defined below, and we define the corresponding variational formulation on $K_n \times Q_n$ with $K_n = \left(\mathbf{H}^1(\Omega(t^n))\right)^d$ (or $K_n = \mathbf{H}^1(\Omega_1(t^n))^d \times Q_n$ if the flow is free-surface) instead of on $V_n \times Q_n$.

$$a_\epsilon(\mathbf{u},\mathbf{v}) = \frac{\rho}{\Delta t}\int_{\Omega(t^n)} \mathbf{u}\cdot\mathbf{v} + 2\mu\int_{\Omega(t^n)} \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) + \alpha\int_{\partial\Omega_S(t^n)} [\mathbf{u}]_{\tan}\cdot[\mathbf{v}]_{\tan} + \frac{1}{\epsilon}\int_{\partial\Omega_S(t^n)} (\mathbf{u}\cdot\mathbf{n})(\mathbf{v}\cdot\mathbf{n})$$

3.3.4 | Penalization technique for the slip boundary condition

Variational formulations (37) and (42) are defined on the Hilbert spaces defined in the 2 previous sections. However, in both cases, the construction of a basis of the space denoted by V_n as well as its finite element approximation is a challenge since V_n contains only functions whose velocity is tangent to $\partial\Omega_S(t^n)$. Therefore, it is a strict subspace of $\left(\mathbf{H}^1(\Omega(t^n))\right)^d$ if the flow is 2-phase and a strict subspace of $\left(\mathbf{H}^1(\Omega_1(t^n))\right)^d$ if it is free-surface. We use here the penalization technique to overcome this difficulty (see the work of Dione and Urquiza[57]). In this method, we no longer consider a strict subspace of $\mathbf{H}^1(\Omega(t^n))^d$ or $\mathbf{H}^1(\Omega_1(t^n))^d$ but, instead, these entire spaces. The nonpenetration condition $\left(\mathbf{u}\cdot\mathbf{n}_{\partial\Omega_S(t^n)} = 0\right)$ is considered by adding a penalization term to the first equation of the variational formulation. In fact, we consider the following variational formulation.

Find $(\mathbf{u}_\epsilon^n, p_\epsilon^n)$ in $K_n \times Q_n$ such that for all $(\mathbf{v}, q) \in K_n \times Q_n$, we have

$$\begin{cases} a_\epsilon(\mathbf{u}_\epsilon^n, \mathbf{v}) + b(\mathbf{v}, p_\epsilon^n) = l(\mathbf{v}), \\ b(\mathbf{u}_\epsilon^n, q) = 0, \end{cases} \tag{45}$$

where

$$K_n = \left(\mathbf{H}^1(\Omega(t^n))\right)^d \quad \text{if the flow is 2-phase,}$$

and

$$K_n = \left(\mathbf{H}^1(\Omega_1(t^n))\right)^d \quad \text{if it is free-surface.}$$

In this technique, the bilinear form a_ϵ is defined on the entire K_n by

$$a_\epsilon(\mathbf{u},\mathbf{v}) = \frac{\rho}{\Delta t}\int_{\Omega(t^n)} \mathbf{u}\cdot\mathbf{v} + 2\mu\int_{\Omega(t^n)} \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) + \alpha\int_{\partial\Omega_S(t^n)} [\mathbf{u}]_{\tan}\cdot[\mathbf{v}]_{\tan} + \frac{1}{\epsilon}\int_{\partial\Omega_S(t^n)} (\mathbf{u}\cdot\mathbf{n})(\mathbf{v}\cdot\mathbf{n}),$$

for a 2-phase flow, and by

$$a_\epsilon(\mathbf{u},\mathbf{v}) = \frac{\rho}{\Delta t}\int_{\Omega_1(t^n)} \mathbf{u}\cdot\mathbf{v} + 2\mu\int_{\Omega_1(t^n)} \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) + \alpha\int_{\partial\Omega_S(t^n)} [\mathbf{u}]_{\tan}\cdot[\mathbf{v}]_{\tan} + \frac{1}{\epsilon}\int_{\partial\Omega_S(t^n)} (\mathbf{u}\cdot\mathbf{n})(\mathbf{v}\cdot\mathbf{n}),$$

for a free-surface one.

The penalization technique is interesting since the solution $(\mathbf{u}_\epsilon^n, p_\epsilon^n)$ of (45) converges (for the usual norm $K_n \times Q_n$) to the solution $(\mathbf{u}^n, p^n)$ of (37) (or (42) if the flow is free-surface) when ϵ tends to 0 (see the work of Dione and Urquiza[57]).

3.3.5 | Resolution by the finite element method

We use the Galerkin finite element approximation to find the following discrete problem: find $(\mathbf{u}_h, p_h) \in K_n^h \times Q_n^h$ such that

$$\begin{cases} \forall \mathbf{v}_h \in K_n^h, & a_\epsilon(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = l(\mathbf{v}_h) \\ \forall q_h \in Q_n^h, & b(\mathbf{u}_h, q_h) = 0, \end{cases} \tag{46}$$

where $K_n^h \subset K_n$ and $Q_n^h \subset Q_n$ represent 2 families of finite-dimensional subspaces constructed from a triangulation $\mathcal{T}_h$ covering the fluid domain $\Omega(t^n)$ (or $\Omega_1(t^n)$). The symbol h denotes the space discretization parameter, which is nothing but the characteristic element size. The discrete border of the triangulation $\mathcal{T}_h$ is denoted by $\partial\mathcal{T}_h$. It can also be seen as a discrete approximation of the boundary of $\Omega(t^n)$ (or $\Omega_1(t^n)$ if it is free-surface). The symbol Γ_h represents the part of $\mathcal{T}_h$ that corresponds to a discrete approximation of the interface/free surface. Moreover, $\partial_S\mathcal{T}_h$ represents the part of $\partial\mathcal{T}_h$ that corresponds to the discretization of the $\partial_S\Omega(t^n)$ part of the border of the fluid domain.

Besides, $a_\epsilon(\mathbf{u}_h, \mathbf{v}_h)$, $b(\mathbf{v}_h, p_h)$, and $l(\mathbf{v}_h)$ are bilinear and linear forms defined on $K_n^h \times K_n^h$, $K_n^h \times Q_n^h$, and K_n^h, respectively, as follows:

$$\begin{aligned} a_\epsilon(\mathbf{u}_h, \mathbf{v}_h) = & \sum_{K\in\mathcal{T}_h} \frac{\rho}{\Delta t}\int_K \mathbf{u}_h.\mathbf{v}_h \mathrm{d}x + \sum_{K\in\mathcal{T}_h} 2\mu \int_K \mathbf{D}(\mathbf{u}_h) : \mathbf{D}(\mathbf{v}_h)\mathrm{d}x + \alpha \sum_{E\in\partial_S\mathcal{T}_h}\int_E \mathbf{u}_h\cdot\mathbf{v}_h \mathrm{d}x \\ & + \left(\frac{1}{\epsilon} - \alpha\right) \sum_{E\in\partial_S\mathcal{T}_h}\int_E (\mathbf{u}_h\cdot\mathbf{n}(E))\,(\mathbf{v}_h\cdot\mathbf{n}(E))\,\mathrm{d}x, \\ b(\mathbf{v}_h, p_h) = & \sum_{K\in\mathcal{T}_h}\int_K - p_h \mathrm{div}\mathbf{v}_h, \\ l(\mathbf{v}_h) = & \sum_{K\in\mathcal{T}_h}\int_K \rho\mathbf{f}_h\cdot\mathbf{v}_h \mathrm{d}x + \frac{\rho}{\Delta t}\sum_{K\in\mathcal{T}_h}\int_K \rho\left(\mathbf{u}_\star^{n-1}\right)_h.\mathbf{v}_h \mathrm{d}x + l_{\text{Surface}}(\mathbf{v}_h), \end{aligned}$$

where $\mathbf{n}(E)$ is the exterior normal (with respect to $\mathcal{T}_h$) of the edge E for all edge E of Γ_h. Moreover, the term $l_{\text{Surface}}(\mathbf{v}_h)$ is a discretization of the surface tension term $-\int_{\Gamma_{t^n}} \gamma\kappa\mathbf{n}_1.\mathbf{v}$ if the flow is 2-phase and is a discretization of the surface tension and atmospheric pressure term $-\int_{\Gamma_{t^n}} (\gamma\kappa + p^a)\,\mathbf{n}_1.\mathbf{v}$ if the flow is free-surface. This term will be the concern in the next section. The existence and uniqueness of the weak formulation for the generalized Stokes problem has been proven (see the works of Dione and Urquiza,[57] Ern and Guermond,[59] and Quarteroni[60]). This proof is due to (i) the ellipticity of the form $a_\epsilon(.,.)$ and (ii) the compatibility of the spaces of velocity and pressure results satisfying the Babuska-Brezzi condition, also called the *inf-sup condition* on the form $b(.,.)$, ie, there exists a positive constant C such that

$$\inf_{q\in Q_n}\sup_{\mathbf{v}\in K_n} \frac{b(\mathbf{v}, p)}{\|\mathbf{v}\|_1\|q\|_0} \geq C > 0, \tag{47}$$

where $\|\mathbf{v}\|_1 = \left(\Sigma_{i=1}^d \|v_i\|_1{}^2\right)^{1/2}$ and $\|.\|_1$, $\|.\|_0$ are standard notations of norms in the Sobolev spaces $H^1(\Omega)$ and $L^2(\Omega)$ (or $H^1(\Omega_1)$ and $L^2(\Omega_1)$ if the problem is free-surface), respectively. The approximative problem also requires a compatibility condition, which means that the discrete spaces of velocity need to be "rich" enough to compare with the one of pressure. For this reason, we choose mini elements ($\mathbb{P}^1$-bubble/$\mathbb{P}^1$) as discrete spaces[‡]. Hence, problem (46) leads to solving the square linear system, ie,

$$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix}\begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}, \tag{48}$$

where the matrices A and B correspond to the bilinear forms a_ϵ and b, respectively, and F corresponds to the right-hand side l of the first equation. System (48) is sparse symmetric but not positive, and its size is $\dim K_n^h + \dim Q_n^h$. In practice, this linear system is solved by the classical *Uzawa* method,[61] as recalled in the following paragraph.

Uzawa algorithm: In order to understand the Uzawa method, let us first remark that the solution of the linear problem (48) is the unique solution to the following saddle-point problem:

$$\begin{cases} \text{Find} \quad (U, P) \in K_n^h \times Q_n^h \quad \text{such that} \\ \mathcal{L}(U, q) \leq \mathcal{L}(U, P) \leq \mathcal{L}(v, P) \quad \forall (v, q) \in K_n^h \times Q_n^h, \end{cases} \tag{49}$$

where

$$\mathcal{L}(v, q) = \frac{1}{2}a_\epsilon(v, v) + b(v, q) - l(v),$$

[‡]Other choices satisfying the inf-sup condition are possible. For instance, we can consider Taylor-Hood elements ($\mathbb{P}^2/\mathbb{P}^1$).

or, equivalently,

$$\mathcal{L}(v,q)=\frac{1}{2}Av\cdot v+Bq\cdot v-F\cdot v.$$

Then, the Uzawa method consists in solving the saddle-point problem (49) using a gradient method applied to the minimization of the dual function. Indeed, we initialize the algorithm by choosing $P_0\in K_n^h$. Then, for each iteration k, we have the following steps.

- We compute velocity U_k satisfying
$$\mathcal{L}(U_k,P_k)\leq\mathcal{L}(v,P_k)\quad\forall v\in K_n^h.$$
- The initial condition P_{k+1} for the following iteration is computed moving in the gradient direction by
$$P_{k+1}=P_k+\beta\nabla_q\mathcal{L}(U_k,P_k),$$
where $\beta>0$ is the gradient method step.

Equivalently, this algorithm writes in the matrix version at an iteration k as follows:

- Solve
$$AU_k=F-B^tP_k.$$
- Set
$$P_{k+1}=P_k+\beta BU_k.$$

Uzawa's algorithm converges for $\beta>0$ sufficiently small.

Approximation of the surface tension term. Surface tension plays a crucial role in the regularization of the free surface for incompressible fluids with small viscosity. The accurate computation of this term is one of the critical stages in any interface tracking or capturing technique. Classically, the level-set function can be used to calculate the unit normal vector and the mean curvature at the interface by the following formulas:

$$\mathbf{n}=\left.\frac{\nabla\phi}{|\nabla\phi|}\right|_{\phi=0};\quad\kappa=\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)|_{\phi=0}.$$

It can be seen that these formulas require an approximation of the gradient; however, in our approach, the interface is explicitly discretized in the triangulation $\mathcal{T}_h$. This feature gives us an alternative technique to approximate the interface via a set of connected segments (faces in 3 dimensions). In the 2D case, we denote by $(\mathbf{x}_i)_{1\leq i\leq ns}$ the set of ordered vertices along the discrete curve Γ_h such that $\mathbf{x}_{i-1},\mathbf{x}_i$, and $\mathbf{x}_{i+1}$ represent its 3 consecutive points and $\mathbf{x}_0\equiv\mathbf{x}_{ns},\mathbf{x}_1\equiv\mathbf{x}_{ns+1}$ if Γ_h is a closed curve. Using the quadrature formula along each edge E of Γ_h, it has been shown in the work of Bui et al[39] that the surface tension term can be rewritten as follows, for all $\mathbf{v}_h\in K_n^h$:

$$\begin{aligned}\int_{\Gamma_h}\gamma\kappa\mathbf{n}_1.\mathbf{v}_h\mathrm{d}s&=\sum_{E\subset\Gamma_h}\frac{|E|}{2}\sum_{\mathbf{x}_i\in E}\gamma\kappa(\mathbf{x}_i)\mathbf{n}_h(\mathbf{x}_i).\mathbf{v}_h(\mathbf{x}_i)\\&=\sum_{\mathbf{x}_i\in\Gamma_h}\gamma\kappa(\mathbf{x}_i)\mathbf{n}_h(\mathbf{x}_i).\mathbf{v}_h(\mathbf{x}_i)\sum_{E\ni\mathbf{x}_i}\frac{|E|}{2},\end{aligned}\tag{50}$$

where the unit normal vector $\mathbf{n}_h$ to Ω_1 is computed from the approximation of the unit tangent vector $\tau=(\tau_1,\tau_2)^t$ at each vertex $\mathbf{x}_i\in\Gamma_h$: $\tau(\mathbf{x})_i=\overrightarrow{\mathbf{x}_{i+1}\mathbf{x}_{i-1}}/\|\overrightarrow{\mathbf{x}_{i+1}\mathbf{x}_{i-1}}\|$; hence, $\mathbf{n}_h(\mathbf{x}_i)=(\tau_2(\mathbf{x}_i),-\tau_1(\mathbf{x}_i))^t$. The mean curvature $\kappa(\mathbf{x}_i)$ is obtained as the inverse of the radius $r(\mathbf{x}_i)$, which can be computed via the following approximation[62]:

$$r(\mathbf{x}_i)=\frac{1}{4}\left(\frac{\langle\overrightarrow{\mathbf{x}_i\mathbf{x}_{i-1}},\overrightarrow{\mathbf{x}_i\mathbf{x}_{i-1}}\rangle}{\langle-\mathbf{n}_h(\mathbf{x}_i),\overrightarrow{\mathbf{x}_i\mathbf{x}_{i-1}}\rangle}+\frac{\langle\overrightarrow{\mathbf{x}_i\mathbf{x}_{i-1}},\overrightarrow{\mathbf{x}_i\mathbf{x}_{i+1}}\rangle}{\langle-\mathbf{n}_h(\mathbf{x}_i),\overrightarrow{\mathbf{x}_i\mathbf{x}_{i+1}}\rangle}\right).$$

Other formulas can be used to approximate $r(\mathbf{x})$ or $\kappa(\mathbf{x})$ (see, for instance, the work of Tryggvason et al[3]).

The atmospheric pressure term can then be added with no difficulty, ie,

$$\int_{\Gamma_h}\left(\gamma\kappa+p^a\right)\mathbf{n}_1.\mathbf{v}_h\mathrm{d}s=\sum_{\mathbf{x}_i\in\Gamma_h}\left(\gamma\kappa+p^a\right)(\mathbf{x}_i)\mathbf{n}_h(\mathbf{x}_i).\mathbf{v}_h(\mathbf{x}_i)\sum_{E\ni\mathbf{x}_i}\frac{|E|}{2}.$$

This technique can be extended straightforwardly to 3 dimensions, where the unit normal is then taken as the weighted average value of the unit normals of all triangles sharing vertex $\mathbf{x_i}$. Moreover, the mean curvature is computed by approximating the discrete contour of the mesh at each vertex by a quadratic surface. This means that for each vertex on the free surface, we intend to find the quadratic surface that minimizes the distance to the discrete contour of the mesh

at this vertex. Therefore, this is done by a solving a polynomial minimization problem (see, eg, the work of Frey[63] for more details).

3.4 | Redistancing

It is well known, in the context of the level-set method, that the level-set function must usually satisfy

$$|\nabla \phi| = 1. \tag{51}$$

Unfortunately, when ϕ is transported by a physical velocity field using Equation (20), all the isolines do not travel with the same speed. As a consequence, the level set does not preserve property (51). A natural choice to reinitialize the level-set function is the signed distance function to the interface for all time iterations $n \in \mathbf{N}$ in the discrete approximation $\mathcal{T}_h$ of the computational domain Ω, ie,

$$\phi(x) = \begin{cases} d(x, \Gamma_h), & \text{if } x \in \Omega^1 \\ 0, & \text{if } x \in \Gamma_h \\ -d(x, \Gamma_h), & \text{if } x \in \Omega^2, \end{cases}$$

where Ω^1 (respectively, Ω^2) represents parts of $\mathcal{T}_h$ corresponding to Ω_1 (respectively, Ω_2). In our scheme, this signed distance function is approximated by the *redistancing procedure* studied in the work of Dapogny and Frey,[64] which consists in the implementation of the following 2 steps.

1. Initialization ϕ^0 of ϕ: denoting by $\mathcal{T}_\Gamma$ the set of mesh elements intersected by the interface, ie, $\mathcal{T}_\Gamma = \{K \in \mathcal{T}_h : K \cup \Gamma_h \neq \emptyset\}$, $\phi^0(x)$ is defined as
$$\phi^0(x) = \begin{cases} \phi(x), & \text{if } x \in \mathcal{T}_\Gamma \\ +\infty, & \text{if } x \in \Omega^1 \setminus \mathcal{T}_\Gamma \\ -\infty, & \text{if } x \in \Omega^2 \setminus \mathcal{T}_\Gamma. \end{cases}$$
2. Numerical computation of ϕ as the *steady solution* of the so-called *Eikonal equation*, ie,
$$\begin{cases} \frac{\partial \phi}{\partial t}(\mathbf{x}, t) + \operatorname{sgn}(\phi_0)(|\nabla \phi| - 1) = 0, \quad \forall (\mathbf{x}, t) \in \Omega \times (0, T) \\ \phi(\mathbf{x}, 0) = \phi^0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega. \end{cases} \tag{52}$$

The long-term solution of this equation keeps a similar behavior in the "vicinity" of the zero isoline, ie, the position of interface $\Gamma(t^n)$ is preserved to ensure constraint (51). The numerical computation of the solution of this equation is based on the explicit expression of its unique uniformly continuous viscosity solution (see, eg, the work of Dapogny and Frey[64] for more details).

3.5 | Mesh adaptation

The key tool that lets us solve the time-discretized fluid problem on the fluid domain (in the case of the free-surface problem) is the anisotropic mesh adaptation of the computational domain Ω based on the location of the free surface or fluids' interface. This adaptation is based on a highly qualified explicit discretization of the free surface/interface, which is implicitly defined by the zero level set of a continuous function defined on the computational domain. As explained in detail in the work of Dapogny et al,[65] the adaptation is done in 2 steps.

1. **Explicit discretization of the zero level set:** Let us consider a mesh $\mathcal{T}_h$ of the computational domain and a finite element function ϕ_h defined on this mesh such that its zero level set describes the free surface/interface. This step consists in constructing a new mesh of the computational domain such that it discretizes explicitly the free surface/interface. At first, the set $\mathcal{K}$ of elements of the mesh, which contains the free surface/interface, is identified by evaluating the values of ϕ_h on degrees of freedom by using the marching cubes algorithm.[66] Then, for each element of this set, the free surface is constructed based on these values of ϕ_h seen as a finite element function. Finally, the elements of $\mathcal{K}$ are split to give a conform explicit discretization of the free surface/interface (see section 5.1 in the work of Dapogny et al[65] for more details).
2. **Improving the conform mesh:** Once the previous step is done, we obtain a new mesh for the computational domain, which discretizes explicitly the free surface/interface. Nevertheless, this new mesh is often very ill shaped. For this

reason, a remeshing procedure is considered to improve the quality of the mesh. The idea here is based on the construction of a *metric tensor* to prescribe the characteristics (*size*, *shape*, and *orientation*) of the mesh elements. The definition of this metric tensor relies on the information related to numerical error estimates as geometric error, interpolation error, and approximation error (see the work of Frey and George[62] for details). The estimates we consider here are from the work of Pironneau[35] for the time-discretized Navier-Stokes problem and from the work of Bui et al[67] for the level-set advection equation as well as for the discrete approximation of the smooth boundary of the computational domain. More precisely, we intend to get a small approximation error based on the error estimate given in the work of Pironneau[35] for the Navier-Stokes equation with Dirichlet boundary conditions. Indeed, let $\mathbf{u}$ be the smooth solution of the Navier-Stokes equations with the Dirichlet boundary condition, whereas $\mathbf{u}_h$ is the solution of the associated problem discretized temporarily by the characteristic method and spatially by the finite element method as described in the previous section. Then, we have the following error estimate[§]:

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2} \leq c(h + \Delta t + h^2/\Delta t), \tag{53}$$

where c is a strictly positive constant and h is the characteristic mesh element size. According to this error estimate, there should be coherence between the time step and the space step in order to give an interesting upper bound for the velocity $\mathbf{L}^2$ error. More precisely, if Δt is too small compared to the mesh elements' size h, this upper bound is not interesting.

Then, we use the estimates in the work of Bui et al[67] for the advection equation on the whole space conjecturing that similar estimates hold true on bounded domains. In fact, we see that the approximation error associated with the level-set function is bounded above by the interpolation error of the level-set function, ie,

$$\|\phi - \phi_h\|_{L^\infty(\Omega)} \leq \|\phi - \Pi_h\phi\|_{L^\infty(\Omega)} + \|\phi_0 - \Pi_h\phi_0\|_{L^\infty(\Omega)} + c_1\|\tilde{\mathbf{u}} - \Pi\tilde{\mathbf{u}}\|_{L^\infty(\Omega)} + c_2 e^{\delta t}\delta t, \tag{54}$$

where c_1 and c_2 are constants depending on initial data ϕ_0 and velocity $\mathbf{u}$, whereas the operator Π_h is the $\mathbb{P}^1$ interpolate over the mesh $\mathcal{T}_n^h$ covering the computational domain Ω on the interval $[t^n, t^{n+1}]$. On the other hand, we have the following geometric estimate. In other words, the Haussdorff distance between the discrete zero-isoline contour Γ_h and the continuous one Γ satisfies

$$d^H(\Gamma_t, \Gamma_n^h) \leq \sup\left(\frac{\sup_{x\in\Omega}|\nabla\phi(x)|}{\inf_{x\in\Omega}|\nabla\phi(x)|^2}, \frac{\sup_{K\in\mathcal{T}_n^h}|\nabla\phi_h(x)|_K|}{\inf_{K\in\mathcal{T}_n^h}|\nabla\phi_h(x)|_K|^2}\right)\|\phi - \phi_h\|_{L^\infty(\Omega)}. \tag{55}$$

Therefore, we see that the geometric error is bounded by the level-set function approximation error, which is, itself, bounded by the interpolation error for the solution of the problem. Using the result in the work of Frey and Alauzet[68] on the $\mathbf{L}^\infty$ error estimate for the Lagrange finite element $\mathbb{P}^1$ interpolation, for all functions defined on Ω and on all elements K of the mesh $\mathcal{T}_n^h$, we have

$$\begin{aligned} ||\varphi - \Pi_h\varphi||_{L^\infty(K)} &\leq c_d \max_{x\in K}\max_{\vec{v}\subset K} <\vec{v}, |\nabla^2\varphi(x)|\vec{v}> \\ &\leq c_d \max_{x\in K}\max_{\vec{e}\subset E_K} <\vec{e}, |\nabla^2\varphi|\vec{e}>, \end{aligned} \tag{56}$$

where $\nabla^2\varphi$ is the Hessian of φ, E_K is the set of edges of the element K, and c_d is a constant depending on the dimension d. This estimate implies that we can control the interpolation error on each element by controlling the size of edges of the element. This estimate lets us define an anisotropic metric for each function on each element, which leads to a small interpolation error on the element (see the work of Alauzet and Frey[69] for details). According to (54) and (55), the geometric error and the approximation error associated with the level-set function are conveniently bounded if the mesh is generated under the intersection of the metrics (see the work of Frey and George[62]) associated with ϕ, ϕ_0, and $\mathbf{u}$. Then, at each step, the new mesh is generated from this metric by using a Delaunay-based local mesh modification procedure (see the work of Ducrot and Frey[70] as well as sections 4 and 5.2 in the work of Dapogny et al[65] for more details).

Let us note that in the case of free-surface problems, we do not use a very sharp metric tensor outside the fluid far from the free surface. Nevertheless, in both problems, since we are especially interested in the behavior of the interface/free

[§]Let us note that for more precision, we must find an error estimate for the free-surface Navier-Stokes problem with slip boundary conditions, which is the actual problem treated in this work. Nevertheless, we consider the result in the work of Pironneau[35] on the time-dependent Navier-Stokes equation with Dirichlet boundary conditions since, to our knowledge, no error estimate result is available on free-surface Navier-Stokes equations.

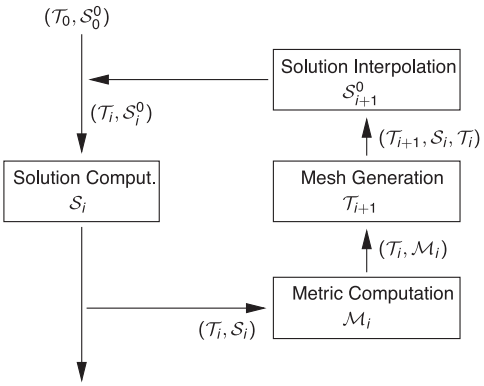

FIGURE 2 An illustrative scheme for different steps of the remeshing procedure. $\mathcal{T}_i$ (respectively, $\mathcal{S}_i$) represents the adapted mesh (respectively, the corresponding solution of the problem) at iteration i

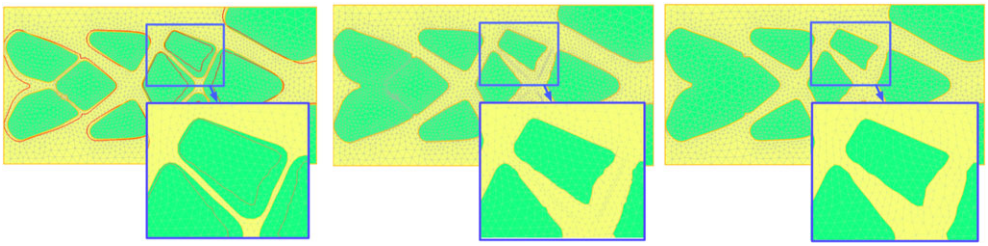

FIGURE 3 Left: initial fluid domains (green and yellow) with the zero level set of the level-set function associated to the evolved interface (red); middle: explicit (but ill shaped) discretization of the new interface; right: high-quality scheme for the new interface [Colour figure can be viewed at wileyonlinelibrary.com]

surface, the mesh elements in the vicinity of the free surface are taken small enough to give a precise approximation of the interface/free surface. Figure 2 illustrates different steps of the remeshing procedure based on the solution given by the algorithm.

Remark 5. As mentioned in Section 2.3, the evolution of the free surface/interface here is characterized by the evolution of the level-set function transported by the fluid velocity (ie, the solution of the level-set advection equation whose flux is constructed using the flow velocity). For this reason, the mesh adaptation at each time iteration of the algorithm is nothing but the explicit discretization and remeshing of the domain defined by the zero level set of the corresponding level-set function. This makes the algorithm efficient in dealing with topological changes. As an illustration, some topological changes happen in the test case presented in Section 5.6.

The 2 steps of mesh adaptation explained above are illustrated in Figure 3 (obtained from the work of Dapogny[71]). See also Figure 8 in the work of Bui et al[39] for another example containing topological changes.

4 | GLOBAL NUMERICAL SCHEME

In this part, we will describe the general scheme used in this work on the time period $[0, T]$. This scheme is based on the numerical tool presented in the previous section. We suppose that $[0, T]$ is divided by N interval $[t^{n-1}, t^n]$. The computational domain Ω is covered here by a time-dependent mesh $\mathcal{T}_h^n$ at iteration n. This mesh is adapted at each iteration based on the position of the fluids' domain, particularly based on the location of the free surface. The subscript h is omitted here at each iteration, but let us remember that the characteristic element size of h holds important meaning in our approach. As explained later, it is prescribed by the anisotropic metric tensor and always adapted at each time step; hence, the mobile unstructured meshes are generated with the scheme.

TABLE 1 Numerical scheme for bifluid flow over $[0, T]$

1. Start with mesh $\mathcal{T}^0$ and initialization $\mathbf{u}^0$,			
2. For $n = 0, \ldots, N-1$ do:			
	Problem	Input	Output
2.1	Regularization (option)	$(T^n, \mathbf{u}^n\|_\Gamma)$	$\tilde{\mathbf{u}}^n$
2.2	Redistancing	$\mathcal{T}^n$	$\tilde{\phi}^n$
2.3	Solving advection	$(\mathcal{T}^n, \tilde{\mathbf{u}}^n, \tilde{\phi}^n)$	ϕ^{n+1}
2.4	Adaptation	$(\mathcal{T}^n, \phi^{n+1})$	T^{n+1}
2.5	Interpolation	$(\mathbf{u}^n, \mathcal{T}^{n+1})$	$\mathbf{u}^n$
2.6	Solving Navier-Stokes	$(\mathcal{T}^{n+1}, \mathbf{u}^n)$	$\mathbf{u}^{n+1}$
3. Return $(\mathbf{u}^N, \phi^N, T^N)$			

4.1 | Two-phase flows

The algorithm starts with an initial discretization of the computational domain $\mathcal{T}^0$, an initial velocity flow $\mathbf{u}^0$ defined on the whole computational mesh if the flow is 2-phase Then, at each iteration $n = 0, \ldots, N-1$, we first regularize the discrete velocity $\mathbf{u}^n$ and denote the regularized velocity by $\tilde{\mathbf{u}}^n$. Let us note that in the case of 2-phase flows, this step is necessary only if the flow's velocity represents large discrepancies. We then generate the signed distance function ϕ^n of the discrete domain $\mathcal{T}^n$. Let us recall that this is a level-set function for $\mathcal{T}^n$ based on the discret approximation of the interface. This function is then transported by the resolution of the level-set advection equation defined using the regularized velocity $\tilde{\mathbf{u}}^n$ or using $\mathbf{u}^n$ if no regularization is necessary. The new level-set function is called $\tilde{\phi}^{n+1}$. It is supposed to characterize the discrete fluid domain $\mathcal{T}^{n+1}$ at time t^{n+1}. The mesh is then adapted according to this level-set function. We now have a new mesh $\mathcal{T}^{n+1}$. The velocity flow $\mathbf{u}^n$ is now interpolated on the new mesh $\mathcal{T}^{n+1}$. Finally, the time-discretized Navier-Stokes equation (32) (with boundary conditions (33)) is solved on $\mathcal{T}^{n+1}$. This overall algorithm is recapitulated hereafter in Table 1.

In comparison with previous studies in the work of Bui et al,[39] this scheme has been much reduced. In that approach, mesh adaptation needed 2 independent mesh, one for the resolution of the fluid equation and another for the advection of the interface. Therefore, an interpolation is necessary to correspond to each solving of the problem. The simplicity of the present scheme is due to the requirement of only one adapted mesh at each time step.

4.2 | Free-surface flows

The algorithm for free-surface flows is based on a similar algorithm as in the 2-fluid case. The main difference here is that the fluid equation is solved on the fluid domain only and not on the whole computational domain. Indeed, at each iteration, the Navier-Stokes equation is solved only on a strict subdomain $\mathcal{T}_1^n$ of $\mathcal{T}^n$, which corresponds to the fluid domain. Besides, the velocity extension step is necessary at each iteration. This step must be placed before the resolution of the Navier-Stokes equation, the reason being the convection term and its numerical treatment with the characteristic method. In fact, if the velocity extension step is placed after the fluid equation resolution, the velocity considered for the flow when the characteristic curve crosses the free surface and goes to the other side of the computational domain will be zero. This is not coherent with the physics of the system nor with the continuous mathematical model that represents it. Indeed, one of the features of this work is to compare the free-surface simulation with the bifluid simulation, where air represents 1 of the 2 phases. In other words, in the case of free-surface flows, the action of the second fluid is described by an atmospheric pressure. On the other hand, according to the interface condition, the flow velocity is continuous through the bifluid interface. This is why the consideration of a 0 value for velocity when the characteristic curve crosses the free surface is not convenient. Of course, in these latter cases, we could also take the value of the last point of the advection, but since we may need to extend the velocity for the advection equation, it would be more elegant to use the extended velocity to solve the Navier-Stokes equation in the fluid domain. This means that even though the Navier-Stokes equation is solved only on the fluid domain, the velocity that the Navier-Stokes solver takes as entry is the extended velocity defined on the whole computational mesh $\mathcal{T}^n$. However, only values of the extended velocity in the vicinity of the free surface and in the fluid domain are used to construct the right-hand side of the linear system (48). As a confirmation, we can check by numerical simulations that the behavior of the free surface is different if we give the nonextended velocity to the Navier-Stokes solver. Let us also mention that if we consider a Stokes flow, the placement of the velocity extension

TABLE 2 Numerical scheme for free-surface flows over $[0, T]$

1. Start with $\mathcal{T}$ and $\mathbf{u}^0$.			
2. For $n = 0, \ldots, N-1$,			
	Problem	Input	Output
2.1	Velocity extension	$(\mathcal{T}_2^n, \mathbf{u}^n)$	$\tilde{\mathbf{u}}^n$
2.2	Redistancing	$\mathcal{T}^n$	ϕ^n
2.3	Solving advection	$(\mathcal{T}^n, \tilde{\mathbf{u}}^n, \phi^n)$	$\tilde{\phi}^{n+1}$
2.4	Mesh adaptation	$(\mathcal{T}^n, \tilde{\phi}^{n+1})$	$\mathcal{T}^{n+1}$
2.5	Interpolation	$(\mathcal{T}^{n+1}, \tilde{\mathbf{u}}^n)$	$\tilde{\mathbf{u}}^n$
2.6	Solving Navier-Stokes	$(\mathcal{T}_1^{n+1}, \tilde{\mathbf{u}}^n)$	$(\mathbf{u}^{n+1}, p^{n+1})$
3. Return $\mathbf{u}^N, p^N, \phi^N, D^N$.			

step before the resolution of the fluid equation is not necessary. This is due to the absence of the convection term and the absence of the characteristic function on the right-hand side of the linear system in the case of Stokes flow.

The algorithm is summarized in Table 2.

Here, $\mathcal{T}_1^n$ (respectively, $\mathcal{T}_2^n$) represents the submeshes of the computational mesh $\mathcal{T}^n$, which corresponds to the fluid domain Ω_1 (respectively, to the vacuum part Ω_2). Let us recall that during the velocity regularization step, the Helmholtz equation (29) is solved on the whole computational mesh $\mathcal{T}^n$ to get a regularized velocity for the advection equation, which coincides with the physical velocity in the vicinity of the free surface. However, to solve the Navier-Stokes equations, we need the physical velocity on the whole fluid subdomain $\mathcal{T}_1^n$. For this reason, a regularization of the velocity field must be added before the advection step to ensure that the flow velocity does not present too many discrepancies between neighboring points. Let us also mention that the resolution of problem (27) is not expensive since the mesh elements of $\mathcal{T}_2^n$ are quite large. This is due to the fact that the value of the physical quantities as well as the extended velocity far from the fluid part are of few, if no, interest.

5 | NUMERICAL RESULTS

In this section, we present several numerical results obtained with our method. The first part contains numerical test cases entering the frame of bifluid problems. First, the Navier-Stokes solver for the monofluid has been validated by the lid-driven cavity test (see Section 5.1). Next, the test of a rising bubble and a Rayleigh-Taylor instability are investigated (see Sections 5.2 and 5.3). In the second part, the free-surface algorithm is simulated. The first example is a viscous fluid in a circular computational domain (see Section 5.4). Then, dam break examples for shampoo and water are presented (see Sections 5.5 and 5.6). The results in both 2 dimensions (2D) and 3 dimensions (3D) of these simulations are given in comparison with some results in other references. Let us note that, here, *hmin* and *hmax* are the prescribed minimal and maximal edge sizes of the mesh, whereas *hgrad* represents the graduation scale of the neighboring mesh elements.

5.1 | The lid-driven cavity problem

The lid-driven cavity problem is known as a standard benchmark for the Navier-Stokes solver in the numerical literature. Therefore, there are many references that can be used to validate our results. The problem corresponds to the flow confined in the unit domain $\Omega = [0, 1]^d (d = 2; 3)$ (the domain in 3 dimensions is given by extending the 2D domain in the z-direction with a unit width), and the Dirichlet boundary conditions are imposed on all boundaries: zero velocity everywhere except on the upper one. The fluid motion is then generated by the upper lid that moves in the x-direction with a constant velocity $u_x = 1$ m/s. The viscosity is adjusted to obtain the desired Reynolds number.

5.1.1 | Two-dimensional lid-driven cavity

In 2 dimensions, we investigated the simulations for Reynolds numbers from 100 up to 10 000. Four meshes have been used: a regular triangulation (carre2) with 2461 nodes and 5000 elements, a uniform triangulation (carre3) with 2143 nodes and 4136 elements, another uniform triangulation (carre4) with 8421 nodes and 16 544 elements, and a regular triangulation (carre7) with 10 201 nodes and 20 000 elements used for the test with high Reynolds numbers ($Re = 10\,000$). This problem involves a primary vortex at the cavity center and the vortices in the corners as Re increases. It is known

TABLE 3 Cavity in 2 dimensions: comparison between the positions of the main vortex for different Reynolds numbers

Reynolds	carre2	carre3	carre4	carre7	Ghia et al	NSIKE
100	$x = 0.595$				$x = 0.617$	$x = 0.610$
	$y = 0.736$				$y = 0.734$	$y = 0.750$
400	$x = 0.544$	$x = 0.544$	$x = 0.552$		$x = 0.554$	$x = 0.580$
	$y = 0.610$	$y = 0.615$	$y = 0.613$		$y = 0.606$	$y = 0.615$
1000	$x = 0.516$	$x = 0.515$	$x = 0.520$	$x = 0.521$	$x = 0.531$	$x = 0.545$
	$y = 0.569$	$y = 0.564$	$y = 0.570$	$y = 0.570$	$y = 0.562$	$y = 0.560$

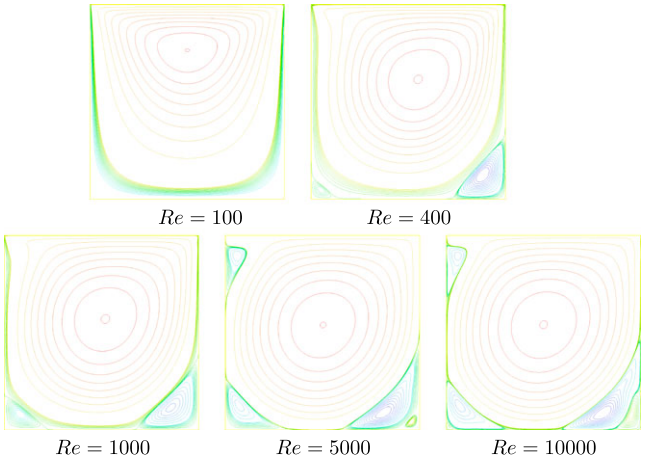

FIGURE 4 Cavity in 2 dimensions: streamlines for different Reynolds numbers [Colour figure can be viewed at wileyonlinelibrary.com]

that with the raise of Reynolds number, the number of vortices increases, and the position of the center of the primary vortex has the tendency to move from the right bottom corner toward the center of cavity. In Table 3, we resume the positions of the center of the primary vortices at the steady state (when the residual between the solutions reaches 10^{-6}) for $Re = 100, 400, 1000$. The streamlines are presented in Figure 4 for higher Reynolds numbers. We also compute the profiles of the velocity along the horizontal and vertical lines passing the geometric center of cavity (see Figure 5). Our numerical computations in these cases are compared to the results obtained in the very well-known works of Shin et al[72] and Gorazdi and Mohammadi[73] and the benchmark result for cavity flow in the work of Erturk et al[74] obtained with a fine uniform grid mesh of 601×601.

We have also obtained a good agreement between our pressure solution and the result showed in Hachem et al[75] (see Figure 6).

5.1.2 | Lid-driven cavity in 3D

Two meshes are employed to simulate the 3D problem: the large one consists of 11 037 vertices and 56 244 tetrahedra used for the cases of $Re = 100, Re = 400$, and the other one consists of 35 723 vertices and 193 586 tetrahedra used for $Re = 1000$. In many references, the results in 3D are compared with 2D results. Therefore, we simulate the 3D problem with the same Reynolds numbers as in 2D: $Re = 100, Re = 400, Re = 1000$. As expected, we observe that the streamlines at each plane $z = const$ correspond to those in 2D (see Figure 7 for example).

We also compare our numerical solutions with the results in the work of Ku et al[77] since the Reynolds numbers are exactly the same in our tests. However, no data for velocity profiles are available in the aforementioned work.[77] For this reason, the comparison is based on the images (see Figure 8, which shows a good agreement between our velocity profiles in 2D and 3D as well as those obtained in the aforementioned reference).

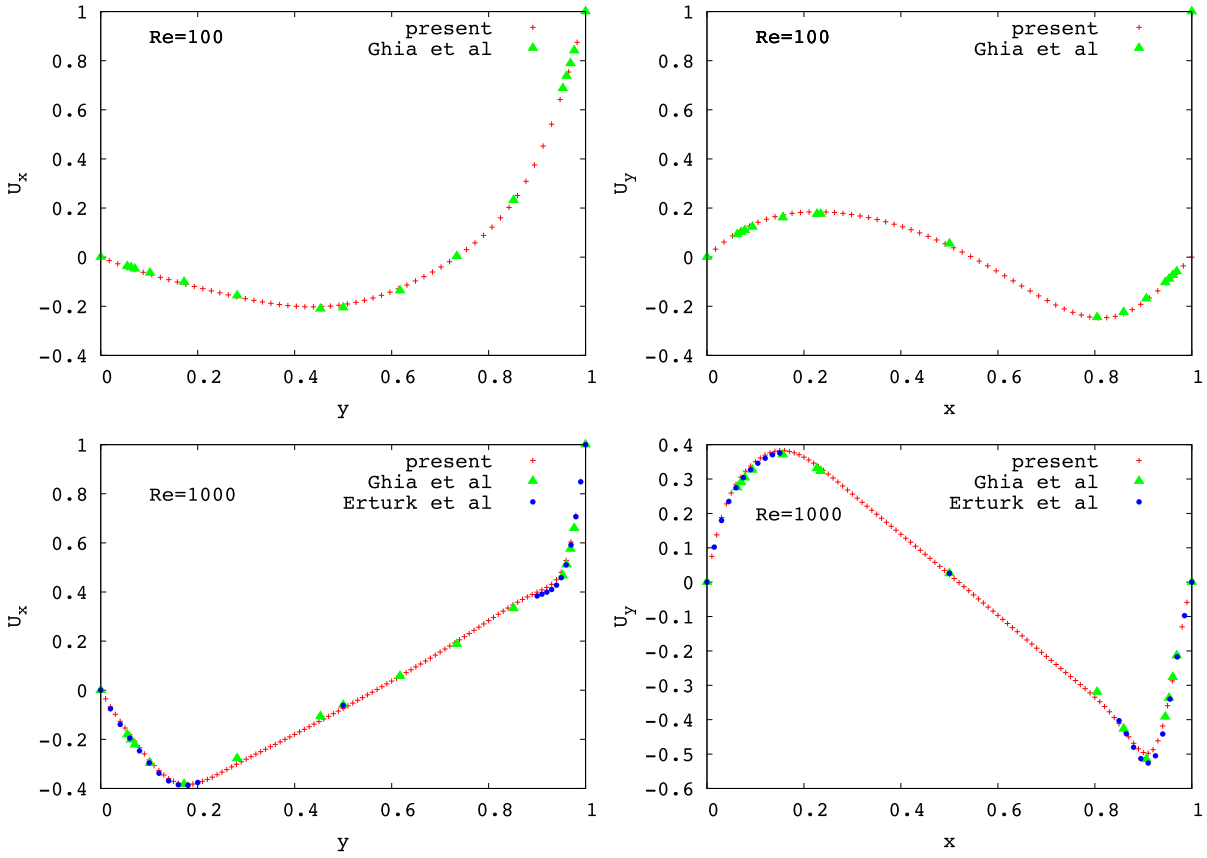

FIGURE 5 Cavity in 2 dimensions: velocity profile for u_x and u_y in the cases of $Re = 100$ and $Re = 1000$ [Colour figure can be viewed at wileyonlinelibrary.com]

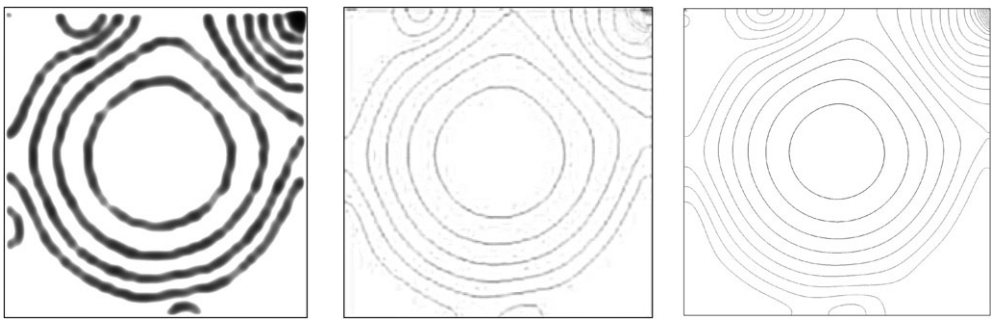

FIGURE 6 Cavity in 2 dimensions: isolines of pressure with $Re = 10\,000$. Left: result from the work of Gravemeier et al[76]; center: result from the work of Hachem et al[75]; right: present result

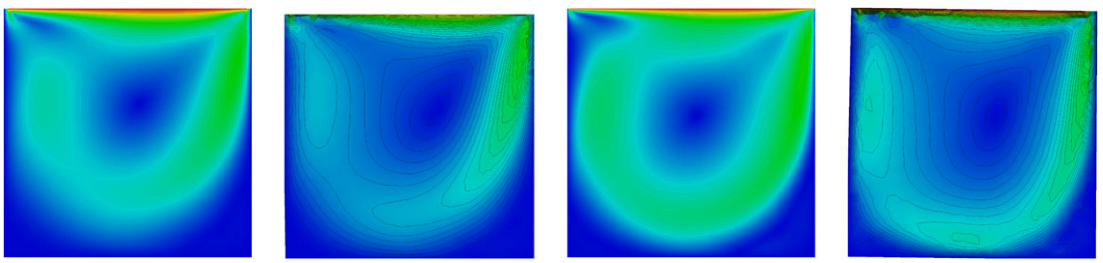

FIGURE 7 Cavity in 3 dimensions (3D): from left to right are streamlines in 2 dimensions and in the plane ($z = 0.5$) of 3D for $Re = 400$ (the first two) and $Re = 1000$ (the last two). The noise that appears at the top of the domain in 3D simulations is caused by mesh discretization in this region and by bad visualization (3D elements are cut; therefore, an interpolation is necessary) [Colour figure can be viewed at wileyonlinelibrary.com]

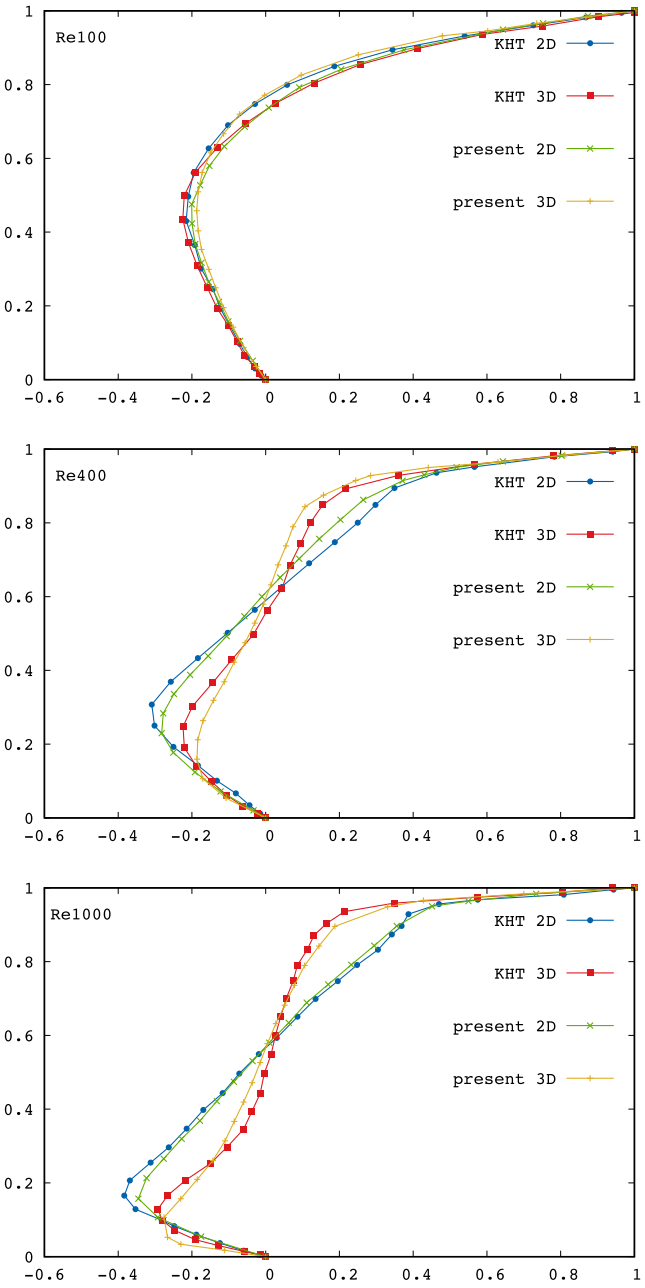

FIGURE 8 Cavity in 2 dimensions and 3 dimensions: comparison of the velocity profiles along the vertical centerline $x = 0.5$, present results, and results obtained from the work of Ku et al[77] [Colour figure can be viewed at wileyonlinelibrary.com]

5.2 | Rising bubble

We consider the raise and the deformation of a single bubble under gravity in a fluid contained in a vertical, rectangular domain. The density of the bubble is lower than the density of the surrounding fluid.

5.2.1 | Rising bubble in 2D

The initial configuration consists in a circular bubble of radius $r = 0.5$ centered at $[2, 1.5]$ in a 4×10 domain of 2494 nodes. The boundary condition considered here is the no-slip condition ($\mathbf{u} = 0$) on the horizontal walls and a free-slip ($\tau.\sigma\mathbf{n} = 0$ and $\mathbf{u}.\mathbf{n} = 0$) on the vertical walls (See Figure 9).

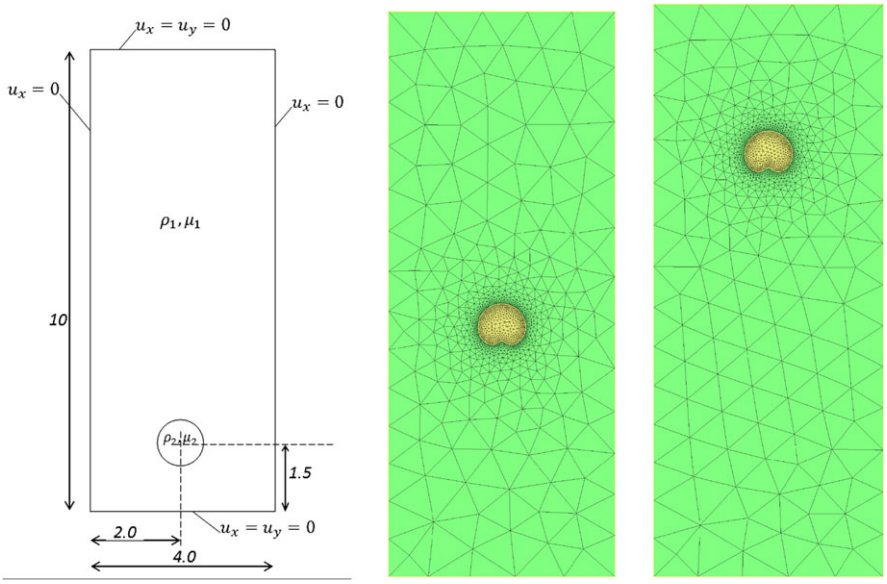

FIGURE 9 Rising bubble in 2 dimensions: initialization (left) and the evolution at $t = 5.0$ s (center) and $t = 10.0$ s [Colour figure can be viewed at wileyonlinelibrary.com]

In many references, different simulations are classified according to the Reynolds number and the Bond number (also called the Eotvos number) defined as follows:

$$Re = \frac{\rho_1 \sqrt{g}(2r)^{3/2}}{\mu_1}, \quad Bo = \frac{4\rho_1 g r^2}{\gamma}. \tag{57}$$

The problem has been set up with densities and viscosities given as follows: $\rho_1 = 100\ \text{kg}\cdot\text{m}^{-3}$, $\mu_1 = 0.1\ \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-1}$, $\rho_2 = 1.0\ \text{kg}\cdot\text{m}^{-3}$, and $\mu_2 = 0.01\ \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-1}$. The gravity is $g = 9.81e{-}3\text{m}\cdot\text{s}^{-2}$. The surface tension constant is $\gamma = 6.e{-}3N\cdot\text{m}^{-1}$, and the adapted mesh is such that $hmin = 0.02$ and $hmax = 1.0$ (as shown in Figure 9). It can be seen that the bubble shape deforms during the raise, and the terminal bubble shape is slightly dimpled at the bottom.

In order to impress the effect of the surface tension, we consider this simulation with different surface tension coefficients. It can be seen in Figure 10 that when tension surface is rather small here ($\gamma = 6e-5$), the bubble bottom becomes more dimpled while it remains flat in the case of more important coefficients ($\gamma = 2.5e{-}2$) and the bubble remains almost circular for larger coefficients ($\gamma = 9.e-2$). This result is in good agreement with fig. 6 in the work of Hua and Lou,[78] representing tests with a low Reynolds number and Bond number (from 10 to 200), which correspond to our simulations with different surface tension coefficients (see Equation (57)).

It is well known that level-set methods are not volume preserving. This is due to the advection of the interface using the advection equation. This, together with interpolation at each iteration, may lead to a variation of the total mass (or the total volume) of the bubble. The variation of the mass of the bubble is measured for this test case (cf Figure 11). This variation is of order 2×10^{-3} at each time step. Nevertheless, no correction is considered for this mass evolution since for the moment, no correction technique seems to be efficient enough to deal with this problem.

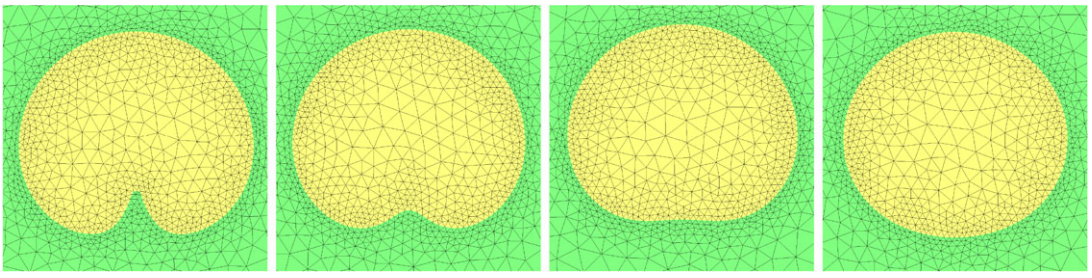

FIGURE 10 Rising bubble in 2 dimensions: interaction of the surface tension on the final bubble (at time $t = 10$ s) with different surface tension coefficients, from left to right: $\gamma = 6e-5$; $\gamma = 6e-3$; $\gamma = 2.5e-2$; $\gamma = 9e-2$ [Colour figure can be viewed at wileyonlinelibrary.com]

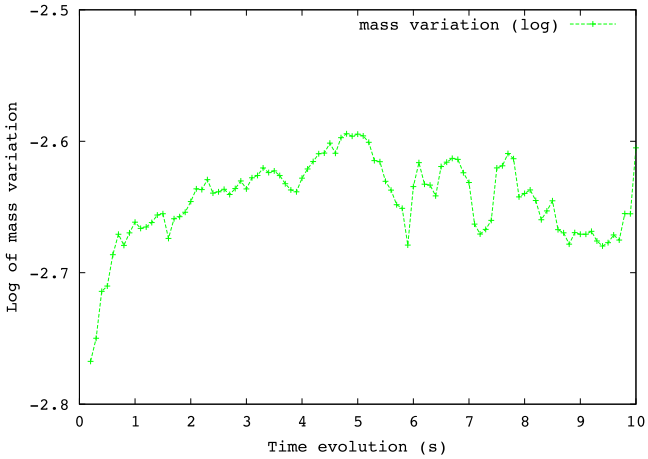

FIGURE 11 Rising bubble in 2 dimensions: variation of the mass of the bubble over time [Colour figure can be viewed at wileyonlinelibrary.com]

5.2.2 | Rising bubble in 3D

We simulate here a 3D bubble rising under the gravity. We consider the problem with similar conditions as in 2D: a bubble with a diameter of 0.5 m centers initially at $[0.75, 0.75, 1.0]$ in the domain $[0, 1.5] \times [0, 1.5] \times [0, 4.5]$. The aim of this simulation is the validation of our code for the 2-phase fluid in 3D by the examination of the bubble shapes during the evolution and the correction of the volume at each time step. As for the simulation with a low Reynolds number, we observe that the shape of the bubble deforms slowly from the beginning, it becomes a dimpled ellipsoidal, and gets more distorted over time. This result is similar to those in many references with corresponding Reynolds and Bond numbers (see the works of Amaya-Bower and Lee[79] and Hua et al[80] for instance). Figure 12 represents the evolution of the bubble from $t = 0$ to $t = 10$. It can be seen that when the bubble is very close to the upper wall of the domain, its shape is rapidly distorted. Figure 13 illustrates a zoomed-in view of the adapted mesh at time 6.0 after the distortion. We can observe that the mesh elements near the interface are smaller compared to other elements.

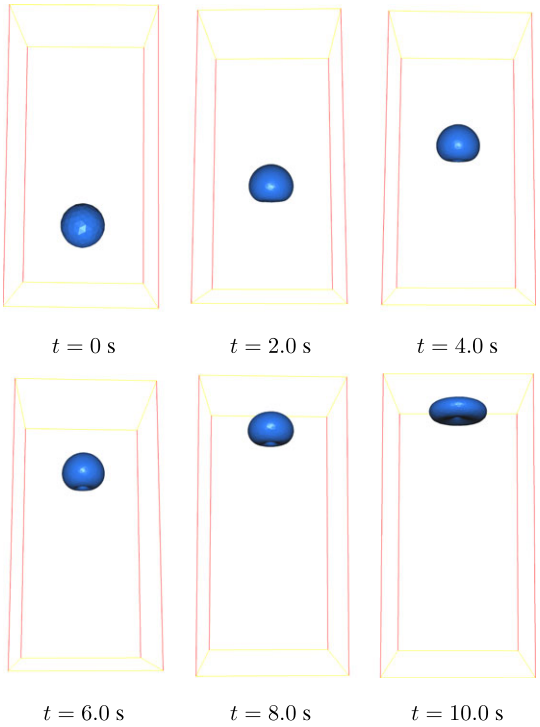

FIGURE 12 Rising bubble in 3 dimensions: evolution of the interface over time [Colour figure can be viewed at wileyonlinelibrary.com]

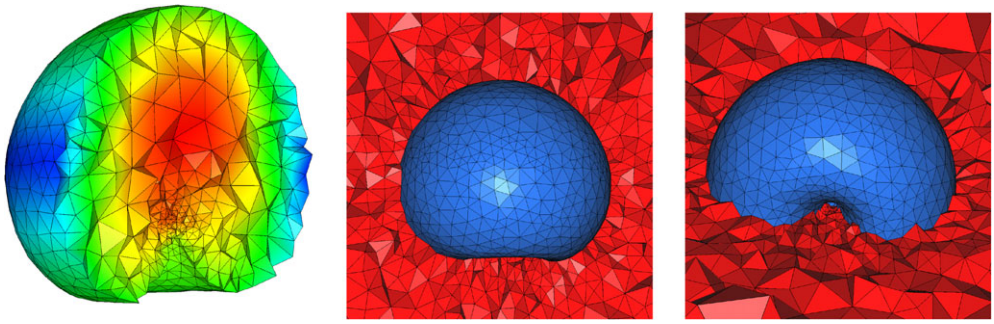

FIGURE 13 Rising bubble in 3 dimensions: zoomed-in view of the adapted mesh at time $t = 6.0$ s [Colour figure can be viewed at wileyonlinelibrary.com]

5.3 | Rayleigh-Taylor instability

In this section, we carry out the simulation of a more interesting problem named Rayleigh-Taylor instability. This instability occurs along the interface of 2 phases when the heavy fluid is superposed on the light one ($\rho_2 > \rho_1$) under the gravity field g. We compare our results with the results in the works of Tryggvason,[81] Fraigneau et al,[82] and Calgaro et al.[83] The criteria we consider here to parametrize the problem are the 2 following numbers.

- The density difference represented by the Atwood number defined as

$$At = \frac{\rho_2 - \rho_1}{\rho_2 + \rho_1}.$$

- The Reynolds number is defined by

$$Re = \frac{\rho_1 d^{\frac{3}{2}} g^{\frac{1}{2}}}{\mu},$$

where d is the width of the computational domain.

5.3.1 | Rayleigh-Taylor instability in 2D

Firstly, we set up the problem in the rectangular domain with a width of $d = 1$ and a height of $4d$. The no-slip conditions are imposed on the upper and lower boundaries, whereas the free-slip conditions are enforced on the vertical sides. The initial interface is set by

$$\tanh \frac{y - 2 - 0.1 \cos 2\pi x}{0.01\sqrt{2}} = 0.$$

We observe that the results satisfy a good symmetry over time while the Atwood number is 0.3. The results displayed in Figure 14 can be compared with those in fig. 8 in the work of Lee et al.[84] Let us notice that the mushroom shapes are not identical since their simulation is done with $At = 0.5$ and with a higher Reynolds number. We will investigate this test case in a reduced domain where the computational time is much less (Figures 15 and 16).

Figure 17 shows the results for different Atwood numbers at the same Reynolds number. We can observe that the mushroom is more roll-up if the Atwood number is higher. This result suggests that the effect of this number on the ratio of the width of the bubble and the spike fluid is in good agreement with the measures represented in fig. 9 in the work of Lee et al.[84] Secondly, assuming that the symmetry of the initial condition is preserved during the time evolution, we consider the problem on the reduced domain with a width of $d/2$ and a height of $2d$. We see again the same configurations with different Atwood numbers.

As mentioned, we also validate our code by the same values as in the work of Fraigneau et al.[82] The evolution of the interface is plotted in Figure 16 in the time scale of Tryggvason,[81] which is related to ours by $t_{\text{ref}} = t\sqrt{d.At.g}$. We can see that these results are qualitatively close to those in the work of Fraigneau et al[82] (and also with the results in the work of Calgaro et al[83]) presented in Figure 18. Indeed, the results are in good agreement at the early stage, and we can observe some slight discrepancies only at the large times of the evolution. In Figure 19, we see a zoomed-in view of the interface in both isotropic and anisotropic mesh adaptation cases. We observe that for the same precision, the size of elements of the anisotropic mesh is larger outside the neighborhood of the interface.

FIGURE 14 Rayleigh-Taylor instability in 2 dimensions: evolution of the interface over time with $At = 0.3$ [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 15 Rayleigh-Taylor instability in 2 dimensions: evolution of the interface with different Atwood numbers on the reduced domain [Colour figure can be viewed at wileyonlinelibrary.com]

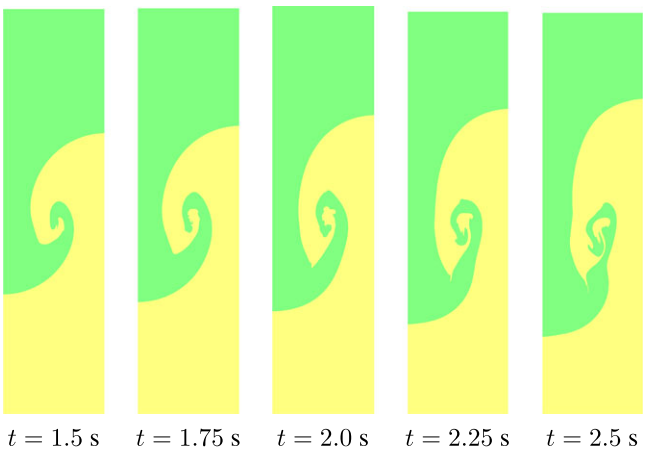

FIGURE 16 Rayleigh-Taylor instability in 2 dimensions: evolution of the interface with $At = 0.5$ and $Re = 1000$ [Colour figure can be viewed at wileyonlinelibrary.com]

5.3.2 | Rayleigh-Taylor instability in 3D

Finally, we consider the 3D computation of the Rayleigh-Taylor instability problem. All the computational conditions are exactly the same as the 2D case, and 3D instability has been well captured by the proposed scheme, as plotted in Figure 20.

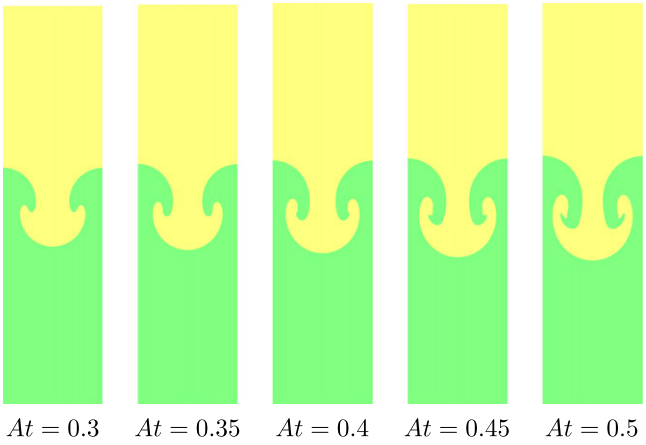

FIGURE 17 Rayleigh-Taylor instability in 2 dimensions: evolution of the interface with different Atwood numbers [Colour figure can be viewed at wileyonlinelibrary.com]

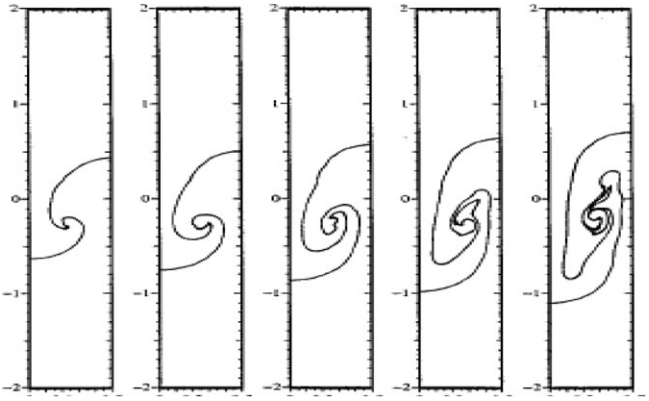

FIGURE 18 Rayleigh-Taylor instability in 2 dimensions: extracted results of evolution of the interface with $At = 0.5$ and $Re = 1000$ in the work of Fraigneau et al[82]

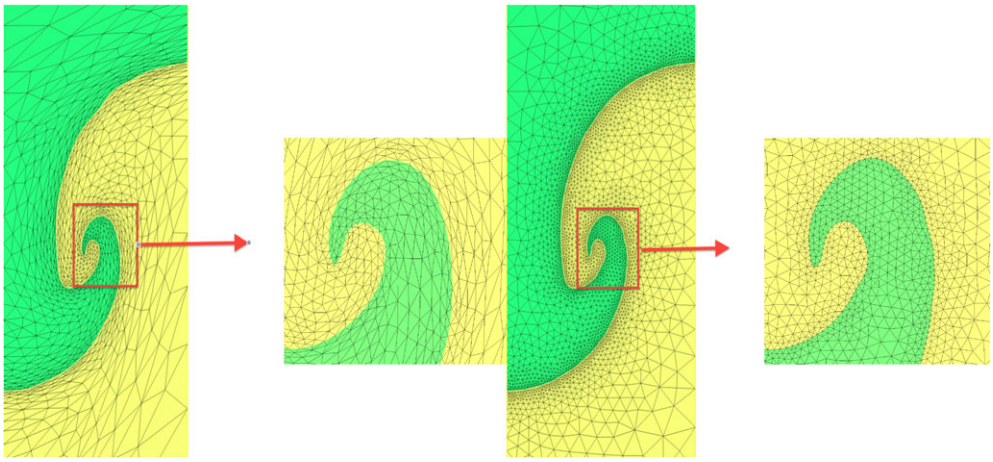

FIGURE 19 Rayleigh-Taylor instability in 2 dimensions: zoomed-in view of the adapted mesh in the vicinity of the interface; left: anisotropic mesh, right: isotropic mesh [Colour figure can be viewed at wileyonlinelibrary.com]

The results performed in Figure 20 are very close to the results presented in fig. 7 in the work of Li et al[85] or fig. 18 in the work of Li et al[86] (presented here in Figure 21). The slight difference of geometric configurations can be understood considering the fact that our computational fluid is incompressible while theirs are compressible. On the other hand, the density ratio is not identical. Therefore, a more detailed comparison with these results may not be convenient.

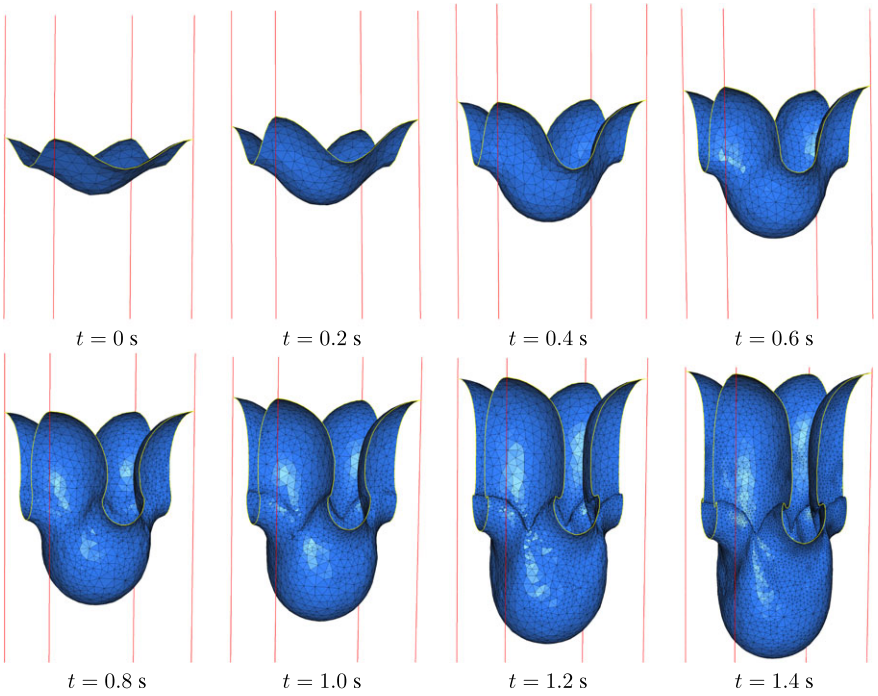

FIGURE 20 Rayleigh-Taylor instability in 3 dimensions: evolution of the interface in time with $At = 0.5$ [Colour figure can be viewed at wileyonlinelibrary.com]

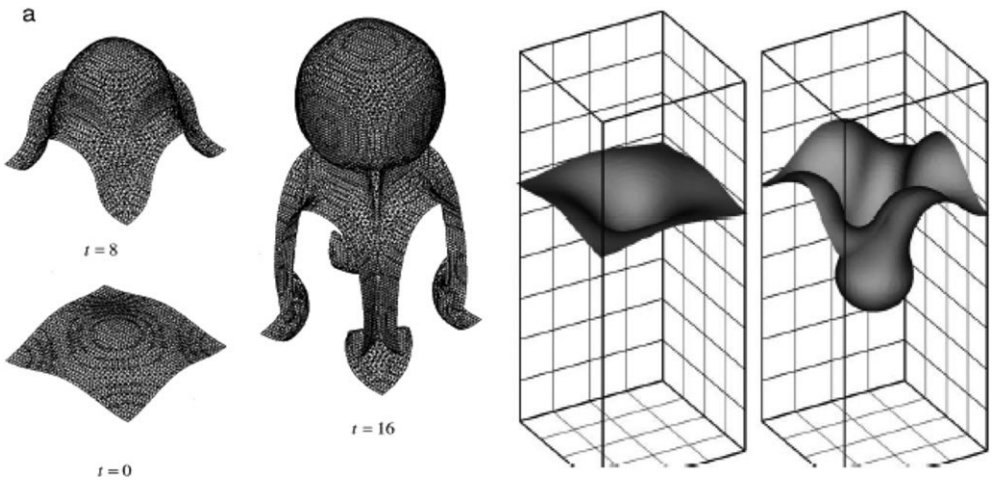

FIGURE 21 Rayleigh-Taylor instability in 3 dimensions: results obtained from the works of Li et al[85] (left) and Li et al[86] (right)

In this simulation, we begin with a mesh of 9.901 vertices (50 429 tetrahedra). The final mesh consists of 103.687 vertices (566.753 tetrahedra) with $hmin = 0.002$ and $hmax = 0.1$. Regarding the mesh adaptation, $hmin$ is decreased at each time step in order to well capture the interface with the minimal time of computation. Level-set method and anisotropic mesh adaptation may also be used for shape optimization in fluid mechanics based on a combination of classical shape derivative.

5.4 | Viscous flow in a circular domain

We consider here a fluid with density $\rho = 1$ and dynamic viscosity $\mu = 1$. The surface tension constant is $\gamma = 7.2$, and the gravity action is modeled by $f = -100$. No atmospheric pressure is considered, ie, the fluid is in the vacuum while a friction constant of $\alpha = 10$ is considered on the border of the computational domain in contact with the fluid. The initial solution is presented in Figure 22. Let us note that the initial solution is obtained by solving the steady Stokes equation with upward gravity.

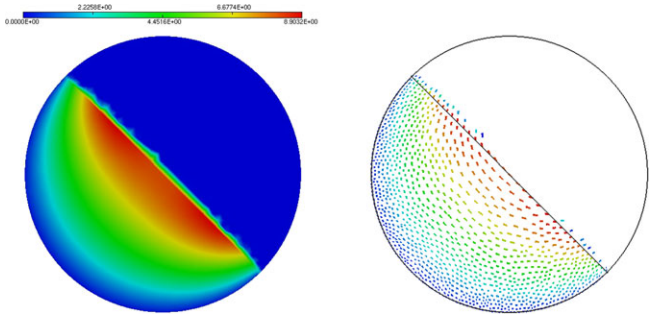

FIGURE 22 Norm of the initial velocity (on the left) and the associated vector field (on the right) [Colour figure can be viewed at wileyonlinelibrary.com]

The computational domain considered here is the unit circle, ie, the circle of radius 1 and the center $(0, 0)$. Let us note that in this test, the mesh elements all over the computational boundary, even out of the fluid domain, must be small enough to preserve the curvy shape of the computational domain. This is also due to the fact that the fluid domain is quite large (half of the computational domain). The characteristic element size of the mesh is about 0.03, and the time step is on the order of 10^{-3}. The test is done over the time interval $[0, 4.435]$ (over 3300 iterations). Figure 23 represents the solution at some time steps.

5.5 | Collapse of a shampoo column

We consider here a fluid with density $\rho = 1024$ and dynamic viscosity $\mu = 8$. The surface tension constant is $\gamma = 0.07$, and the gravity action is modeled by the force density $f = -9.8$. The fluid is subjected to no atmospheric pressure, ie, $p^a = 0$, and the friction constant is equal to $\alpha = 10^{-2}$. This friction is computed following the law

$$\alpha = \frac{\rho\,|\mathbf{U}|}{C_f^2},$$

where $C_f = 190$ as in the work of Cruchaga et al,[87] and $|\mathbf{U}|$ is the order of the norm of the flow velocity. The value that is taken here for $|\mathbf{U}|$ is 0.4. However, the result does not change significantly if we take, for instance, $|\mathbf{U}| = 0.04$ or $|\mathbf{U}| = 4$.

The initial fluid is a zero-velocity fluid column. The computational domain dimension as well as the fluid column width and height are the same as in the work of Cruchaga et al[87]: the computational domain is $0.42\,\text{m} \times 0.44\,\text{m}$, whereas the fluid domain is $0.114\,\text{m} \times 0.114\,\text{m}$. Figure 24 compares our results with the physical experiments and the numerical results presented in the work of Cruchaga et al.[87] The problem considered in the aforementioned work[87] is a bifluid air-shampoo problem. Therefore, the fluid equation is solved on the whole computational domain. Similarly to our work, the interface between 2 flows is captured by solving an advection equation (edge-track interface locator technique). The time step we consider here is $\Delta t = 0.01$ for the first 10 iterations (until time $t = 0.1$) and is $\Delta t = 0.02$ for other iterations. The values we consider for *hmin* and *hmax* are 0.0009 and 1.0. Moreover, $hgrad = 2.5$. The mesh adaptation at each step leads to large elements out of the fluid and fine elements inside (see Figure 25). Let us note that the time step in the work of Cruchaga et al[87] is $\Delta t = 0.001$. Nevertheless, the results are similar.

5.6 | Collapse of a water column

This test case is a dam break test case with water whose characteristics are $\rho = 1000$ and dynamic viscosity $\mu = 0.001$. The high value of the Reynolds number of this problem leads to some turbulent effects. As mentioned in the work of Mohammadi and Pironneau,[88] when the Reynolds number is large, there are usually strong gradients for the velocity and the vorticity in the vicinity of solid walls. For this reason, other models are suggested for this kind of problems (see, for instance, the work of Mohammadi and Pironneau[88]). The model we consider here is the model recalled in the work of Cruchaga et al.[87] In this simple model, the viscosity of the fluid is modified to take into account the energy dissipated by the turbulent effects. The law according to which the flow is modified is given by

$$\mu_{\text{mod}} = \min\left(\mu + l_{\text{mix}}^2 \rho \sqrt{\mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{u})/2}; \mu_{\max}\right),$$

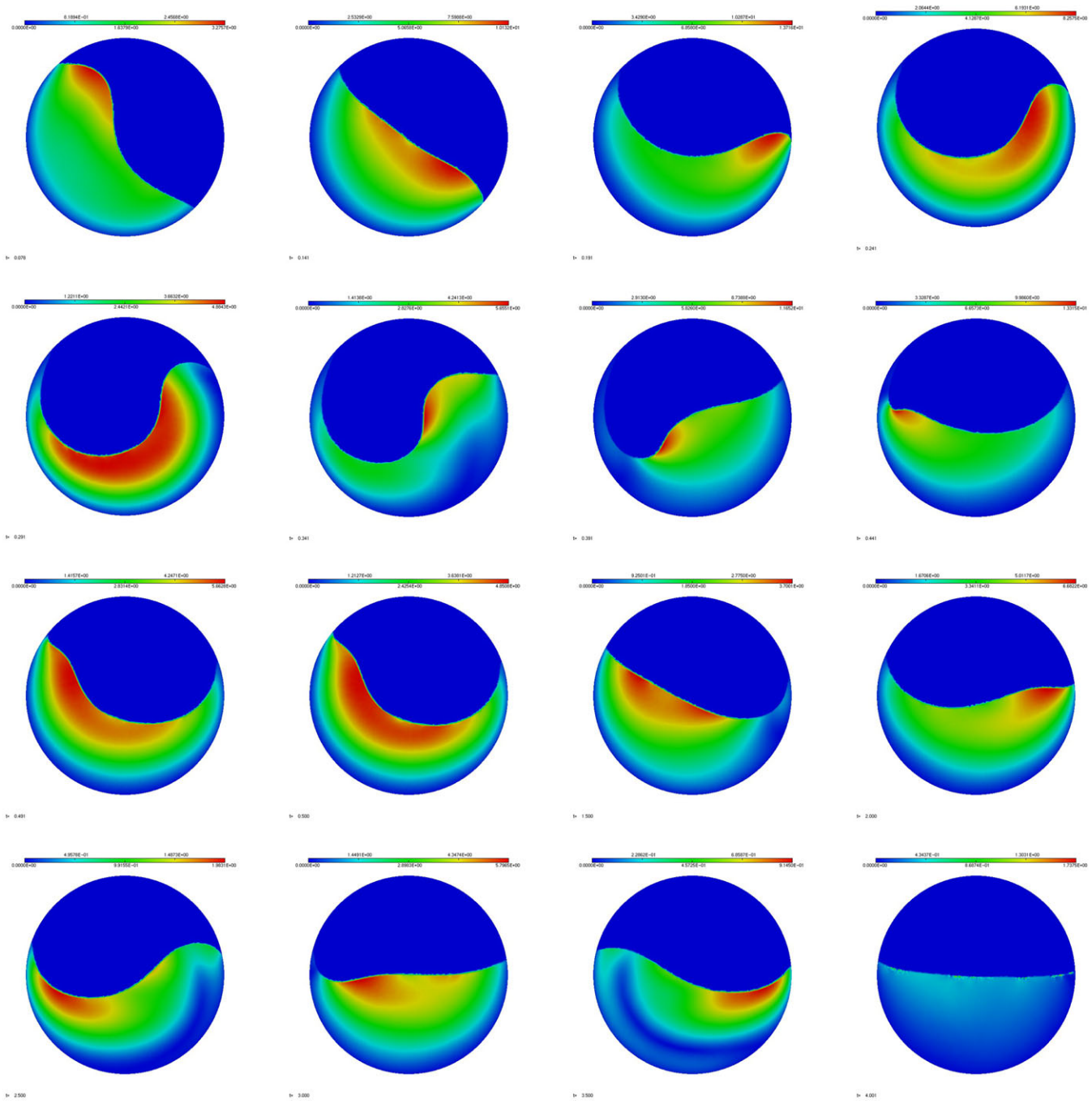

FIGURE 23 Evolution of a 2-dimensional viscous fluid in the unit circle. The equilibrium is reached at about time $T = 4$ [Colour figure can be viewed at wileyonlinelibrary.com]

where $l_{\text{mix}} = C_t h_{\text{UGN}}$ such that C_t is a modeling parameter, h_{UGN} is a characteristic element size, and $\mu_{\max}$ is a cut-off value. The values suggested in this test case for this parameters are

$$C_t = 3.57, \quad h_{\text{UGN}} = hmin = 9 \times 10^{-4}, \quad \mu_{\max} = 1.5.$$

In practice, we take here $\mu = 0.001$ and $\Delta t = 0.001$ for the first iteration of the algorithm. Then, we take the cut-off value $\mu_{\max}$ for viscosity and set $\Delta t = 0.005$ for others. Boundary conditions considered here are slip boundary conditions without friction. Besides, no atmospheric pressure is taken into account. Other parameters are set as $hmax = 1$ and $hgrad = 2.5$. Figure 26 shows this numerical test compared to the result presented in the work of Cruchaga et al.[87]

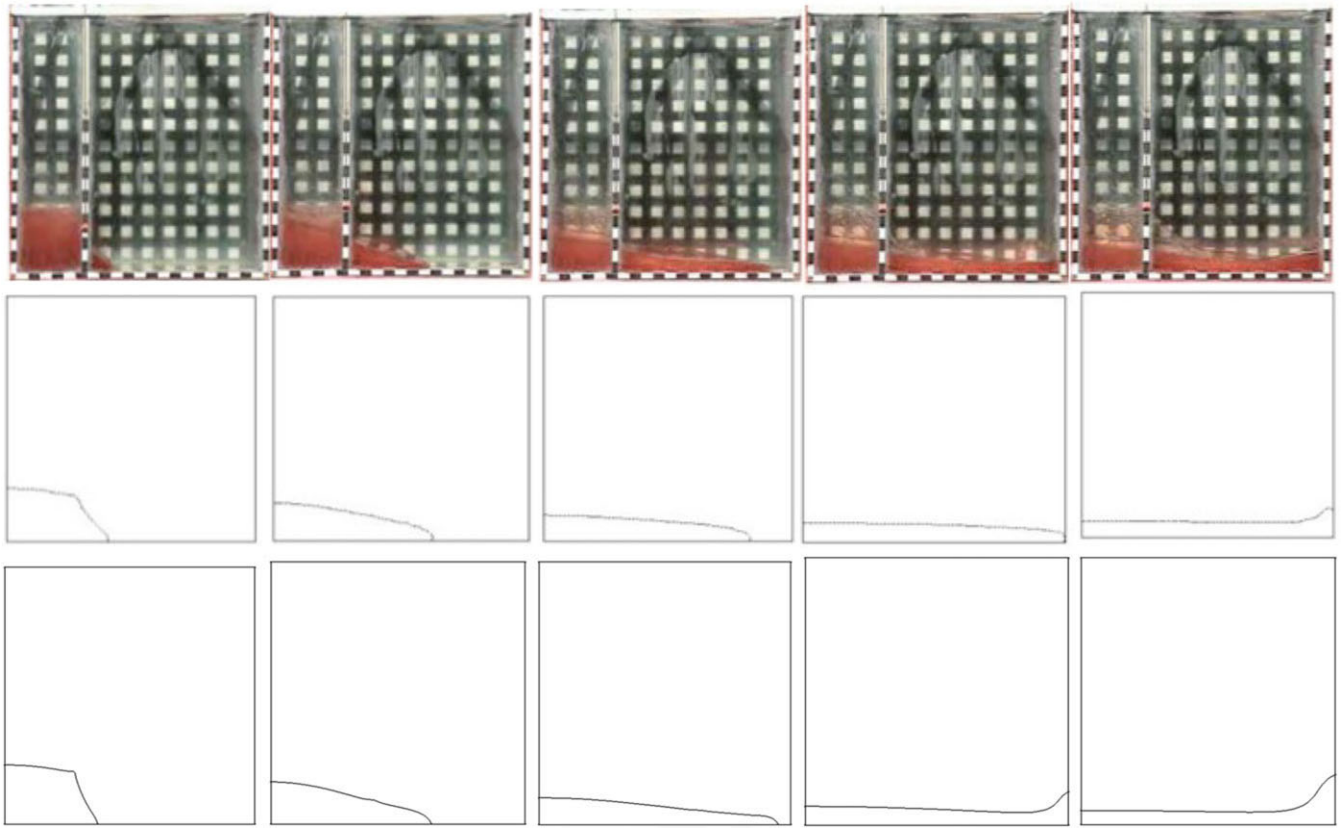

FIGURE 24 Comparison between the physical experiments in the work of Cruchaga et al[87] (top), the numerical result from the aforementioned work[87] with the wall friction (middle) and our results (bottom) at times $t = 0.1, 0.2, 0.3, 0.4, 0.5$ (from left to right) [Colour figure can be viewed at wileyonlinelibrary.com]

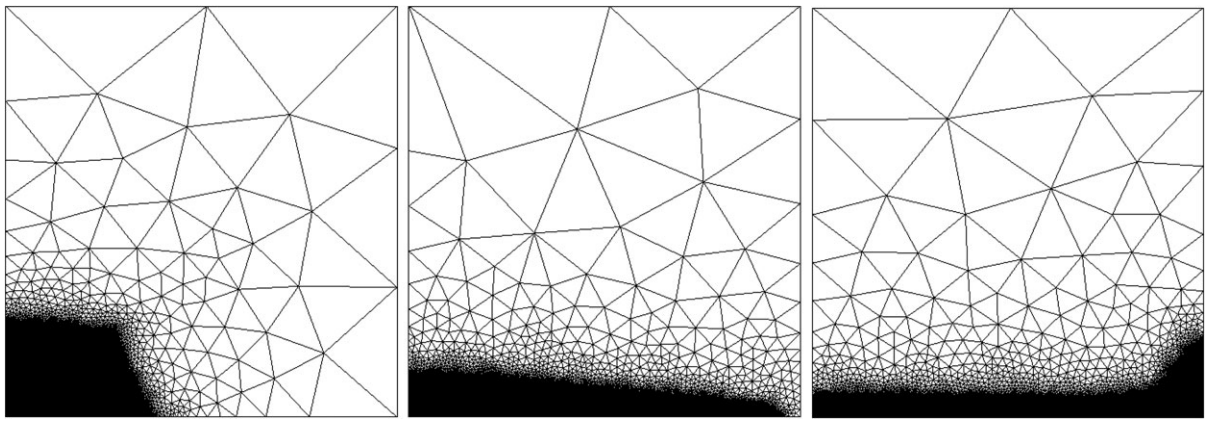

FIGURE 25 Adapted mesh for the dam break with shampoo at times $t = 0.1$ (left), $t = 0.3$ (middle), and $t = 0.5$ (right). We see that the mesh elements' size is small inside the fluid, very small in the vicinity of the free surface, and large out of the fluid far from the surface

Remark 6.

1. In all dam break test cases, simulations are proved to be very sensitive to the time step. Indeed, even small variations of the time step play a very important role in the behavior of the fluid. Moreover, for a fixed size of the mesh elements, a too small time step may lead to an incoherent simulation. This may be eventually due to the presence of the time step in a denominator of the right-hand side of the error estimate of the problem.
2. We can also see in this test case that the method deals well with topological changes of the fluid domain. Figure 27 shows the fluid domain and the adapted mesh at times where a breaking wave occurs. Figure 28 shows the mesh with a zoomed-in view on the breaking wave just before and after a topological change of the fluid domain.

FIGURE 26 Comparison between the physical experiments in the work of Cruchaga et al[87] (left column), the numerical result from the aforementioned work[87] with a discontinuity capturing interface dissipation (middle column), and our results (right column) at times $t = 0.1, 0.2, 0.3, 0.4, 0.5$ (top left), at times $0.6, 0.7, 0.8, 0.9, 1.0$ (top right), and at times $1.1, 1.2, 1.3, 1.4, 1.5$ (bottom) (from top to bottom) [Colour figure can be viewed at wileyonlinelibrary.com]

5.7 | Numerical convergence in space of the algorithm

The objective of this test case is to obtain an idea of the order of convergence in space of the algorithm. In other words, we want to quantify the evolution of the numerical solution as a function of the characteristic size of the mesh, at large times, for a given time step.

To do so, we consider again the test case with shampoo presented in Section 5.5 where the computational domain is quite small (particularly compared to the test case of Section 5.4). The fluid characteristics as well as the friction coefficient

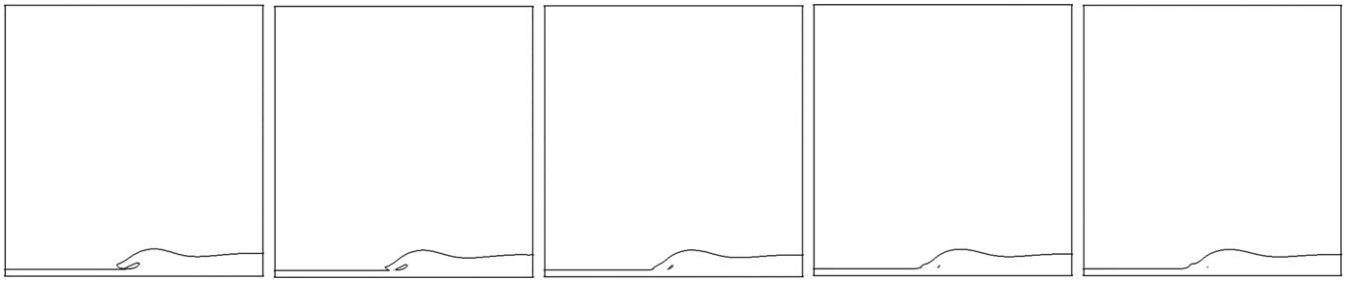

FIGURE 27 Geometric shape of the fluid domain, in the test case presented in Section 5.6, at times 0.741, 0.746, 0.756, 0.761, 0.766 (from left to right)

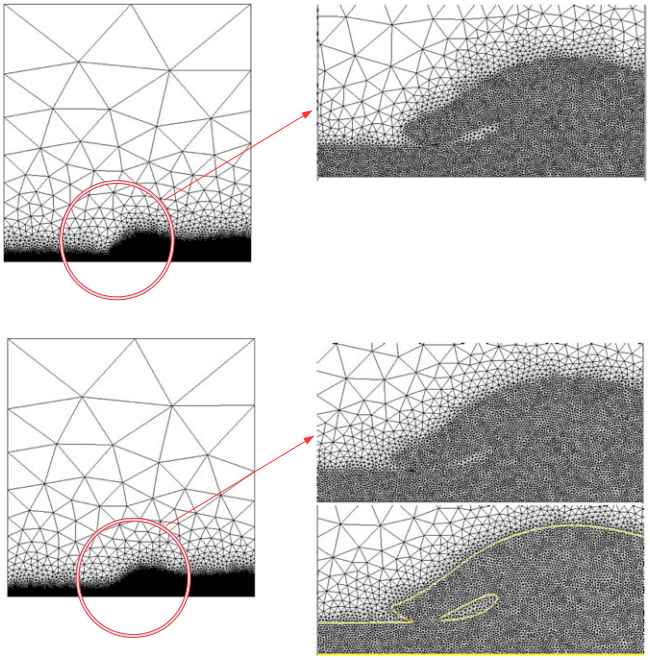

FIGURE 28 Zoomed-in view of the mesh associated to the fluid domain in the test case presented in Section 5.6, at time 0.741 just before (left) and after (right) a topological change occurs [Colour figure can be viewed at wileyonlinelibrary.com]

and the time step, $h_{\max}$ and h_{grad}, are exactly the same as in Section 5.5. Figure 29 shows the $\mathbf{L}^2$ norm[¶] of the velocity flow at $T = 6.160$, where the equilibrium is reached, for different values of $h_{\min}$. According to the Figure, the order of the convergence following $h_{\min}$ seems to be linear. Nevertheless, it is important to remark that since the adapted mesh is unstructured, $h_{\min}$ does not necessarily represent the characteristic size of the edges. In addition, since the problem is time dependent (the error estimates presented in Section 3.5 are both time step and space step dependent), the choice of the time step may influence the behavior of the $\mathbf{L}^2$ norm of the solution as a function of $h_{\min}$.

Remark 7. Let us remark that no exact solution of the free-surface Navier-Stokes equations is available. This is why we considered a test case where the long-time physical solution of the problem is supposed to be the equilibrium, ie, the null velocity flow everywhere on the fluid domain. Indeed, even though we do not know the exact solution of this test cas, we know that, from a physical point of view, the system will reach the equilibrium for enough large times. Obviously, the $\mathbb{L}^2$ norm of the velocity flow corresponding to the physical equilibrium is 0. For this reason, we computed the $\mathbb{L}^2$ norm of the numerical equilibrium reached by the algorithm (at an enough large time $T = 6.160$)

[¶]As mentioned in Section 3.3.5 (more precisely, in the fist paragraph of page 16), we use the $\mathbb{P}^1$-bubble/$\mathbb{P}^1$ finite element method for the numerical resolution of the time-discretized Navier-Stokes equations on the fluid domain. This is to say that the numerical velocity field at each iteration is a $\mathbb{P}^1$-bubble function defined on the mesh approximating the fluid domain. Therefore, it defines a piecewise first-order polynomial function on this mesh. Hence, using the values of this function on the degrees of freedom of the mesh, we are able to compute its $\mathbb{L}^2$ norm by using exact integration formulas for first-order polynomial functions defined on triangles.

Minimal size of the edge (h_{min})	$\mathbf{L}^2$ norm of velocity
0.0007	0.0015277
0.0011	0.00165902
0.0012	0.00169476

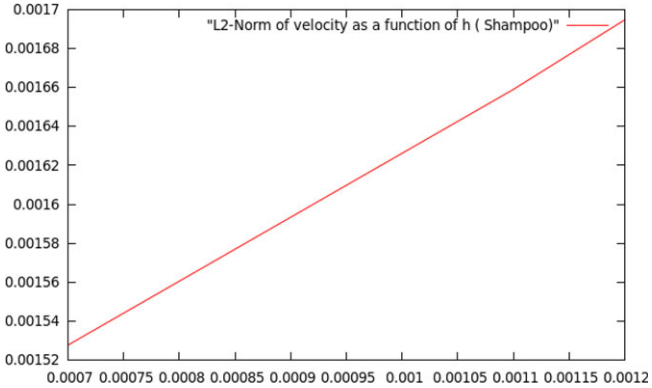

FIGURE 29 $\mathbf{L}^2$ norm of velocity flow at equilibrium for different mesh sizes as a function of $h_{\min}$ for the dam break with the shampoo test case [Colour figure can be viewed at wileyonlinelibrary.com]

to compare the difference between the numerical equilibrium and the physical one. The plot presented in Figure 29 is obtained based on this difference for different sizes of the mesh.

ORCID

D. Kazerani http://orcid.org/0000-0002-4976-4744

REFERENCES

1. Gross S, Reusken A. *Numerical Methods for Two-Phase Incompressible Flows*. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2011.
2. Anderson DM, McFadden GB, Wheeler AA. Diffuse-interface methods in fluid mechanics. *Ann Rev Fluid Mech*. 1998;30:139-165.
3. Tryggvason G, Bunner B, Esmaeeli A, et al. A front-tracking method for the computations of multiphase flow. *J Comput Phys*. 2001;169:708-759.
4. Barré de Saint-Venant AJC. Théorie du mouvement non permanent des eaux, avec application aux crues des rivières et à l'introduction des marées dans leur lit. *C R Acad Sci Paris*. 1871;73:147-154.
5. Green AE, Naghdi PM. A derivation of equations for wave propagation in water of variable depth. *J Fluid Mech*. 1976;78(2):237-246.
6. Gerbeau JF, Perthame B. Derivation of viscous Saint-Venant system for laminar shallow water; numerical validation. *Discret Contin Dyn Syst Ser B*. 2001;1(1):89-102.
7. Lannes D. *The Water Waves Problem: Mathematical Analysis and Asymptotics*. Vol. 188. Providence, RI: American Mathematical Society; 2013.
8. Denisova I. Problem of the motion of two viscous incompressible fluids separated by a closed free interface. *Acta Appl Math*. 1994;37(1-2):31-40.
9. Prüss J, Simonett G. On the two-phase Navier-Stokes equations with surface tension. *Interfaces Free Bound*. 2010;12(3):311-345.
10. Desjardins B. Regularity results for two-dimensional flows of multiphase viscous fluids. *Arch Ration Mech Anal*. 1997;137(2):135-158.
11. Tanaka N. Global existence of two-phase nonhomogeneous viscous incompressible fluid flow. *Comm Partial Differ Equ*. 1993;18(1-2):41-81.
12. Allain G. Un problème de Navier-Stokes avec surface libre et tension superficielle. *Ann Fac Sci Toulouse Mat (5)*. 1985;7(1):29-56.
13. Allain G. Small-time existence for the Navier-Stokes equations with a free surface. *Appl Math Optim*. 1987;16(1):37-50.
14. Tani A, Tanaka N. Large-time existence of surface waves in incompressible viscous fluids with or without surface tension. *Arch Ration Mech Anal*. 1995;130(4):303-314.
15. Tani A. Small-time existence for the three-dimensional Navier-Stokes equations for an incompressible fluid with a free surface. *Arch Ration Mech Anal*. 1996;133(4):299-331.
16. Xu L, Zhang P, Zhifei Z. Global solvability of a free boundary three-dimensional incompressible viscoelastic fluid system with surface tension. *Arch Ration Mech Anal*. 2013;208(3):753-803.
17. Unverdi SO, Tryggvason G. A front-tracking method for viscous incompressible multi-fluid flows. *J Comput Phys*. 1992;100(1):25-37.
18. Esmaeeli A, Tryggvason G. Direct numerical simulations of bubbly flows. Part 1. Low Reynolds number arrays. *J Fluid Mech*. 1998;377:313-345.

19. Esmaeeli A, Tryggvason G. Direct numerical simulations of bubbly flows. Part 2. Moderate Reynolds number arrays. *J Fluid Mech*. 1999;385:325-358.
20. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J Comput Phys*. 1997;135(2):203-216.
21. Bänsch E. Finite element discretization of the Navier-Stokes equations with a free capillary surface. *Numer Math*. 2001;88(2):203-235.
22. Behr M. Stabilized space-time finite element formulations for free-surface flows. *Comm Numer Meth Eng*. 2001;17(11):813-819.
23. Ganesan S, Tobiska L. A coupled arbitrary Lagrangian-Eulerian and Lagrangian method for computation of free surface flows with insoluble surfactants. *J Comp Phys*. 2009;228(8):2859-2873.
24. Giga Y, Takahashi S. On global weak solutions of the nonstationary two-phase Stokes flow. *SIAM J Math Anal*. 1994;25(3):876-893.
25. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *J Comput Phys*. 1981;39(1):201-225.
26. Scardovelli R, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *Ann Rev Fluid Mech*. 1999;31(1):567-603.
27. Gerlach D, Tomar G, Biswas G, Durst F. Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows. *Int J Heat Mass Transf*. 2006;49(3-4):740-754.
28. Tomé MF, Filho AC, Cuminato JA, Mangiavacchi N, McKee S. GENSMAC3D: a numerical method for solving unsteady three-dimensional free surface flows. *Int J Numer Methods Fluids*. 2001;37(7):747-796.
29. de Sousa FS, Mangiavacchi N, Nonato LG, et al. A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces. *J Comput Phys*. 2004;198(2):469-499.
30. Dervieux A, Thomasset F. A finite element method for the simulation of a Rayleigh-Taylor instablity. *Approximation Methods for Navier-Stokes Problems. Lecture Notes in Mathematics*. Vol. 771. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 1980:145-158.
31. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J Comput Phys*. 1988;79(1):12-49.
32. Crandall M, Ishii H, Lions P-L. User's guide to viscosity solutions of second order partial differential equations. *Bull Amer Math Soc*. 1992;27(1):1-67.
33. Sethian JA. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. New York, NY: Cambridge University Press; 1999.
34. Benqué JP, Ibler B, Keramsi A, Labadie G. A finite element method for the Navier-Stokes equations. Paper presented at: Proceedings of the 3rd International Conference on Finite Elements in Flow Problems; 1980; Banff, Canada.
35. Pironneau O. On the transport-diffusion algorithm and its applications to the Navier-Stokes equations. *Numerische Mathematik*. 1982;38(3):309-332.
36. Agresar G, Linderman JJ, Tryggvason G, Powell KG. An adaptive, Cartesian, front-tracking method for the motion, deformation and adhesion of circulating cells. *J Comput Phys*. 1998;143(2):346-380.
37. Enright D, Fedkiw R, Ferziger J, Mitchell I. A hybrid particle level set method for improved interface capturing. *J Comput Phys*. 2002;183(1):83-116.
38. Min C, Gibou F. A second order accurate level set method on non-graded adaptive cartesian grids. *J Comput Phys*. 2007;225(1):300-321.
39. Bui TTC, Frey P, Maury B. A coupling strategy based on anisotropic mesh adaptation for solving two-fluid flows. *Int J Numer Methods Fluids*. 2010;66(10):1226-1247.
40. Chang YC, Hou TY, Merriman B, Osher S. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *J Comput Phys*. 1996;124(2):449-464.
41. Sussman M, Fatemi E, Smereka P, Osher S. An improved level set method for incompressible two-fluid flows. *Comput Fluids*. 1998;27(5-6):663-680.
42. Aujol JF, Aubert G. Signed Distance Functions and Viscosity Solutions of Discontinuous Hamilton-Jacobi Equations [PhD dissertation]. Villeneuve d'Ascq, France: INRIA; 2002.
43. Dapogny C, Frey P. Computation of the signed distance function to a discrete contour on adapted triangulation. *Calcolo*. 2012;49(3):193-219.
44. Crouzeix M, Raviart PA. Conforming and nonconforming finite element methods for solving the stationary Stokes equations I. *ESAIM: Math Model Numer Anal*. 1973;7(R3):33-75.
45. Boris JP, Book DL. Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. *J Comput Phys*. 1973;11(1):38-69.
46. Lesaint P, Raviart PA. Résolution numérique de l'équation de continuité par une méthode du type éléments finis. *Publications Mathématiques et Informatique de Rennes*. (S5) 1976:1-10.
47. Pironneau O. *The Finite Element Methods for Fluids*. Hoboken, NJ: John Wiley & Sons; 1989.
48. Pironneau O. Finite element characteristic methods requiring no quadrature. *J Sci Comput*. 2010;43(3):402-415.
49. Sethian JA, Smereka P. Level set methods for fluid interfaces. *Ann Rev Fluid Mech*. 2003;35(1):341-372.
50. Sussman M, Smereka P, Osher SJ. A level set approach to computing solutions to incompressible two-phase flow. *J Comput Phys*. 1994;114(1):146-159.
51. Burger M. A framework for the construction of level-set methods for shape optimization and reconstruction. *Interfaces Free Boundaries*. 2003;5(3):301-329.
52. de Gournay F. Velocity extension for the level-set method and multiple eigenvalues in shape optimization. *SIAM J Control Optim*. 2006;45(1):343-367.

53. Glowinski R. Finite element methods for incompressible viscous flows. In: *Handbook in Numerical Analysis*. Vol. 9. Amsterdam, The Netherlands: North-Holland; 2003.
54. Verfürth R. Finite element approximation of steady Navier-Stokes equations with mixed boundary conditions. *ESAIM: Math Model Numer Anal*. 1985;19(3):461-475.
55. Verfürth R. Finite element approximation of incompressible Navier-Stokes equations with slip boundary condition. *Numerische Mathematik*. 1987;50(6):697-721.
56. Verfürth R. Finite element approximation of incompressible Navier-Stokes equations with slip boundary condition, II. *Numerische Mathematik*. 1991;59(1):615-636.
57. Dione I, Urquiza JM. Penalty: finite element approximation of Stokes equations with slip boundary conditions. *Numerische Mathematik*. 2015;129(3):587-610.
58. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods. Springer Series in Computational Mathematics*. Vol. 15. New York, NY: Springer-Verlag; 1991.
59. Ern A, Guermond JL. Theory and practice of finite elements. *Applied Mathematical Sciences*. Vol. 159. New York, NY: Springer Science + Business Media; 2004.
60. Quarteroni A. *Numerical Models for Differential Problems. Modeling, Simulation and Applications*. Vol. 2. Milan, Italy: Springer; 2009.
61. Arrow KJ, Hurwicz L, Uzawa H, Chenery HB. *Studies in Linear and Non-Linear Programming*. Vol. 2. Redwood, CA: Stanford University Press; 1958.
62. Frey P, George PL. *Mesh Generation Application to Finite Elements*. 2nd ed. Hoboken, NJ: John Wiley & Sons; 2008.
63. Frey P. About surface remeshing. Paper presented at: Proceedings of the 9th International Meshing Roundtable; 2000; New Orleans, LA.
64. Dapogny C, Frey P. Computation of the signed distance function to a discrete contour on adapted triangulation. *Calcolo*. 2011;49(3):193-219.
65. Dapogny C, Dobrzynski C, Frey P. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *J Comput Phys*. 2014;262:358-378.
66. Lorensen WE, Cline HE. Marching cubes: a high resolution 3D surface construction algorithm. *ACM SIGGRAPH Comp Graph*. 1987;21(4):163-169.
67. Bui C, Dapogny C, Frey P. An accurate anisotropic adaptation method for solving the level set advection equation. *Int J Numer Methods Fluids*. 2012;70(7):899-922.
68. Frey PJ, Alauzet F. Anisotropic mesh adaptation for CFD computations. *Comput Methods Appl Mech Eng*. 2005;194(48-49):5068-5082.
69. Alauzet F, Frey P. Estimateur D'Erreur Géométrique et Métriques Anisotropes Pour l'Adaptation de Maillage. Partie I: Aspects Théoriques [PhD dissertation]. Villeneuve d'Ascq, France: INRIA; 2003.
70. Ducrot V, Frey P. Anisotropic level set adaptation for accurate interface capturing. Paper presented at: Proceedings of the 17th International Meshing Roundtable. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2008.
71. Dapogny C. Shape Optimization, Level Set Methods on Unstructured Meshes and Mesh Evolution [PhD Dissertation]. Paris, France: Universite Pierre et Marie Curie; 2012.
72. Shin CT, Ghia U, Ghia KN. High-resolution for incompressible flow using the Navier-Stokes equations and a multigrid method. *J Comput Phys*. 1982;48(3):387-411.
73. Gorazd M, Mohammadi B. NSIKE- an incompressible Navier-Stokes solver for unstructured meshes. Rocquencourt, France: INRIA; 1999.
74. Erturk E, Corke TC, Gokcol C. Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers. *Int J Numer Methods Fluids*. 2005;48(7):747-774.
75. Hachem E, Rivaux B, Kloczko T, Diggonet H, Coupez T. Stabilized finite element method for incompressible flows with high Reynolds number. *J Comput Phys*. November 2010;229(23):8643-8665.
76. Gravemeier V, Wall WA, Ramm E. A three-level finite element method for the instationary incompressible Navier-Stokes equations. *Comput Methods Appl Mech Eng*. 2004;193(15-16):1323-1366.
77. Ku HC, Hirsh RS, Taylor TD. A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations. *J Comput Phys*. 1987;70(2):439-462.
78. Hua J, Lou J. Numerical simulation of bubble rising in viscous liquid. *J Comput Phys*. 2007;222(2):769-795.
79. Amaya-Bower L, Lee T. Single bubble rising dynamics for moderate Reynolds number using lattice Boltzmann method. *J Comput Phys*. 2010;39(7):1191-1207.
80. Hua J, Stene JF, Lin P. Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method. *J Comput Phys*. 2008;227(6):3358-3382.
81. Tryggvason G. Numerical simulation of the Rayleigh-Taylor instability. *J Comput Phys*. 1988;75(2):253-282.
82. Fraigneau Y, Guermond J-L, Quartapelle L. Approximation of variable density incompressible flows by means of finite elements and finite volumes. *Commun Numer Methods Eng*. 2001;17(12):893-902.
83. Calgaro C, Creusé E, Goudon T. An hybrid finite volume-finite element method for variable density incompressible flows. *J Comput Phys*. April 2008;227(9):4671-4696.
84. Lee HG, Kim K, Kim J. On the long time simulation of the Rayleigh-Taylor instability. *Int J Numer Methods Eng*. 2010;85(13):1633-1647.
85. Li XL, Jin BX, Glimm J. Numerical study for the three-dimensional Rayleigh-Taylor instability through the TVD/AC scheme and parallel computation. *J Comput Phys*. 1996;126(2):343-355.

86. Li Q, Fu S, Xu K. A compressible Navier-Stokes flow solver with scalar transport. *J Comput Phys*. 2005;204(2):692-714.
87. Cruchaga MA, Celentano DJ, Tezduyar TE. Collapse of a liquid column: numerical simulation and experimental validation. *Comput Mech*. 2007;39(4):453-476.
88. Mohammadi B, Pironneau O. *Analysis of the K-Epsilon Turbulence Model*. Hoboken, NJ: Wiley; 1994.
89. Bui C. Simulation Des Écoulements Bifluides, Une Stratégie de Couplage Basée Sur l'Adaptation de Maillage Anisotrope [PhD Dissertation]. Paris, France: Universite Pierre et Marie Curie; 2009.
90. Green G. *Mathematical Papers*. Ferrers NM, ed. New York, NY: Chelsea Publishing Co; 1970.

How to cite this article: Frey P, Kazerani D, Ta TTM. An adaptive numerical scheme for solving incompressible 2-phase and free-surface flows. *Int J Numer Meth Fluids*. 2018;87:543–582. https://doi.org/10.1002/fld.4502

APPENDIX A: A FOURTH-ORDER RUNGE-KUTTA SCHEME TO SOLVE THE EQUATION OF CHARACTERISTICS

We introduce a small subintegration time step $\delta t << \Delta t$ and subdivide the interval $]t^{n-1}, t^n[= \cup_{l=0}^{L}]t^l, t^{l+1}[$ with $t^l = t^{n-1} + l\delta t$. A fourth-order Runge-Kutta scheme yields, for all $\mathbf{x} \in \Omega$, to the following approximation $\tilde{\mathbf{X}}(\mathbf{x}, t^n, \cdot)$ of $\mathbf{X}(\mathbf{x}, t^n, \cdot)$ (solution of (25)), defined on $\{t^l\}_{l\in\{0,..,L\}}$ computed by

$$\begin{cases} \tilde{\mathbf{X}}(\mathbf{x}, t^n; t^n) = \mathbf{x} \\ \tilde{\mathbf{X}}(\mathbf{x}, t^n; t^l) = \tilde{\mathbf{X}}(\mathbf{x}, t^n; t^{l+1}) - \frac{\delta t}{6}(v_1 + 2v_2 + 2v_3 + v_4), \\ \text{with} \quad v_1 = \tilde{\mathbf{u}}(\tilde{\mathbf{X}}(\mathbf{x}, t^n; t^l)) \\ \qquad v_2 = \tilde{\mathbf{u}}(\tilde{\mathbf{X}}(\mathbf{x}, t^n; t^{l+1}) - \frac{\delta t}{2} v_1) \\ \qquad v_3 = \tilde{\mathbf{u}}(\tilde{\mathbf{X}}(\mathbf{x}, t^n; t^{l+1}) - \frac{\delta t}{2} v_2) \\ \qquad v_4 = \tilde{\mathbf{u}}(\tilde{\mathbf{X}}(\mathbf{x}, t^n; t^{l+1}) - \delta t v_3). \end{cases}$$

We are now able to use (24) to approximate ϕ^n for all integer $n \in [1, \frac{T}{\Delta t}]$. This problem can be solved by a Galerkin numerical scheme that involves the resolution of a linear system with using quadrature formulas for approximating integrals. Another alternative is the Lagrange interpolation. In fact, both methods have been used in previous works. The first approach in the work of Bui et al[39] is more expensive than the second one in another work of Bui et al.[67] Moreover, the second approach leads to an estimate that enables us to control the geometric error on the interface by the interpolation error. This is very interesting for the mesh adaptation step. Therefore, this second approach is considered in the present work. Indeed, we endow here the computational domain Ω with a mesh $\mathcal{T}_h$, and we consider a suitable Lagrange finite element space V (eg, $\mathbb{P}^1$ or $\mathbb{P}^2$). Then, we denote by ϕ_h^0 the V projection of the initial level-set function ϕ_0. Moreover, at each degree of freedom represented by $\mathbf{x}$, we compute the approximation ϕ_h^n of ϕ^n as follows:

$$\phi_h^n(\mathbf{x}) = \phi_h^{n-1}(\tilde{\mathbf{X}}(\mathbf{x}, t^{n+1}; t^n)). \tag{A1}$$

As mentioned in the work of Bui et al,[67] this method requires the resolution of 1 ODE at each degree of freedom of $\mathcal{T}_h$ and is efficient since no matrix inversion nor quadrature formulas for approximating integrals are considered. As proved in the work of Bui et al,[67] this approximation is sufficient since the advection equation contains no diffusive term. Moreover, due to a mesh adaptation step, the mesh elements are very small in the vicinity of the interface. We also refer the reader to previous works in the work of Bui,[89] for more details on these schemes.

APPENDIX B: PROOFS OF THE PROPOSITION IN SUBSECTION 3.3.2

We first take the scalar product of a test function $\mathbf{v} \in V_n$ with the first equation of (32) and integrate on Ω_{t^n} to get

$$\int_{\Omega(t^n)} \frac{\rho}{\Delta t}\mathbf{u}^n \cdot \mathbf{v} - \int_{\Omega(t^n)} \mu \Delta \mathbf{u}^n \cdot \mathbf{v} + \int_{\Omega(t^n)} \nabla p^n \cdot \mathbf{v} = \int_{\Omega(t^n)} \rho \left(\mathbf{f} + \frac{\mathbf{u}_\star^{n-1}}{\Delta t} \right) \cdot \mathbf{v}, \tag{B1}$$

where $\mathbf{u}_\star^{n-1}$ is set equal to $\mathbf{u}^{n-1}\circ\mathbf{X}^{n-1}(\mathbf{x})$. Then, we multiply the second equation of (32) with a test function $q\in Q_n$ and integrate on $\Omega(t^n)$ to get

$$\int_{\Omega(t^n)} \mathrm{div}\mathbf{u}^n \quad q = 0. \tag{B2}$$

The following lemma is the next step to the variational formulation.

Lemma 1. *Let us consider the solution $(\boldsymbol{u}^n, p^n)\in V_n\times Q_n$ of system (32) with boundary conditions (33). Then, we have for all $\boldsymbol{v}\in V_n$*

$$\int_{\Omega(t^n)} -\mu\Delta\boldsymbol{u}^n\cdot\boldsymbol{v}+\nabla p^n\cdot\boldsymbol{v} = \int_{\Omega(t^n)}\mu\left(\nabla\boldsymbol{u}^n+{}^t\nabla\boldsymbol{u}^n\right):\nabla\boldsymbol{v}-\int_{\Omega(t^n)}p^n div\boldsymbol{v}+\int_{\Gamma(t^n)}\gamma\kappa\mathbf{n}_1\cdot\boldsymbol{v}+\alpha\int_{\partial\Omega(t^n)}[\boldsymbol{u}^n]_{\tan}\cdot[\boldsymbol{v}]_{\tan}.$$

Proof. The first step to prove the lemma is the following equality, which holds true under the incompressibility condition $\mathrm{div}\mathbf{u}^n=0$,

$$\Delta\mathbf{u}^n = \mathrm{div}\left(\nabla\mathbf{u}^n+{}^t\nabla\mathbf{u}^n\right). \tag{B3}$$

Next, we use Green's formula[90] to remark that

$$-\mu\int_{\Omega(t^n)}\Delta\mathbf{u}^n\cdot\mathbf{v} = \mu\int_{\Omega(t^n)}\left(\nabla\mathbf{u}^n+{}^t\nabla\mathbf{u}^n\right):\nabla\mathbf{v}-\mu\int_{\partial\Omega(t^n)}\left(\nabla\mathbf{u}^n+{}^t\nabla\mathbf{u}^n\right)\mathbf{n}_{\partial\Omega(t^n)}\cdot\mathbf{v}-\mu\int_{\Gamma(t^n)}\left[\left(\nabla\mathbf{u}^n+{}^t\nabla\mathbf{u}^n\right)\mathbf{n}_1\right]_{\Gamma(t^n)}, \tag{B4}$$

where the symbol : denotes the usual square matrix product, ie, $A:B=\Sigma_{i,j=1}^d A_{ij}B_{ij}$ for all d-square matrices A and B. Then, we have by the free surface condition on $\Gamma(t^n)$ (the last equation of (33)):

$$\mu\int_{\Gamma(t^n)}\left[\left(\nabla\mathbf{u}^n+{}^t\nabla\mathbf{u}^n\right)\mathbf{n}_1\right]_{\Gamma(t^n)}\cdot\mathbf{v} = \int_{\Gamma(t^n)}\left(\left[p^n\right]_{\Gamma(t^n)}-\gamma\kappa\right)\mathbf{n}_1\cdot\mathbf{v}. \tag{B5}$$

On the other hand, we have

$$\mu\int_{\partial\Omega(t^n)}\left(\nabla\mathbf{u}^n+{}^t\nabla\mathbf{u}^n\right)\mathbf{n}\cdot\mathbf{v} = \mu\int_{\partial\Omega(t^n)}\left(\left(\nabla\mathbf{u}^n+{}^t\nabla\mathbf{u}^n\right)\mathbf{n}\cdot\mathbf{n}\right)(\mathbf{v}\cdot\mathbf{n})+\mu\int_{\partial\Omega(t^n)}\left[\left(\nabla\mathbf{u}^n+{}^t\nabla\mathbf{u}^n\right)\mathbf{n}\right]_{\tan}[\mathbf{v}]_{\tan}. \tag{B6}$$

The first integral on the right-hand side of (B6) vanishes since $v\in V_n$. Then, the slip boundary conditions on $\partial\Omega(t^n)$ (the third and fourth equations of (33)) lead us to

$$\mu\int_{\partial\Omega(t^n)}\left[\left(\nabla\mathbf{u}^n+{}^t\nabla\mathbf{u}^n\right)\mathbf{n}\right]_{\tan}[\mathbf{v}]_{\tan} = -\alpha\int_{\partial\Omega(t^n)}[\mathbf{u}]_{\tan}[\mathbf{v}]_{\tan}. \tag{B7}$$

Hence, (B6) becomes

$$\mu\int_{\partial\Omega(t^n)}\left(\nabla\mathbf{u}^n+{}^t\nabla\mathbf{u}^n\right)\mathbf{n}\cdot\mathbf{v} = -\alpha\int_{\partial\Omega(t^n)}[\mathbf{u}]_{\tan}[\mathbf{v}]_{\tan}. \tag{B8}$$

Gathering (B4), (B7), and (B8), we find

$$-\mu\int_{\Omega(t^n)}\Delta\mathbf{u}^n\cdot\mathbf{v} = \mu\int_{\Omega(t^n)}\left(\nabla\mathbf{u}^n+{}^t\nabla\mathbf{u}^n\right):\nabla\mathbf{v}+\alpha\int_{\partial\Omega(t^n)}[\mathbf{u}]_{\tan}[\mathbf{v}]_{\tan}-\int_{\Gamma(t^n)}\left(\left[p^n\right]_{\Gamma(t^n)}-\gamma\kappa\right)\mathbf{n}\cdot\mathbf{v}. \tag{B9}$$

We now apply Green's formula to $\int_{\Omega(t^n)}\nabla p^n\cdot\mathbf{v}$ and get

$$\int_{\Omega(t^n)}\nabla p^n\cdot\mathbf{v} = -\int_{\Omega(t^n)}p^n\mathrm{div}\mathbf{v}+\int_{\partial\Omega(t^n)}p^n\mathbf{n}\cdot\mathbf{v}+\int_{\Gamma(t^n)}\left[p^n\right]_{\Gamma(t^n)}\mathbf{n}\cdot\mathbf{v}. \tag{B10}$$

Since the test function $\mathbf{v}$ belongs to V_n, we get

$$\int_{\Omega(t^n)}\nabla p^n\cdot\mathbf{v} = -\int_{\Omega(t^n)}p^n\mathrm{div}\mathbf{v}+\int_{\Gamma(t^n)}\left[p^n\right]_{\Gamma(t^n)}\mathbf{n}\cdot\mathbf{v}. \tag{B11}$$

Adding both sides of equalities (B9) and (B11) yields the result. □

Now, we use the lemma to rewrite (B1) and (B2) as follows:

$$\int_{\Omega(t^n)} \frac{\rho}{\Delta t}\mathbf{u}^n \cdot \mathbf{v} + \int_{\Omega(t^n)} \mu \left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) : \nabla \mathbf{v} + \alpha \int_{\partial\Omega(t^n)} [\mathbf{u}^n]_{\tan} \cdot [\mathbf{v}]_{\tan} - \int_{\Omega(t^n)} p^n \mathrm{div}\mathbf{v}$$
$$= \int_{\Omega(t^n)} \rho \left(\mathbf{f} + \frac{\mathbf{u}_\star^{n-1}}{\Delta t}\right) \cdot \mathbf{v} - \int_{\Gamma(t^n)} \gamma\kappa \mathbf{n}_1 \cdot \mathbf{v}, \tag{B12}$$

$$\int_{\Omega(t^n)} \mathrm{div}\mathbf{u}^n q = 0. \tag{B13}$$

Let us also remark that

$$\left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) : \nabla \mathbf{v} = \left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) : {}^t\nabla \mathbf{v}.$$

Therefore, we have

$$\left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) : \nabla \mathbf{v} = 2\mathbf{D}(\mathbf{u}^\mathbf{n}) : \mathbf{D}(\mathbf{v}).$$