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

Pascal Frey, Dena Kazerani, Thi Thanh Mai Ta

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

P. Frey, D. Kazerani, T. T. M. Ta*

Sorbonne Universités, UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France
CNRS, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France

SUMMARY

In this paper, we present a numerical scheme for solving two-phase or free surface flows. Here, the interface/free surface is modelled using the level-set formulation. Besides, the mesh is anisotropic and adapted at each iteration. 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 on 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 which 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. Finally, some numerical results are presented for both bi-fluid and free-surface problems.

KEY WORDS: Two-phase flows; free surface flows; incompressible Navier–Stokes equations; level set method; method of characteristics; finite element method; anisotropic mesh adaptation.

1. INTRODUCTION

Over the last decades, tremendous progress has been achieved on the development and the analysis of numerical methods for one-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 one fluid is already a good starting point for dealing with two fluids (or two-phase flow) as well as free-surface flow problems. Actually, research on these topics has started in the last few years. However, there are several specific issues relevant to two-fluid and free-surface flow problems that are not present in one-phase or one-fluid incompressible flow problems. In this context, the numerical treatment of the interface between two immiscible fluids ad well as the free surface, is certainly a difficult challenge that raises many related and 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 [GR11] on two-phase incompressible flows and [AMW98] or [T+01] for an overview of numerical methods for the simulation of multiphase

*Correspondence to: Dena KAZERANI, email: kazerani@ljll.math.upmc.fr; Pascal FREY, email: frey@ann.jussieu.fr

flows. Let us also mention that in most applications especially in oceanography, free-surface flows are modeled by a shallow water approximation of incompressible Navier–Stokes or free surface Euler equations (see for instance [BdSV71, GN76, GP01, Lan13]). These models are simpler to analyze. Nevertheless, they are approximative and are valid only in the shallow water regime. We try in this work to approach the free-surface problem by solving the free-surface Navier–Stokes equation by adapting the algorithm we propose for two fluids problem. In other words, no shallow water model is considered here. Nevertheless, as done in some test cases, the comparison between numerical simulation of free-surface Navier–Stokes equation and analytical solution of shallow water equations is an interesting issue.

Well-posedness results for the general weak formulation of the Navier–Stokes problem for two-phase flows including the interface condition $V_\Gamma = \mathbf{u}\cdot\mathbf{n}$ (see Section 2) have been analyzed only for special cases, e.g. when the two 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 [Den94] or when it is close to a half-plane [PS10], or under suitable conditions on the viscosity [Des97]. The case of a bounded domain Ω for arbitrary time intervals $[0,T]$, $T>0$ is treated in [Tan93]; 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 [All85, All87, TT95, Tan96, XZZ13]. Most analysis 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 two-fluids flows.

1.1. Interface/free-surface representation

In this section, we briefly recall the two most important methods for representing the interface (or the free surface) $\Gamma(t)$. Broadly speaking, these methods can be classified as Lagrangian ODE techniques and Eulerian PDE 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, the *level set* representation.

1.1.1. Interface tracking. When the interface regularity is sufficient, then normal, curvature and immiscibility condition are well-defined. Given a velocity field $\mathbf{u}\in V$, where V 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 ordinary differential equations:

$$\begin{cases} \dfrac{d\mathbf{X}(\mathbf{x_0},t_0;t)}{dt} = \mathbf{u}(\mathbf{X}(\mathbf{x_0},t_0;t),t) \qquad t\geq 0 \\ \mathbf{X}(\mathbf{x_0},t_0;t_0) = \mathbf{x_0}\,. \end{cases} \tag{1}$$

For $\mathbf{u}(\mathbf{x},t)$ 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 $T>t_0$ sufficiently close to t_0, there is a one to one mapping between $\Omega\times[t_0,T]$ and Ω in the sens 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 equation:

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

which represents a transformation from Eulerian to Lagrangian coordinates. The Navier-Stokes problem can be transformed accordingly into a non stationary problem with a stationary interface $\Gamma(0)$ [Tan93]. 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 a $\mathbf{x_0} \in \Gamma(0)$. This class of method is called *interface tracking*.

Practically, the Navier-Stokes problem is solved on a fixed grid or an unstructured mesh using an 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 mark 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 to transfer 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 two time steps. More details about this method and its implementation can be found in the references [UT92, ET98, ET99, HAC97].

To partially overcome these problems, hybrid approaches like the arbitrary Lagrangian-Eulerian (ALE) method have been proposed and revealed especially efficient for solving fluid-structure interaction [E.n01, Beh01, GT09, GT94]. They consist in solving the interface using a grid or a mesh and then the later 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 methods (VOF), the treatment of the interface is based on a weak formulation of the 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 one of the two fluids if the system is bi-fluid and the fluid domain if the problem is free-surface. This domain is delimited by the border of the support of this characteristic function i.e. $\partial\Omega_1(t) = \partial \text{supp}(\chi_1(\cdot, t))$. The function χ_1 is discontinuous across the interface and the transport equation requires a specific treatment. Given an arbitrary small elementary volume of fluid W and integrating leads to the following equation:

$$\frac{\partial \chi_1}{\partial t} \int_W \chi_1 \, dx + \int_{\partial W} \chi_1 \mathbf{u} \cdot \mathbf{n} \, ds = 0 \,, \tag{4}$$

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

This VOF approach is widely used for the numerical simulation of two-fluid and free-surface flows, mainly because it enjoys good mass (volume) conservation property and can handle topology changes without difficulty [HN81, SZ99]. The main drawbacks are twofold: VOF methods are tedious to implement on unstructured meshes and tend to lose accuracy. In addition, a CFL condition must be satisfied that 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 [GTBD06, TFC+01, dSMN+04].

1.1.2. Interface capturing. As pointed out, in Volume Of Fluid methods, the discontinuous characteristic function χ_1 across the interface imposes specific numerical treatment of the transport equation (3). An interesting alternative consists in introducing a continuous auxiliary function. The level set method, introduced by [DT80, OS88] suggest to use the signed distance function to the initial interface as auxiliary function ϕ:

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

Assuming 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:

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

and when differentiate this equation with respect to t it becomes:

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

This equation (similar to (3)) is well-defined in its current formulation given 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:

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

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 [CIL92].

The level set (6) is not only used for the mathematical analysis of well-posedness of two-fluid flows but has also 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 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 two to three dimensions can be achieved easily. Like the ALE method described above, variants of the level set method can be considered, in which the interface is explicitely discretized (hence the terminology *interface capturing*) by a mesh and this mesh is moved with the flow velocity. As will be seen, intrinsic properties of the interface can be accurately computed using the level set function [Set99].

1.2. Proposed approach

In the present paper, we describe in details our work and propose a general strategy for solving two-phase/free-surface flows which takes advantage of the flexibility of the level set method for capturing and tracking 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 two fluids is obviously not new. The numerical resolution of incompressible Navier–Stokes equations using a Lagrange–Galerkin scheme was first introduced by [BIKL80] and studied in [Pir82]. Nevertheless, unlike similar approaches that solve the flow problem and the transport equation on Cartesian grids [ALTP98, GTBD06, EFFM02, MG07], our approach relies on an adaptive unstructured mesh to carry out these computations. This allows us to solve the time-discretized Navier–Stokes equation on the fluid domain only, in the case of free surface problems.

In [BFM10], authors proposed a former version of this method in two dimensions that required much more computational effort to resolve the problem at each time step and proved to be difficult to get extended in three dimensions. Our method has several things in common with this work: a computational domain Ω is defined and discretized with an unstructured mesh which 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 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 for the extension from the two-dimensional case to the three-dimensional case. This is a tremendous feature insofar as mesh adaptation is concerned, since the mesh generation is known to be more difficult to deal with in three dimensions. Most of the difficulty related to meshing is recast

as a robust *remeshing* problem.

The paper is organized in five sections. The next section briefly presents the mathematical model and the initial, boundary and interfacial conditions. Section 3 is the central section of the paper and presents numerical tools used for the global algorithm. Especially, two variants of the method of characteristics used to resolve the material derivative of Navier-Stokes problem, corresponding to the non linear 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.

2. A MODEL FOR TWO-PHASE/FREE SURFACE FLOWS PROBLEM

In this paper, we are interested in two-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 two different immiscible incompressible phases (i.e. 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 two open subdomains $\Omega_1(t)$ and $\Omega_2(t)$. Each subdomain contains at most one phase such that $\overline{\Omega_1(t)} \cup \overline{\Omega_2(t)} = \overline{\Omega}$, $\Omega_1(t) \cap \Omega_2(t) = \emptyset$. The contact region between the two subdomains is called the *interface* and is denoted by $\Gamma(t) = \overline{\Omega_1(t)} \cap \overline{\Omega_2(t)}$ (cf. Fig 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.

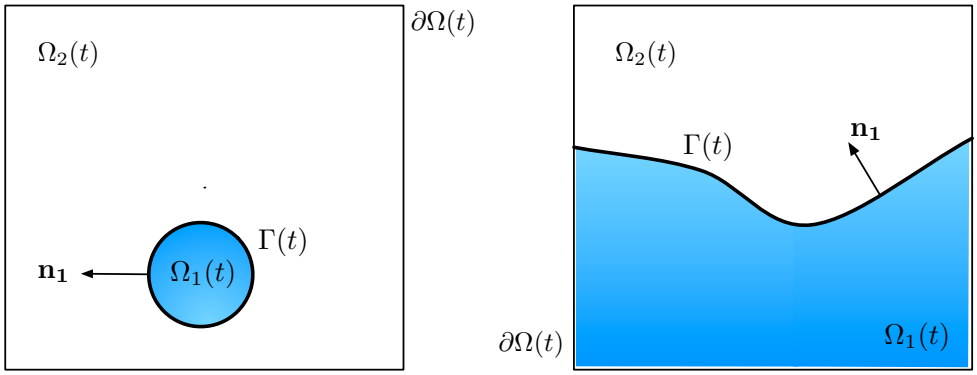


Figure 1. *Illustration of a computational domain $\Omega(t)$ composed of two subdomains $\Omega_1(t)$ and $\Omega_2(t)$. These subdomains are separated by an interface $\Gamma(t)$ which may evolve in time.*

2.1. A Navier-Stokes model for two-phase flows

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

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

where $\sigma_i = -p\mathbf{I} + 2\mu_i \mathbf{D}(\mathbf{u})$ is the stress tensor and $\mathbf{D}(\mathbf{u}) = (\nabla \mathbf{u} + \nabla \mathbf{u}^t)/2$ is the deformation tensor, with $p(x, t)$ the pressure, $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ the velocity and $\mathbf{f}$ is the density of an external volume

force exerted on the phases e.g. 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 the 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 come of two types: *natural* and *essential* boundary conditions. We assume that the outer boundary $\partial\Omega(t)$ remains unchanged in time, hence omitting the time dependency, and is partitioned into three possibly non connected 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 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 inflow velocity profile and conditions along the outer boundaries, e.g. 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

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

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 two conditions:

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

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 $\int_{\partial\Omega_S}\alpha(\mathbf{u}\cdot\tau_{\partial\Omega_S})\cdot(\mathbf{v}\cdot\tau_{\partial\Omega_S})\,ds$ in the left-hand side (see Section 3). Slip without friction corresponds to the case $\alpha = 0$ which makes then 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 i.e. by adding the term $\int_{\partial\Omega_S}\mathcal{A}^*(\mathbf{u}.\mathbf{n}).(\mathbf{v}.\mathbf{n})ds$ in the left-hand side of the weak form where $\mathcal{A}^*$ is a penalization coefficient (large about 10^6).

Next, the 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:

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

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, i.e. $[\cdot]_{\Gamma(t)} = \cdot|_{\Omega_1} - \cdot|_{\Omega_2}$. 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 towards Ω_1 and negative otherwise. The parameter $\gamma > 0$ is the surface tension coefficient (*i.e.* the magnitude of the surface tension force), which is a material property of the model and is assumed to be constant here for sake of simplicity.

Finally, the immiscibility assumption leads to introduce 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 $\Gamma(t)$:

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

Gathering the Navier-Stokes equations (8), the initial and the boundary conditions (9), (10), (11), (12) and the velocity condition on the interface (13) leads to the following model for two-phase

flows:

$$\left\{\begin{array}{rr} \rho_i\left(\dfrac{\partial \mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{u}\right)=\mathrm{div}\sigma_i+\rho_i\mathbf{f} \quad \text{in } \Omega_i(t)\times[0,T]\,(i=1,2) \\ \mathrm{div}\,\mathbf{u}=0 \quad \text{in } \Omega_i\times[0,T]\,(i=1,2) \\ \mathbf{u}(\mathbf{x},t)=\mathbf{u}_D(\mathbf{x},t) \quad \text{on } \partial\Omega_D \\ \sigma\mathbf{n}_{\partial\Omega_N}=-p_e\mathbf{n}_{\partial\Omega_N} \quad \text{on } \partial\Omega_N \\ \mathbf{u}\cdot\mathbf{n}_{\partial\Omega_S}=0 \quad \text{and} \quad \alpha\,\mathbf{u}\cdot\tau_{\partial\Omega_S}+\tau_{\partial\Omega_S}\cdot\sigma\mathbf{n}_{\partial\Omega_S}=0 \quad \text{on } \partial\Omega_S \\ \mathbf{u}(\mathbf{x},0)=\mathbf{u}_0(\mathbf{x}) \quad \text{in } \Omega_i\,(i=1,2) \\ [\mathbf{u}]_{\Gamma(t)}=0\,, \qquad [\sigma]_{\Gamma(t)}\cdot\mathbf{n}_1=-\gamma\,\kappa\mathbf{n}_1 \quad \text{on } \Gamma(t) \\ \mathbf{u}_{\Gamma(t)}=\mathbf{u}\cdot\mathbf{n}_1 \quad \text{on } \Gamma(t) \end{array}\right. \tag{14}$$

Let us mention that by introducing two 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 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 the system (8) with boundary conditions writes

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

Since, no fluid is considered in part $\Omega_2(t)$, the continuity condition on the interface disappear. 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 cinematic boundary condition is written

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

Similarly to the bi-fluid 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 [C+96, S+98, Set99]. 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 $\Gamma(t)$, *i.e.* $\phi(\mathbf{x},t) = d_{\Gamma(t)}(\mathbf{x}, \Gamma(t))$ 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. Especially its gradient is of unit norm wherever it is defined, *i.e.* $|\nabla d_{\Gamma(t)}| = 1$ 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* [Set99, AA02]:

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

with ψ_0 any level set function associated with Ω. A numerical scheme to compute the solution of this problem is devised in [DF12]. This scheme relies on an explicit formula [AA02] and that has been implemented as an iterative algorithm on unstructured meshes.

2.4. *Level set formulation of two-phase/free-surface flows*

The evolution of the implicitly defined interface $\Gamma(t)$ is numerically carried out using the level set method [OS88]. Suppose that the motion of $\Gamma(t)$ over a time period $[0,T]$ is driven by a velocity field $\tilde{\mathbf{u}} : \mathbb{R}^d \to \mathbb{R}^d$ and follows the advection equation:

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

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 cinematic 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:

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

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 algorithm in a way such 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 [DF12].

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 numerical techniques necessary to understand the global scheme of Section 4. Some of these numerical tools let us advect the interface/free surface: Characteristic methods for the resolution of the transport equation, velocity regularization and redistancing whereas some others let us solve the times discretized equation on the fluid domain: 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 [CR73, BB73, LR76, BIKL80] is known to be very efficient for solving advection-diffusion problems, including the Navier-Stokes equations, see [Pir82, Pir89, Pir10] for the mathematical and numerical method and its application to fluid mechanics problems.

Here, we consider the Cauchy problem for the advection equation (19): *given a velocity field* $\tilde{\mathbf{u}}(\mathbf{x},t)$ *defined on* Ω*, find a scalar solution* $\phi(\mathbf{x},t)$ *defined on* $\Omega\times[0,T]$ *solving the equation*:

$$\begin{cases} \dfrac{\partial\phi}{\partial t}(\mathbf{x},t)+\tilde{\mathbf{u}}(\mathbf{x},t)\cdot\nabla\phi(\mathbf{x},t)=0 & \text{on } \Omega\times[0,T] \\ \phi(\mathbf{x},0)=\phi_0(\mathbf{x}) & \text{for } \mathbf{x}\in\Omega. \end{cases} \tag{21}$$

Under proper hypotheses on the regularity and the growth of $\tilde{\mathbf{u}}$ and ϕ_0 [Pir89], the unique C^1 solution ϕ of (21) is

$$\phi(\mathbf{x},t)=\phi_0(\mathbf{X}(\mathbf{x},t;0)) \tag{22}$$

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

$$\begin{cases} \dfrac{d\mathbf{X}(\mathbf{x},t;s)}{ds}=\tilde{\mathbf{u}}(\mathbf{X}(\mathbf{x},t;s),t) & \text{for } s\in\mathbb{R} \\ \mathbf{X}(\mathbf{x},t;t)=\mathbf{x}. \end{cases} \tag{23}$$

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 (21) implies that $\phi(\mathbf{x},t)$ is constant along the characteristics $\mathbf{X}(\mathbf{x},t;s)$. Hence, the solution of the Cauchy problem (21) reads:

$$\phi(\mathbf{x},t)=\phi_0(\mathbf{X}(\mathbf{x},t;0),0)\,,\qquad \text{on } \Omega\times[0,T] \tag{24}$$

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

$$\begin{cases} \dfrac{\partial\phi^n}{\partial t}(\mathbf{x},t)+\tilde{\mathbf{u}}(\mathbf{x},t)\cdot\nabla\phi^n(\mathbf{x},t)=0\,, & (\mathbf{x},t)\in\Omega\times(t^{n-1},t^n) \\ \phi(\mathbf{x},t^n)=\phi^{n-1}(\mathbf{x})\,, & \forall\mathbf{x}\in\Omega. \end{cases} \tag{25}$$

Again, we use the characteristic method to solve (25). In other words, the solution ϕ^n of this equation is approximated by

$$\phi^n(\mathbf{x})=\phi^{n-1}(\mathbf{X}(\mathbf{x},t^n;t^{n-1}))\,,\qquad \text{for all } \mathbf{x}\in\Omega \tag{26}$$

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 the time t^n. This function is computed by

$$\begin{cases} \dfrac{d\mathbf{X}(\mathbf{x},t^n;t)}{dt}=\tilde{\mathbf{u}}(\mathbf{X}(\mathbf{x},t^n;t),t) & \text{for all } (\mathbf{x},t)\in\Omega\times(t^{n-1},t^n) \\ \mathbf{X}(\mathbf{x},t^n;t^n)=\mathbf{x} & \text{for all } x\in\Omega. \end{cases} \tag{27}$$

Let us also note that the first equation of (27) is equivalent to

$$\mathbf{X}(\mathbf{x}, t^n; t) = \mathbf{x} - \int_t^{t^n} \tilde{\mathbf{u}}(\mathbf{X}(\mathbf{x}, t^n; t), t)dt. \tag{28}$$

This integral can be approximated by any method used for solving ordinary differential equations. For example, introducing a small sub-integration time step $\delta t << \Delta t$ and subdividing 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 schemes yields for all $\mathbf{x} \in \Omega$ to the following approximation $\tilde{\mathbf{X}}(\mathbf{x}, t^n, \cdot)$ of $\tilde{\mathbf{X}}(\mathbf{x}, t^n, \cdot)$, defined on $\{t^l\}_{l\in\{0,..,L\}}$ computed by:

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

We are now able to use (26) to approximate ϕ^n for all integer $n \in [1, \frac{T}{\Delta t}]$. This problem can be solved by a Galerkin numerical scheme which involves resolution of a linear system with using quadrature formulas for approximating integrals in the right hand side. Another alternative is the Lagrange interpolation. In fact, both methods have been used in previous works. The first approach in [BFM10] is more expensive than the second one in [CDF12]. Moreover, the second approach leads to an estimate which 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 (e.g. $\mathbb{P}^1$ 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 following:

$$\phi_h^n(\mathbf{x}) = \phi_h^{n-1}(\tilde{\mathbf{X}}(\mathbf{x}, t^{n+1}; t^n)) \tag{29}$$

As mentioned in [CDF12], this method requires the resolution of one 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. This first-order space approximation is sufficient here since in our frame work, only values of the level set function in the vicinity of the free surface are of interest. Moreover, thanks to a mesh adaptation step, mesh elements are very small in the vicinity of the interface. We also refer the reader to the previous works in [Bui09], for more details on these schemes .

Remark

It is worth to forecast some difficulties: in the numerical framework, the characteristic curves may go out of the computational domain specially 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, the definition of the velocity field $\tilde{\mathbf{u}}$ that appears in the equation (19). 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 [SS03]. In two-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 two-flows 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 [SSO94, C+96] 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 [Bur03, dG06], 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 form:

$$\begin{cases} -a\Delta\tilde{\mathbf{u}} + \tilde{\mathbf{u}} = 0 & \text{on } \Omega \\ \nabla\tilde{\mathbf{u}} \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega \quad \text{and} \quad \tilde{\mathbf{u}} = \mathbf{u}_{|\Gamma(t)} & \text{on } \Gamma(t) \end{cases} \tag{30}$$

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

$$a \int_{\Omega(t)} \nabla\tilde{\mathbf{u}} \cdot \nabla\mathbf{v} + \int_{\Omega(t)} \tilde{\mathbf{u}}\mathbf{v} = \int_{\Gamma(t)} \mathbf{u}\mathbf{v} \qquad \text{for all } \mathbf{v} \in V\,. \tag{31}$$

The left hand side of equation (32) 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 i.e. (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

$$\begin{cases} -a\Delta\tilde{\mathbf{u}} + \tilde{\mathbf{u}} = 0 & \text{on } \Omega_2(t) \\ \nabla\tilde{\mathbf{u}} \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega \cap \partial\Omega_1 \quad \text{and} \quad \tilde{\mathbf{u}} = \mathbf{u}_{|\Gamma(t)} & \text{on } \Gamma(t) \end{cases} \tag{32}$$

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 0-level set during the advection. Therefore, the level set function does not represent any more the fluid domain during the 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 are 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

We are interested in this part 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 [BIKL80, Pir82, Glo03]. 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 non linear 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 elements 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 [Pir82]). Moreover, this method is already implemented in our framework to solve the advection equation (21). 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 use obviously the same time step Δt for both advection and Navier–Stokes equation. 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 (23) i.e. the operator $\frac{\partial}{\partial t}+\mathbf{u}.\nabla$ may be turned into a total derivative (also called Lagrange derivative) $\frac{d}{dt}$. Therefore, the first equation of (14) and (15) is recast into the following form:

$$\rho\frac{d\mathbf{u}(\mathbf{X}(\mathbf{x},t;s),s)}{ds}\,|_{s=t}\,-\mu\Delta\mathbf{u}+\nabla p=\rho\mathbf{f}. \tag{33}$$

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

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

since

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

This approximation is equivalent, using the latter notation together with the definition (23) of the characteristic curve, to

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

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

$$\begin{cases}\rho\dfrac{\mathbf{u}^n(\mathbf{x})-\mathbf{u}^{n-1}\circ\mathbf{X}^{n-1}(\mathbf{x})}{\Delta t}-\mu\Delta\mathbf{u}^n(\mathbf{x})+\nabla p^n(\mathbf{x}) &= \rho\mathbf{f}^n \quad \text{in } \Omega_i(t^n)\\ \qquad\qquad \text{div}\mathbf{u}^n(\mathbf{x}) &= 0 \quad \text{in } \Omega_i(t^n)\end{cases} \tag{34}$$

or equivalently

$$\begin{cases}\rho\dfrac{\mathbf{u}^n(\mathbf{x})}{\Delta t}-\mu\Delta\mathbf{u}^n(\mathbf{x})+\nabla p^n(\mathbf{x}) &= \rho\mathbf{f}^n+\rho\dfrac{\mathbf{u}^{n-1}\circ\mathbf{X}^{n-1}(\mathbf{x})}{\Delta t} \quad \text{in } \Omega_i(t^n),\\ \qquad\qquad \text{div}\mathbf{u}^n(\mathbf{x}) &= 0 \quad \text{in } \Omega_i(t^n)\end{cases} \tag{35}$$

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 the time t^{n-1}. Here $i\in\{1,2\}$ if the flow is bifluid and $i=1$ is the flow is free-surface. We endow problem (35) with slip boundary and free surface conditions. Indeed, the boundary condition of (14) writes here for

two phase flows as *

$$\begin{cases} \mathbf{u}^n = 0 \quad \text{on} \quad \partial\Omega_D(t^n), \\ \sigma^n \mathbf{n}_{\partial\Omega_N(t^n)} = -p_e \mathbf{n}_{\partial\Omega_N(t^n)} \quad \text{on} \quad \partial\Omega_N(t^n), \\ \mathbf{u}^n \cdot \mathbf{n}_{\partial\Omega_S(t^n)} = 0 \quad \text{on} \quad \partial\Omega_S(t^n), \\ \left[\alpha \mathbf{u}^n + \mu \left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) \mathbf{n}_{\partial\Omega_S(t^n)}\right]_{tan} = 0 \quad \text{on} \quad \partial\Omega_S(t^n), \\ [\sigma]_{\Gamma(t^n)} \mathbf{n}_1 = -\gamma\kappa\mathbf{n}_1 \quad \text{in} \quad \Gamma(t^n). \end{cases} \tag{36}$$

Likewise, the boundary conditions of (15) for free-surface flow writes

$$\begin{cases} \mathbf{u}^n = 0 \quad \text{on} \quad \partial\Omega_D(t^n) \cap \partial\Omega_1(t^n), \\ \sigma^n \mathbf{n}_{\partial\Omega_N(t^n)} = -p_e \mathbf{n}_{\partial\Omega_N(t^n)} \quad \text{on} \quad \partial\Omega_N(t^n) \cap \partial\Omega_1(t^n), \\ \mathbf{u}^n \cdot \mathbf{n}_{\partial\Omega_S(t^n)} = 0 \quad \text{on} \quad \partial\Omega_S(t^n) \cap \partial\Omega_1(t^n), \\ \left[\alpha \mathbf{u}^n + \mu \left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) \mathbf{n}_{\partial\Omega_S(t^n)}\right]_{tan} = 0 \quad \text{on} \quad \partial\Omega_S(t^n) \cap \partial\Omega_1(t^n), \\ \left(\mu \left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) - p^n\right) \mathbf{n}_1 = -\left(\gamma\kappa + p^a\right) \mathbf{n}_1 \quad \text{in} \quad \Gamma(t^n). \end{cases} \tag{37}$$

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 two 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

- 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 by the same way as in Subsection 3.1.
- Similarly to Subsection 3.1, the characteristic curves may cross some elements or go out the computational domain. In this cases we identify the last element 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 bi-fluid problem. As mentioned previously, we intend to solve the time discretized Navier–Stokes equation (35) together with boundary conditions (36) by finite elements method for all integer $n \in [0, \frac{T}{\Delta t}]$. Equation (35) is nothing but a generalized Stokes equation. The associated variational formulation is then obtained by the same method as for generalized stokes problems. We assume in this part for sake of simplicity that the fluid satisfies everywhere on $\partial\Omega$, the Navier slip boundary conditions i.e. $\partial\Omega = \partial\Omega_S$ 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, Hilbert spaces we consider here for the velocity flow and the pressure respectively V_n and Q_n defined by (see for instance [Ver85, Ver87, Ver91, DU15]):

$$V_n = \{\mathbf{v} \in \left(\mathbb{H}^1\left(\Omega(t^n)\right)\right)^d, \mathbf{v} \cdot \mathbf{n}_{\partial\Omega(t^n)} = 0 \text{ on } \partial\Omega(t^n)\}, \tag{38}$$

*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 sake of simplicity, we do not use any n superscript for these symbols in (35) either in the following section.

†Nevertheless, if the flow is bi-fluid 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 $Q_n = \mathbf{L}^2_0(\Omega)$.

$$Q_n = \mathbb{L}^2(\Omega(t^n)). \tag{39}$$

We are now able to find the variational formulation of the problem. We first take the scalar product of a test function $\mathbf{v} \in V_n$ with the first equation of (35) and integrate on Ω_{t^n} to get:

$$\int_{\Omega(t^n)} \frac{\rho}{\Delta t}\mathbf{u}^n \cdot \mathbf{v} - \int_{\Omega(t^n)} \mu\Delta\mathbf{u}^n \cdot \mathbf{v} + \int_{\Omega(t^n)} \nabla p^n \cdot \mathbf{v} = \int_{\Omega(t^n)} \rho\left(\mathbf{f} + \frac{\mathbf{u}_\star^{n-1}}{\Delta t}\right) \cdot \mathbf{v}, \tag{40}$$

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 (35) with a test function $q \in Q_n$ and integrate on $\Omega(t^n)$ to get

$$\int_{\Omega(t^n)} div\mathbf{u}^n \; q = 0. \tag{41}$$

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

lemma

Let us consider the solution $(\mathbf{u}^n, p^n) \in V_n \times Q_n$ of system (35) with boundary conditions (36). Then, we have for all $\mathbf{v} \in V_n$

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

Proof

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

$$\Delta\mathbf{u}^n = div\left(\nabla\mathbf{u}^n + {}^t\nabla\mathbf{u}^n\right). \tag{42}$$

Next, we use the Green's formula [Gre70] to remark that

$$-\mu\int_{\Omega(t^n)} \Delta\mathbf{u}^n \cdot \mathbf{v} = \mu\int_{\Omega(t^n)} \left(\nabla\mathbf{u}^n + {}^t\nabla\mathbf{u}^n\right) : \nabla\mathbf{v} - \mu\int_{\partial\Omega(t^n)} \left(\nabla\mathbf{u}^n + {}^t\nabla\mathbf{u}^n\right)\mathbf{n}_{\partial\Omega(t^n)} \cdot \mathbf{v} - \mu\int_{\Gamma(t^n)} \left[\left(\nabla\mathbf{u}^n + {}^t\nabla\mathbf{u}^n\right)\mathbf{n}_1\right]_{\Gamma(t^n)}, \tag{43}$$

where Symbol : denotes the usual square matrix product i.e. $A : B = \Sigma_{i,j=1}^d A_{ij}B_{ij}$ for all d-square matrices A and B. Then we have by the free surface condition on $\Gamma(t^n)$ (the last equation of (36)):

$$\mu\int_{\Gamma(t^n)} \left[\left(\nabla\mathbf{u}^n + {}^t\nabla\mathbf{u}^n\right)\mathbf{n}_1\right]_{\Gamma(t^n)} \cdot \mathbf{v} = \int_{\Gamma(t^n)} \left([p^n]_{\Gamma(t^n)} - \gamma\kappa\right)\mathbf{n}_1 \cdot \mathbf{v}. \tag{44}$$

On the other hand, we have

$$\mu\int_{\partial\Omega(t^n)} \left(\nabla\mathbf{u}^n + {}^t\nabla\mathbf{u}^n\right)\mathbf{n} \cdot \mathbf{v} = \mu\int_{\partial\Omega(t^n)} \left(\left(\nabla\mathbf{u}^n + {}^t\nabla\mathbf{u}^n\right)\mathbf{n} \cdot \mathbf{n}\right)(\mathbf{v} \cdot \mathbf{n}) + \mu\int_{\partial\Omega(t^n)} \left[\left(\nabla\mathbf{u}^n + {}^t\nabla\mathbf{u}^n\right)\mathbf{n}\right]_{tan}[\mathbf{v}]_{tan}. \tag{45}$$

The first integral of the right hand side of (45) vanishes since $v \in V_n$. Then, the slip boundary conditions on $\partial\Omega(t^n)$ (the third and fourth equations of (36)) leads us to

$$\mu\int_{\partial\Omega(t^n)} \left[\left(\nabla\mathbf{u}^n + {}^t\nabla\mathbf{u}^n\right)\mathbf{n}\right]_{tan}[\mathbf{v}]_{tan} = -\alpha\int_{\partial\Omega(t^n)} [\mathbf{u}]_{tan}[\mathbf{v}]_{tan} \tag{46}$$

Hence, (45) becomes

$$\mu \int_{\partial\Omega(t^n)} \left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) \mathbf{n} \cdot \mathbf{v} = -\alpha \int_{\partial\Omega(t^n)} [\mathbf{u}]_{tan} [\mathbf{v}]_{tan} \tag{47}$$

Gathering (43),(46) and (47), we find

$$-\mu \int_{\Omega(t^n)} \Delta \mathbf{u}^n \cdot \mathbf{v} = \mu \int_{\Omega(t^n)} \left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) : \nabla \mathbf{v} + \alpha \int_{\partial\Omega(t^n)} [\mathbf{u}]_{tan} [\mathbf{v}]_{tan} - \int_{\Gamma(t^n)} \left([p^n]_{\Gamma(t^n)} - \gamma\kappa\right) \mathbf{n} \cdot \mathbf{v}. \tag{48}$$

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

$$\int_{\Omega(t^n)} \nabla p^n \cdot \mathbf{v} = -\int_{\Omega(t^n)} p^n \, div \mathbf{v} + \int_{\partial\Omega(t^n)} p^n \mathbf{n} \cdot \mathbf{v} + \int_{\Gamma(t^n)} [p^n]_{\Gamma(t^n)} \mathbf{n} \cdot \mathbf{v}. \tag{49}$$

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

$$\int_{\Omega(t^n)} \nabla p^n \cdot \mathbf{v} = -\int_{\Omega(t^n)} p^n \, div \mathbf{v} + \int_{\Gamma(t^n)} [p^n]_{\Gamma(t^n)} \mathbf{n} \cdot \mathbf{v}. \tag{50}$$

Adding the both sides of equalities (48) and (50) yields the result. □

Now, we use the lemma to rewrite (40)-(41) as following:

$$\int_{\Omega(t^n)} \frac{\rho}{\Delta t} \mathbf{u}^n \cdot \mathbf{v} + \int_{\Omega(t^n)} \mu \left(\nabla \mathbf{u}^n + {}^t\nabla \mathbf{u}^n\right) : \nabla \mathbf{v} + \alpha \int_{\partial\Omega(t^n)} [\mathbf{u}^n]_{tan} \cdot [\mathbf{v}]_{tan} - \int_{\Omega(t^n)} p^n div \mathbf{v} =$$
$$\int_{\Omega(t^n)} \rho \left(\mathbf{f} + \frac{\mathbf{u}_\star^{n-1}}{\Delta t}\right) \cdot \mathbf{v} - \int_{\Gamma(t^n)} \gamma\kappa \mathbf{n}_1 \cdot \mathbf{v}, \tag{51}$$

$$\int_{\Omega(t^n)} div \mathbf{u}^n \, q = 0. \tag{52}$$

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,

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

The variational formulation is then given in the following proposition.

Proposition

The variational formulation corresponding to (35) , (36) is:
Find $(\mathbf{u}^n, p^n)$ in $V_n \times Q_n$ such that we have for all $(\mathbf{v}, q) \in V_n \times Q_n$

$$\begin{cases} a(\mathbf{u}^n, \mathbf{v}) + b(\mathbf{v}, p^n) = l(\mathbf{v}), \\ b(\mathbf{u}^n, q) = 0. \end{cases} \tag{53}$$

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

$$a(\mathbf{u}, \mathbf{v}) = \frac{\rho}{\Delta t} \int_{\Omega(t^n)} \mathbf{u} \cdot \mathbf{v} + 2\mu \int_{\Omega(t^n)} \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) + \alpha \int_{\partial\Omega(t^n)} [\mathbf{u}]_{tan} \cdot [\mathbf{v}]_{tan}. \tag{54}$$

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

$$b(\mathbf{u}, q) = -\int_{\Omega(t^n)} div \mathbf{u} q. \tag{55}$$

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

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

Remark

Let us mention that the operator b satisfies the Babuska-Brezzi inf-sup condition and the variational formulation (53) is well-posed (for instance, see [BF91]).

3.3.3. Variational formulation for free-surface problem. In order to obtain the variational formulation for the time-discretized Navier–Stokes free-surface problem. We consider (35) together with boundary conditions (37). 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 bi-fluid 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

The variational formulation corresponding to (35) , (37) is:

$$V_n = \{\mathbf{v} \in \left(\mathbb{H}^1\left(\Omega_1(t^n)\right)\right)^d, \mathbf{v}\cdot\mathbf{n}_{\partial\Omega_S(t^n)} = 0 \text{ on } \partial\Omega_S(t^n)\}, \tag{56}$$

$$Q_n = \mathbb{L}^2(\Omega_1(t^n)). \tag{57}$$

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

$$\begin{cases} a(\mathbf{u}^n, \mathbf{v}) + b(\mathbf{v}, p^n) = l(\mathbf{v}), \\ b(\mathbf{u}^n, q) = 0. \end{cases} \tag{58}$$

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

$$a(\mathbf{u}, \mathbf{v}) = \frac{\rho}{\Delta t}\int_{\Omega_1(t^n)} \mathbf{u}\cdot\mathbf{v} + 2\mu\int_{\Omega_1(t^n)} \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) + \alpha\int_{\partial\Omega_S(t^n)} [\mathbf{u}]_{tan}\cdot[\mathbf{v}]_{tan}\,. \tag{59}$$

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

$$b(\mathbf{u}, q) = -\int_{\Omega_1(t^n)} div\mathbf{u}q. \tag{60}$$

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

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

3.3.4. Penalization technique for slip boundary condition. Variational formulations (53) and (58) are defined on the Hilbert spaces defined in two 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 two 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 with this difficulty (see [DU15]). In this method, we do not consider any more a strict subspace of $\mathbf{H}^1(\Omega(t^n))^d$ or $\mathbf{H}^1(\Omega_1(t^n))^d$ but these entire spaces. The non-penetration 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 we have for all $(\mathbf{v}, q) \in K_n \times Q_n$

$$\begin{cases} a_\epsilon(\mathbf{u}_\epsilon^n, \mathbf{v}) + b(\mathbf{v}, p_\epsilon^n) = l(\mathbf{v}), \\ b(\mathbf{u}_\epsilon^n, q) = 0. \end{cases} \tag{61}$$

where

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

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 two 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 (61) converges (for the usual norm $K_n \times Q_n$) to the solution $(\mathbf{u}^n, p^n)$ of (53) (or (58) if the flow is free-surface) when ϵ tends to 0 (see [DU15]).

Remark

Simple computations lead to

$$\alpha\int_{\partial\Omega_S(t^n)} [\mathbf{u}]_{tan}\cdot[\mathbf{v}]_{tan} = \alpha\int_{\partial\Omega_S(t^n)} \mathbf{u}\cdot\mathbf{v} - (\mathbf{u}\cdot\mathbf{n})(\mathbf{v}\cdot\mathbf{n}).$$

Therefore, the bilinear operator a_ϵ can also write

$$a_\epsilon(\mathbf{u},\mathbf{v}) = \frac{\rho}{\Delta t}\int_{\Omega(t^n)} \mathbf{u}\cdot\mathbf{v} + 2\mu\int_{\Omega(t^n)} \mathbf{D}(\mathbf{u}):\mathbf{D}(\mathbf{v}) + \alpha\int_{\partial\Omega_S(t^n)} \mathbf{u}\cdot\mathbf{v} + \left(\frac{1}{\epsilon}-\alpha\right)\int_{\partial\Omega_S(t^n)} (\mathbf{u}\cdot\mathbf{n})(\mathbf{v}\cdot\mathbf{n}).$$

or

$$a_\epsilon(\mathbf{u},\mathbf{v}) = \frac{\rho}{\Delta t}\int_{\Omega_1(t^n)} \mathbf{u}\cdot\mathbf{v} + 2\mu\int_{\Omega_1(t^n)} \mathbf{D}(\mathbf{u}):\mathbf{D}(\mathbf{v}) + \alpha\int_{\partial\Omega_S(t^n)} \mathbf{u}\cdot\mathbf{v} + \left(\frac{1}{\epsilon}-\alpha\right)\int_{\partial\Omega_S(t^n)} (\mathbf{u}\cdot\mathbf{n})(\mathbf{v}\cdot\mathbf{n}).$$

if the flow is free-surface.

3.3.5. Resolution by 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 :

$$\begin{cases} \forall \mathbf{v}_h \in K_n^h & a_\epsilon(\mathbf{u}_h,\mathbf{v}_h) + b(\mathbf{v}_h, p_h) = l(\mathbf{v}_h) \\ \forall q_h \in Q_n^h & b(\mathbf{u}_h, q_h) = 0 \end{cases} \tag{62}$$

where $K_n^h \subset K_n$ and $Q_n^h \subset Q_n$ represent two families of finite dimensional subspaces constructed from a triangulation $\mathcal{T}_h$ covering the fluid domain $\Omega(t^n)$ (or $\Omega_1(t^n)$). 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). Symbol Γ_h represents the part of $\mathcal{T}_h$ which corresponds to a discrete approximation of the interface/free surface. Moreover, $\partial_S\mathcal{T}_h$ represents the part of $\partial\mathcal{T}_h$ which corresponds to the discretization of the $\partial_S\Omega(t^n)$ part of the border of the fluid domain.

Besides, $a_\epsilon(\mathbf{u}_h,\mathbf{v}_h)$, $b(\mathbf{v}_h,p_h)$, $l(\mathbf{v}_h)$ 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:

$$\begin{aligned} a_\epsilon(\mathbf{u}_h,\mathbf{v}_h) = &\sum_{K\in\mathcal{T}_h}\frac{\rho}{\Delta t}\int_K \mathbf{u}_h.\mathbf{v}_h dx + \sum_{K\in\mathcal{T}_h} 2\mu\int_K \mathbf{D}(\mathbf{u}_h):\mathbf{D}(\mathbf{v}_h)dx + \alpha\sum_{E\in\partial_S\mathcal{T}_h}\int_E \mathbf{u}_h\cdot\mathbf{v}_h dx \\ &+ \left(\frac{1}{\epsilon}-\alpha\right)\sum_{E\in\partial_S\mathcal{T}_h}\int_E (\mathbf{u}_h\cdot\mathbf{n}(E))\,(\mathbf{v}_h\cdot\mathbf{n}(E))dx; \end{aligned}$$

$$b(\mathbf{v}_h, p_h) = \sum_{K\in\mathcal{T}_h} \int_K -p_h \text{div}\mathbf{v}_h;$$

$$l(\mathbf{v}_h) = \sum_{K\in\mathcal{T}_h} \int_K \rho \mathbf{f}_h \cdot \mathbf{v}_h dx + \frac{\rho}{\Delta t} \sum_{K\in\mathcal{T}_h} \int_K \rho\left(\mathbf{u}_\star^{n-1}\right)_h . \mathbf{v}_h dx + l_{Surface}(\mathbf{v}_h);$$

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 $l_{Surface}(\mathbf{v}_h)$ is a discretization of the surface tension term $-\int_{\Gamma_{t^n}} \gamma\kappa\mathbf{n}_1.\mathbf{v}$ if the flow is two-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 concerned in the next section.
The existence and uniqueness of the weak formulation for generalized Stokes problem has been proven, see [EG04], [Qua09] and [DU15]. This proof dues to: i. the ellipticity of the form $a_\epsilon(.,.)$; ii. the compatibility of the spaces of velocity and pressure results satisfying the Babuska–Brezzi condition, also called *inf-sup condition* on the form $b(.,.)$, i.e. there exists a positive constant C such that:

$$\inf_{q\in Q_n} \sup_{\mathbf{v}\in K_n} \frac{b(\mathbf{v}, p)}{\|\mathbf{v}\|_1 \|q\|_0} \geq C > 0 \tag{64}$$

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 $H^1(\Omega)$, $L^2(\Omega)$ (or $H^1(\Omega_1)$, $L^2(\Omega_1)$ if the problem is free-surface) respectively.
The approximative problem also requires a compatibility condition, meaning that the discrete spaces of velocity needs 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, the problem (62) leads to solve the square linear system:

$$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix} \tag{65}$$

where the matrices A, B correspond to the bilinear forms a_ϵ and b respectively, and F corresponds to the right hand side l of the first equation. The system (65) 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 classical *Uzawa* method [AHU58] as recalled in the following paragraph.

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 following formulas:

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

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 three dimensions). In 2-dimensional 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 $\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}$ represent its three 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 quadrature formula along each edge E of Γ_h, it has been shown in [BFM10]

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

that the surface tension term can be rewritten as follows, for all $\mathbf{v}_h \in K_n^h$:

$$\begin{aligned}\int_{\Gamma_h} \gamma\kappa\mathbf{n}_1.\mathbf{v}_h ds &= \sum_{E\subset\Gamma_h} \frac{|E|}{2} \sum_{\mathbf{x}_i\in E} \gamma\kappa(\mathbf{x}_i)\mathbf{n}_h(\mathbf{x}_i).\mathbf{v}_h(\mathbf{x}_i) \\ &= \sum_{\mathbf{x}_i\in\Gamma_h} \gamma\kappa(\mathbf{x}_i)\mathbf{n}_h(\mathbf{x}_i).\mathbf{v}_h(\mathbf{x}_i) \sum_{E\ni\mathbf{x}_i} \frac{|E|}{2} \end{aligned} \tag{66}$$

where the unit normal vector $\mathbf{n}_h$ to Ω_1 is computed from the approximation of 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 [FG08]:

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

Other formulas can be used to approximate $r(\mathbf{x})$ or $\kappa(\mathbf{x})$, see for instance [T+01].

The atmospheric pressure term can be then added with no difficulty:

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

This technique can be extended straightforwardly to three 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 which minimize the distance to discrete contour of the mesh at this vertex. Therefore, this is done by a solving a polynomial minimization problem (see [Fre00] for more details).

3.4. Redistancing

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

$$|\nabla\phi| = 1. \tag{67}$$

Unfortunately, when ϕ is transported by a physical velocity field using equation (21), all the isolines do not travel with the same speed. As a consequence, the level set does not preserve the property (67). A natural choice to reinitialize the level-set function is the signed distance function to the interface for all time iteration $n \in \mathbf{N}$ in the discrete approximation $\mathcal{T}_h$ of the computational domain Ω:

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

where Ω^1 (resp. Ω^2) represents parts of $\mathcal{T}_h$ corresponding to Ω_1 (resp. Ω_2). In our scheme, this signed distance function is approximated by *redistancing procedure* studied in [DF11], consists in implementation the following two steps:

- Step 1: Initialization ϕ^0 of ϕ: denoting $\mathcal{T}_\Gamma$ the set of mesh elements intersected by the interface, i.e. $\mathcal{T}_\Gamma = \{K \in \mathcal{T}_h : K \cup \Gamma_h \neq \emptyset\}$, $\phi^0(x)$ is defined as:

$$\phi^0(x) = \begin{cases} \phi(x) & \text{if } x \in \mathcal{T}_\Gamma \\ +\infty & \text{if } x \in \Omega^1 \setminus \mathcal{T}_\Gamma \\ -\infty & \text{if } x \in \Omega^2 \setminus \mathcal{T}_\Gamma. \end{cases}$$

- Step 2: Numerical computation of ϕ as *steady solution* of so-called *Eikonal equation*:

$$\begin{cases} \dfrac{\partial\phi}{\partial t}(\mathbf{x},t) + sgn(\phi_0)(|\nabla\phi| - 1) &= 0 \quad \forall(\mathbf{x},t) \in \Omega\times(0,T) \\ \phi(\mathbf{x},0) &= \phi^0(\mathbf{x}) \quad \forall\mathbf{x}\in\Omega. \end{cases} \tag{68}$$

The long term solution of this equation keeps a similar behavior in "vicinity" of the zero isoline i.e. the position of interface $\Gamma(t^n)$ is preserved to ensure the constraint (67). The numerical computation of the solution of this equation is based on the explicit expression of its unique uniformly continuous viscosity solution, see [DF11] for more details.

3.5. Mesh adaptation

The key tool which lets us solve the time-discretized fluid problem on the fluid domain (in the case of free-surface problem), is the anisotropic mesh adaptation of the computational domain Ω based on the location of the free surface. In the case of two phase problems, this adaptation lets us find an approximative solution satisfying given errors. The idea of this method is based on 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, approximation error (see [FG08] and for details). The estimates we consider here, are from [Pir82] for time-discretized Navier–Stokes problem and from [CDF12] 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 [Pir82] for the Navier–Stokes equation with Dirichlet boundary conditions. Indeed, let $\mathbf{u}$ be the smooth solution of the Navier–Stokes equations with Dirichlet boundary condition. While $\mathbf{u}_h$ 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[§]

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2} \leq c(h + \Delta t + h^2/\Delta t), \tag{69}$$

where c is a strictly positive constant and h is the characteristic mesh element size. According to this error estimate, there should be a coherence between the time step and 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 of [CDF12] for the advection equation on the whole space conjecturing that similar estimates holds true on bounded domains. In fact, we see that the approximation error associated with the level set function is bounded by above by the interpolation error of the level set function:

$$\|\phi - \phi_h\|_{L^\infty(\Omega)} \leq \|\phi - \Pi_h\phi\|_{L^\infty(\Omega)} + \|\phi_0 - \Pi_h\phi_0\|_{L^\infty(\Omega)} + c_1\|\tilde{\mathbf{u}} - \Pi\tilde{\mathbf{u}}\|_{L^\infty(\Omega)} + c_2 e^{\delta t}\delta t, \tag{70}$$

where c_1, c_2 are constants depending on initial data ϕ_0 and velocity $\mathbf{u}$ while 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 0-isoline contour Γ_h and the continuous one Γ satisfies

$$d^H(\Gamma_t, \Gamma_n^h) \leq \sup\left(\frac{\sup_{x\in\Omega}|\nabla\phi(x)|}{\inf_{x\in\Omega}|\nabla\phi(x)|^2}, \frac{\sup_{K\in\mathcal{T}_n^h}|\nabla\phi_h(x)|_K|}{\inf_{K\in\mathcal{T}_n^h}|\nabla\phi_h(x)|_K|^2}\right)\|\phi - \phi_h\|_{L^\infty(\Omega)} \tag{71}$$

Therefore, we see that the geometric error is bounded by the level set function approximation error which is itself bounded by the interpolation error of the solution of the problem. Using the result of [FA05] on the $\mathbf{L}^\infty$ error estimate for the Lagrange finite element $\mathbb{P}^1$-interpolation, we have for all

[§]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 [Pir82] on time dependent Navier–Stokes equation with Dirichlet boundary conditions since in our knowledge, no error estimate result is available on free surface Navier–Stokes equations.

function defined on Ω and on all elements K of the mesh $\mathcal{T}_n^h$:

$$
\begin{aligned}
||\varphi - \Pi_h\varphi||_{L^\infty(K)} &\leq c_d \max_{x\in K}\max_{\overrightarrow{v}\subset K} < \overrightarrow{v}, |\nabla^2\varphi(x)|\overrightarrow{v} > \\
&\leq c_d \max_{x\in K}\max_{\overrightarrow{e}\subset E_K} < \overrightarrow{e}, |\nabla^2\varphi|\overrightarrow{e} >
\end{aligned}
\tag{72}
$$

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 [AF03] for details). According to (70) and (71), 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 [FG08]) associated with ϕ, ϕ_0 and $\mathbf{u}$. Then, at each step, the generation of the mesh from this metric is then obtained by using a Delaunay-based local mesh modification procedure (see [DF08]).

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

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 is covered here by a time depending 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 anisotropic metric tensor and always adapted at each time step, hence the mobile unstructured meshes are generated during the scheme.

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 two-phase Then, at each iteration $n = 0, \cdots, N-1$, 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 two 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 (35) (with boundary conditions (36)) is solved on $\mathcal{T}^{n+1}$. This overall algorithm is recapitulated in the following table.

Numerical scheme for bifluid flow over $[0,T]$			
1. Start with mesh $\mathcal{T}^0$ and initialization $\mathbf{u}^0$,			
2. For n = 0, ..., 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 $(\mathbf{u}^N, \phi^N, T^N)$			

In comparison with previous studies in [BFM10], this scheme has been much reduced. In that approach, mesh adaptation needed two 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 two 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 sub-domain $\mathcal{T}_1^n$ 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 is 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 0. This is not coherent with the physics of the system nor with the continuous mathematical model which represents it. Indeed, one of the features of this work is to compare the free surface simulation with the bi-fluid simulation where air represents on of the two phases. In other words, in the case of free-surface flows, the action of the second fluid is describe 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 regularize 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 on the vicinity of free surface and in the fluid domain which are used to construct the right hand side of the linear system (65). As a confirmation, we can check by numerical simulations that the behavior of the free surface is different if we give the non extended velocity to the Navier–Stokes solver. Let us also mention that if we consider a Stokes flow, the placement of the velocity extension 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.

Let us also note that of the flow velocity represents too many discrepancies. A regularization of velocity must be considered before the advection step. The algorithm is recapitulated as following:

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

1. Start with $\mathcal{T}$ and $\mathbf{u}^0$.
2. For $n = 0, ..., N-1$,

Smooth velocities	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	$(\mathcal{T}_1^{n+1}, \tilde{\mathbf{u}}^n)$	$(\mathbf{u}^{n+1}, p^{n+1})$

Non-smooth velocities	Problem	Input	Output
2.1	Velocity regularization	$(\mathcal{T}^n, \mathbf{u}^n)$	$\tilde{\mathbf{u}}^n\|_{adv}$
2.2	Velocity extension	$(\mathcal{T}_2^n, \mathbf{u}^n)$	$\tilde{\mathbf{u}}^n\|_{NS}$
2.3	Redistancing	$\mathcal{T}^n$	ϕ^n
2.4	Solving advection	$(\mathcal{T}^n, \tilde{\mathbf{u}}^n\|_{adv}, \phi^n)$	$\tilde{\phi}^{n+1}$
2.5	Mesh adaptation	$(\mathcal{T}^n, \tilde{\phi}^{n+1})$	$\mathcal{T}^{n+1}$
2.6	Interpolation	$(\mathcal{T}^{n+1}, \tilde{\mathbf{u}}^n\|_{NS})$	$\tilde{\mathbf{u}}^n\|_{NS}$
2.7	Solving Navier–Stokes	$(\mathcal{T}_1^{n+1}, \tilde{\mathbf{u}}^n\|_{NS})$	$(\mathbf{u}^{n+1}, p^{n+1})$

3. We return $\mathbf{u}^N, p^N, \phi^N, D^N$.

Here, $\mathcal{T}_1^n$ (resp. $\mathcal{T}_2^n$) represents the submeshes of the computational mesh $\mathcal{T}^n$ which corresponds to the fluid domain Ω_1 (resp. to the vacuum part Ω_2). Let us recall that during the velocity regularization step, the Helmholtz equation (32) 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, we also need to solve the Helmholtz equation (30) to get an extended velocity to give to the time-discretized Navier–Stokes solver. Let us also mention that the resolution of the problem (30) is not expensive since mesh elements of $\mathcal{T}_2^n$ are quite large, since 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 bi-fluid problems. First, the Navier–Stokes solver for monofluid has been validated by the Lid-driven cavity test (see Subsection 5.1). Next, the test of a rising bubble and a Rayleigh–Taylor instability are investigated (see Subsections 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 Subsection 5.4). Then, dam break examples for shampoo and water are presented (see Subsection 5.5 and 5.6). The results in both two dimensions (2D) and three 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 size of the mesh whereas $hgrad$ represents the graduation scale of 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 three dimension is given by extending the two dimensional domain in 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 $u_x = 1m/s$. The viscosity is adjusted to obtain the

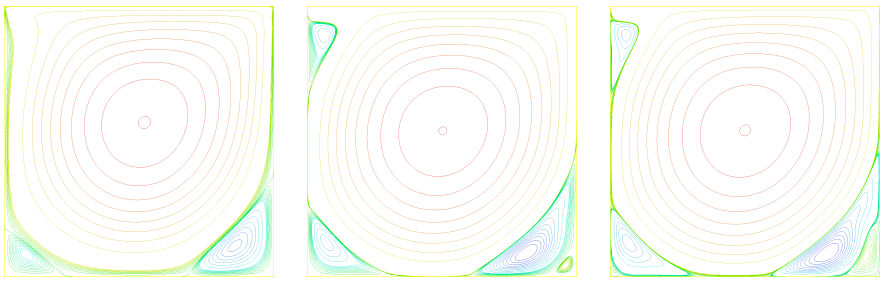
desired Reynolds number.

5.1.1. Two-dimensional lid-driven cavity

In two dimensions we investigated the simulations for Reynolds numbers from 100 up to 10000. Four meshes have been used: a regular triangulation (carre2) with 2461 nodes, 5000 elements; an uniform triangulation (carre3) with 2143 nodes and 4136 elements; another uniform triangulation (carre4) with 8421 nodes,16544 elements; and a regular triangulation (carre7) with 10201 nodes, 20000 elements used for the test with high Reynolds numbers ($Re = 10000$). This problem involves a primary vortex at the cavity center and the vortices in the corners as Re increases. It is known 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 towards the center of cavity. In Table I, we resume the positions of the center of the primary vortices at the steady-state (when the residual between the solutions reaches to 10^{-6}) for $Re = 100, 400, 1000$. The streamlines are presented in Figure 2 for higher Reynolds numbers. We also compute the profiles of velocity along horizontal and vertical lines passing the geometric center of cavity, see Figure 3. Our numerical computations in these cases are compared to the results obtained in the very well-known references [UG82], in [GM99], and the benchmark result for cavity flow in [ECG05] obtained with a fine uniform grid mesh of 601×601.

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

Table I. *Cavity in 2D: comparison between the positions of the main vortex for different Reynolds numbers.*



$Re = 1000$ $Re = 5000$ $Re = 10000$

Figure 2. *Cavity in 2D: Streamlines for different Reynolds number.*

We have also obtained a good agreement between our pressure solution and the result showed in [HRK+10], see Figure 4.

5.1.2. Lid-driven cavity in 3D

Two meshes are employed to simulate the 3D problem: the large one consists in 11037 vertices, 56244 tetrahedra used for the cases of $Re = 100$, $Re = 400$ and the other one consists in 35723 vertices, 193586 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 $z = const$ corresponds to those in 2D, see Figure 5 for example.

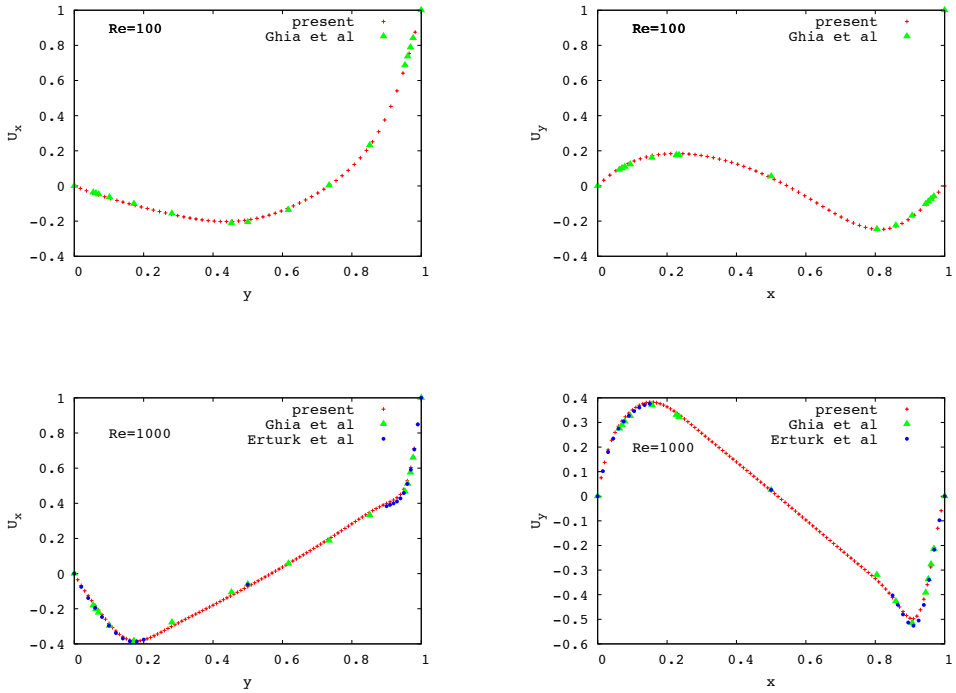


Figure 3. *Cavity in 2D: velocity profile for u_x and u_y in the cases of Re=100 and Re =1000.*

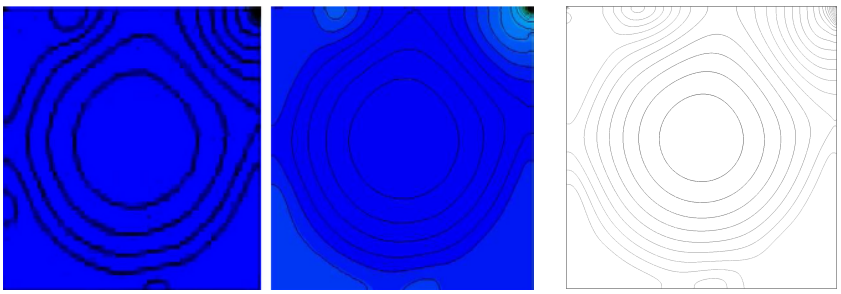


Figure 4. *Cavity in 2D: Isolines of pressure with Re = 10000. Left: result in [GWR04]. Center: result in [HRK+10]. Right: present result*

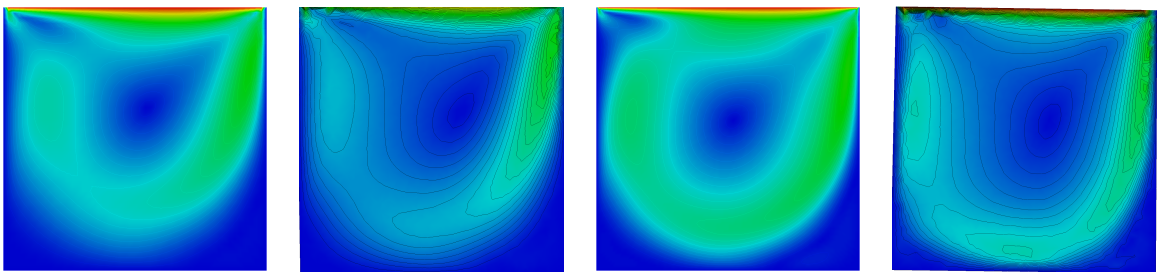


Figure 5. *Cavity in 3D: from left to right, streamlines in 2D and in the plane ($z = 0.5$) of 3D for $Re = 400$ (the first two), $Re = 1000$ (the last two).*

We also compare our numerical solutions with the results in [KHT87] since the Reynolds numbers are exactly the same in our tests. However, no data for velocity profiles are available in [KHT87] . For this reason, the comparison is based on the images (see Figure 6 which shows a good agreement between our velocity profiles in 2D, 3D and those obtained in the given reference).

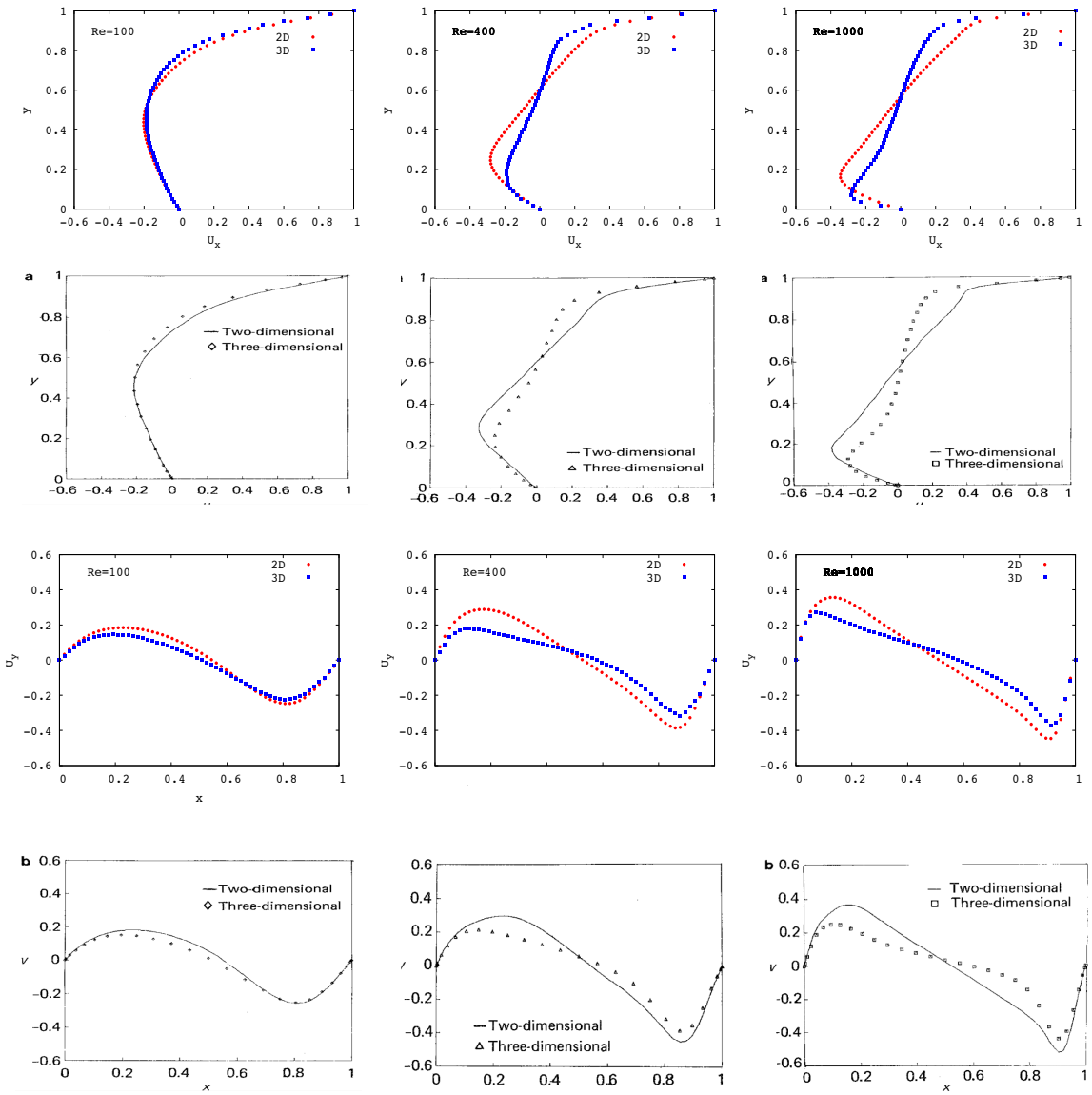


Figure 6. *Cavity in 3D: velocity profiles on the vertical centerline. The first and third rows: present results. The second and fourth rows: results obtained in [KHT87]*

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 (**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 Fig 7).
In many references, different simulations are classified according to the Reynolds number and the

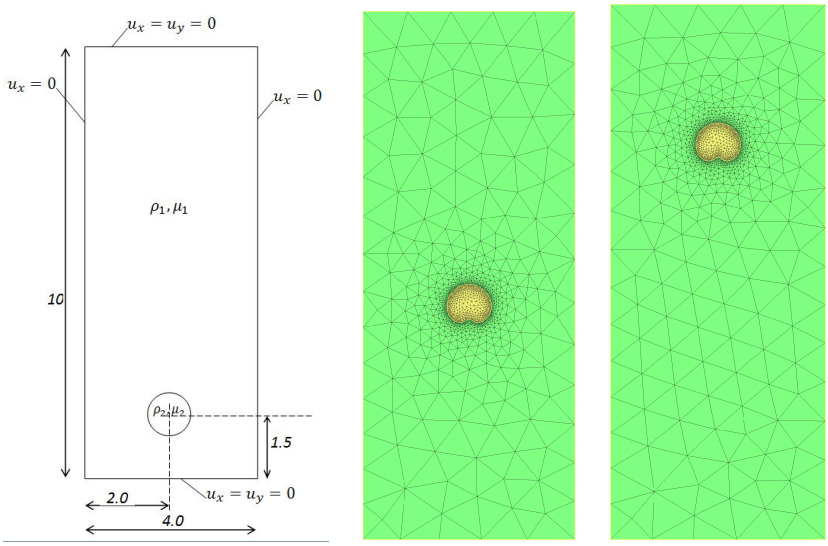


Figure 7. *Rising bubble in 2D: initialization (left) and the evolution at times: t = 5.0s (center), t = 10.0s (right)*

Bond number (also called the Eotvos number) defined as follows:

$$Re = \frac{\rho_1 \sqrt{g}(2r)^{3/2}}{\mu_1}, \quad Bo = \frac{4\rho_1 g r^2}{\gamma} \tag{73}$$

The problem has been set up with densities and viscosities given as : $\rho_1 = 100\text{kg.m}^{-3}, \mu_1 = 0.1\text{kg.m}^{-1}.s^{-1}, \rho_2 = 1.0\text{kg.m}^{-3}, \mu_2 = 0.01\text{kg.m}^{-1}.s^{-1}$. The gravity is $g = 9.81e-3\text{m.s}^{-2}$. The surface tension constant is $\gamma = 6.e-3N.m^{-1}$ and the adapted mesh is such that $hmin = 0.02$ and $hmax = 1.0$ (as shown in Figure 7). 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 the Fig 8 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 Figure 6 of [HSL07] representing tests with low Reynolds number and Bond number (from 10 to 200) which correspond to our simulations with different surface tension coefficient (see (73)).
We have known that during the long simulation, the mass of bubble can't be guaranteed. The advection of the bubble front and the interpolation at each iteration may lead to a loss of mass. We measure the variation of mass at each time step and show it Figure 9. A mass variation of order 2×10^{-3} is observed at each time step.

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 $0.5m$ 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 two-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 the simulation with low Reynolds number, we observe that the shape of the bubble deforms slowly from the beginning, it becomes a dimpled

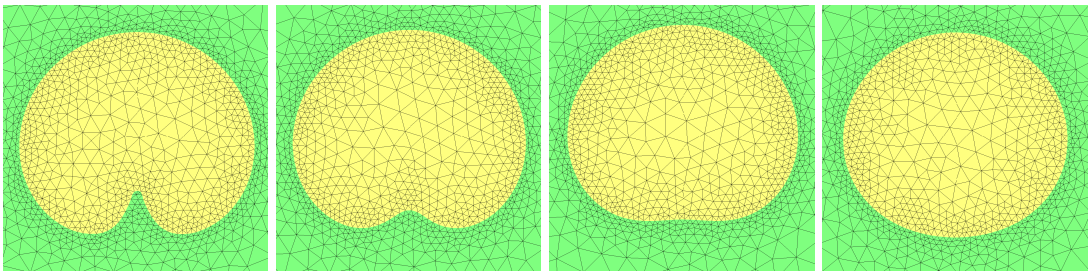


Figure 8. *Rising bubble in 2D: interaction of the surface tension on the final bubble (at time t=10s) with different surface tension coefficients, from left to right:* $\gamma = 6e-5$*;* $\gamma = 6e-3$*;* $\gamma = 2.5e-2$*;* $\gamma = 9e-2$*.*

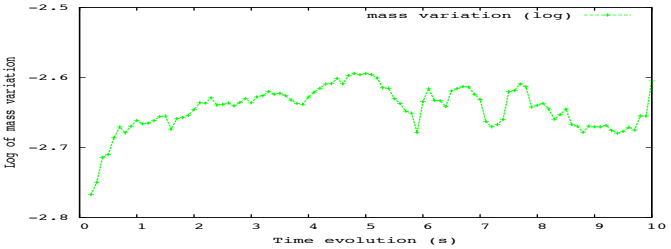


Figure 9. *Rising bubble in 2D: variation of mass correction with time evolution*

ellipsoidal and gets more distorted over time. This result is similar to those in many references with corresponding Reynolds and Bond numbers (see [LAB10], [HSL08] for instance). Figure 10 represents the evolution of the bubble from $t = 0$ to $t = 10$. It can be seen that when the bubble is very close to upper wall of domain, its shape is rapidly distorted.

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 two 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 [Try88] and [FGQ01, CCG08]. The criteria we consider here to parametrize the problem are the two 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 while free-slip conditions are enforced on the vertical sides. The initial interface is set by:

$$\tanh \frac{y - 2 - 0.1cos2\pi x}{0.01\sqrt{2}} = 0$$

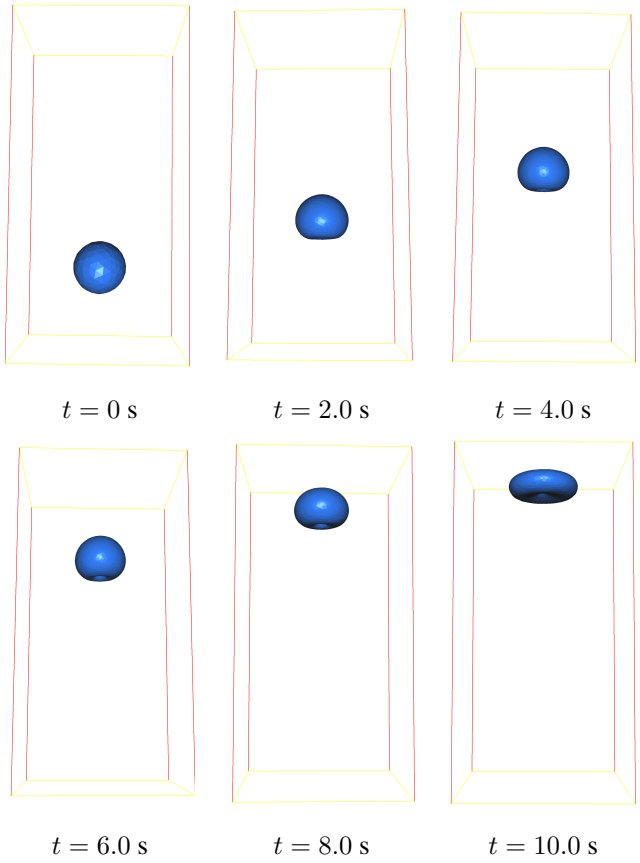


Figure 10. *Rising bubble in 3D: evolution of the interface over time.*

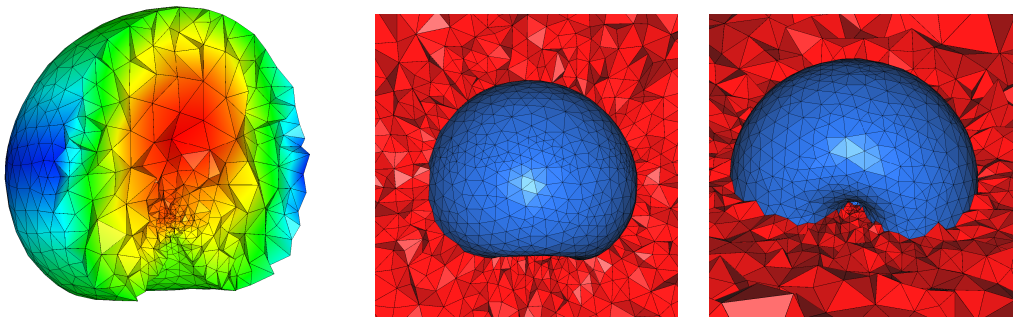


Figure 11. *Rising bubble in 3D: zoom of and adapted mesh at time t = 6.0 s*

We observe that the results satisfies a good symmetry over time while the Atwood number is 0.3. The results displayed in Figure 12 can be compared with Figure 8 of [LKK10]. Let us notice that the mushroom shapes are not identical since their simulation is done with $At = 0.5$ and with higher Reynolds number. We will investigate this test cas in a reduced domain where the computational time is much less.

Figure 13 shows the results for to 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 results suggest 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 Figure 9 of [LKK10]. Secondly, assuming that the

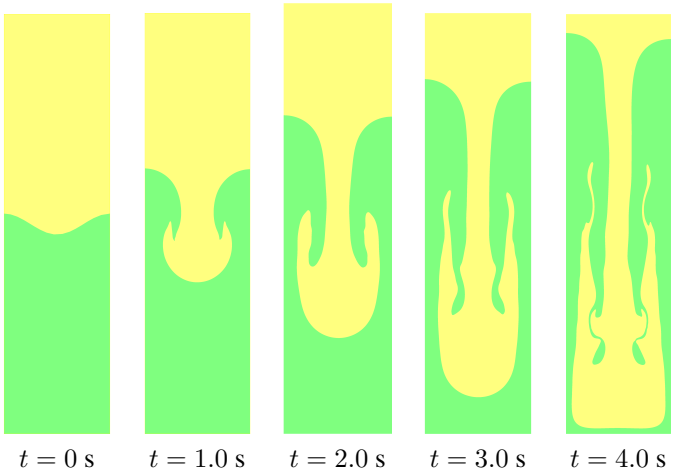


Figure 12. *Rayleigh-Taylor instability in 2D: evolution of the interface over time with* $At = 0.3$

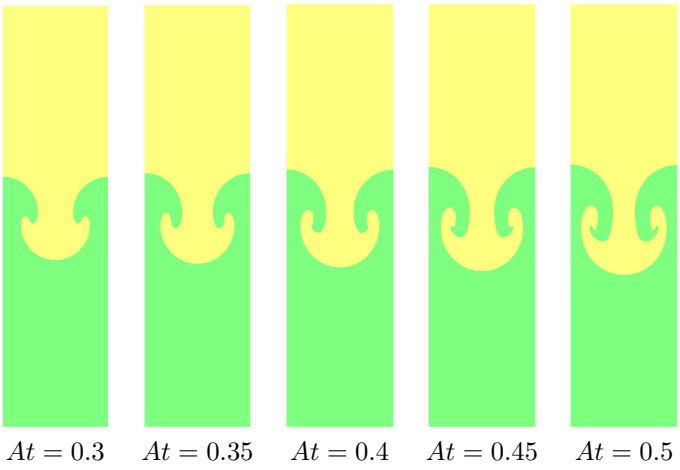


Figure 13. *Rayleigh-Taylor instability in 2D: evolution of the interface with different Atwood numbers.*

symmetry of the initial condition 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 [FGQ01]. The evolution of the interface is plotted in Figure 15 in the time scale of [Try88] which is related to ours by $t_{ref} = t\sqrt{d.At.g}$. We can see that these results are qualitatively close to those of Figure 1 in [FGQ01] (and also with results in [CCG08]). 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.

5.3.2. Rayleigh-Taylor instability in 3D

Finally, we consider the 3D computation of Rayleigh–Taylor instability problem. All the computational conditions are exactly the same as the 2D case and 3D instability has been well captured by proposed scheme, plotted in Figure 18. The results performed in the figure 18 are very close to the results presented in Figure 7 of [LJG96], or Figure 18 of [LFX05]. 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.

In this simulation, we begin with a mesh of 9.901 vertices (50429 tetrahedra). The final mesh

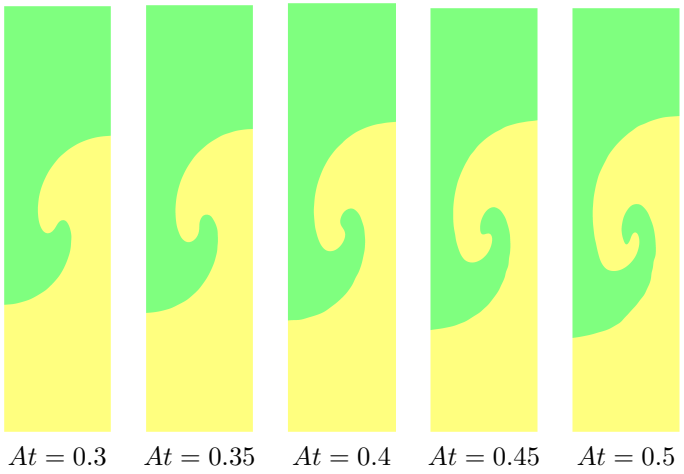


Figure 14. *Rayleigh-Taylor instability in 2D: evolution of the interface with different Atwood numbers on the reduced domain.*

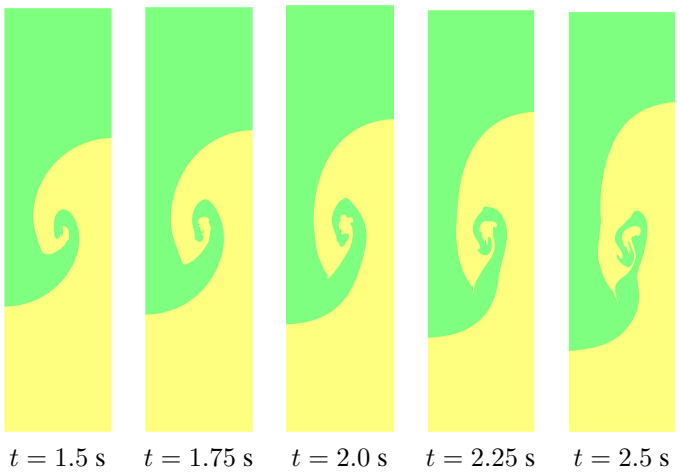


Figure 15. *Rayleigh-Taylor instability in 2D: evolution of the interface with At = 0.5, Re = 1000.*

consists in 103.687 vertices (566.753 tetrahedra) with $hmin = 0.002$, $hmax = 0.1$. Regarding the mesh adaptation, $hmin$ is decreased at each time step in order to well capture the interface with the minimal time of computation.

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 i.e. the fluid is in the vacuum. The initial solution is presented in Figure 20. Let us note that the initial solution is obtained by solving the steady Stokes equation with a upward gravity. The computational domain considered here is the unit circle i.e. 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 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 in the order of 10^{-3}. The test is done over the time interval $[0, 4.435]$ (over 3300 iterations). Figure 21 represents the solution at some time steps.

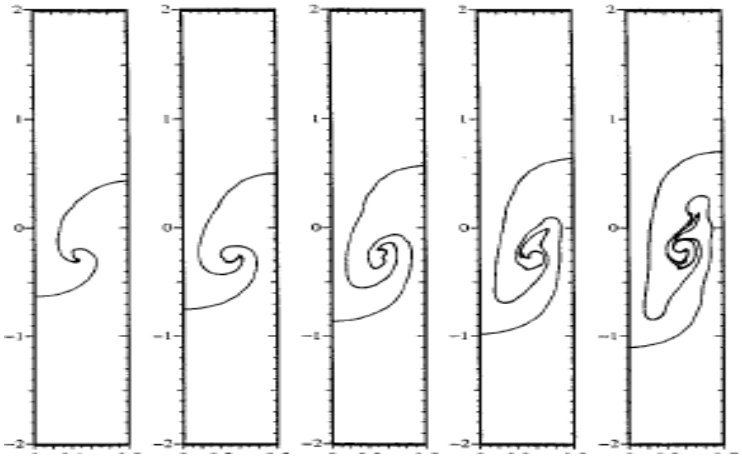


Figure 16. *Rayleigh-Taylor instability in 2D: extracted results of evolution of the interface with At = 0.5, Re = 1000 in [FGQ01].*

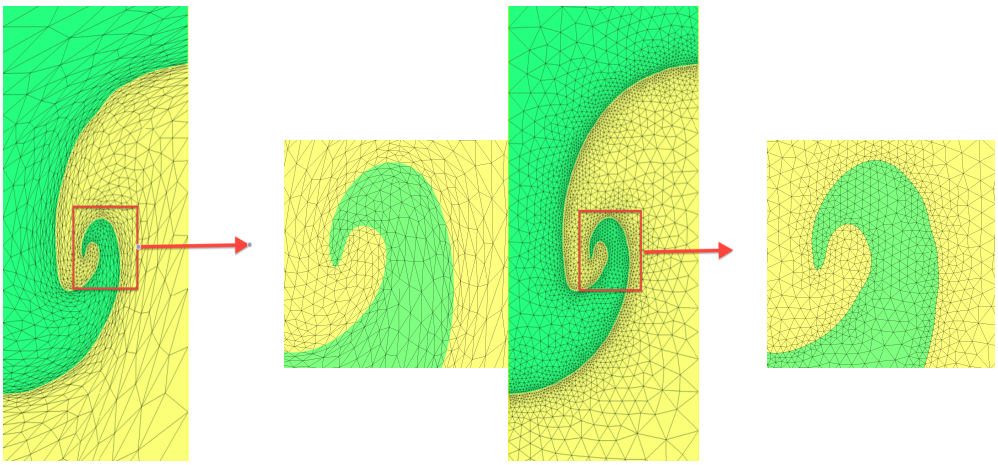


Figure 17. *Rayleigh-Taylor instability in 2D: zoom of adapted mesh in the vicinity of the interface, left: anisotropic mesh, right: isotropic mesh*

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 i.e. $p^a = 0$ 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 [CCT07] and $|\mathbf{U}|$ is the order of the norm of the flow velocity. The value which is taken here f$|\mathbf{U}|$ 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 0-velocity fluid column. The computational domain dimension as well as the fluid column width and height are the same as in [CCT07]: The computational domain is $0.42m \times 0.44m$ whereas the fluid domain is $0.114m \times 0.114m$. Figure 22 compares our result with the physical experiments and the numerical results presented in [CCT07]. The problem is considered in [CCT07] as a bi-fluid air-shampoo problem. Therefore, the fluid equation is solved on the whole computational domain. Similarly to our work, the interface between two flows is captured by solving

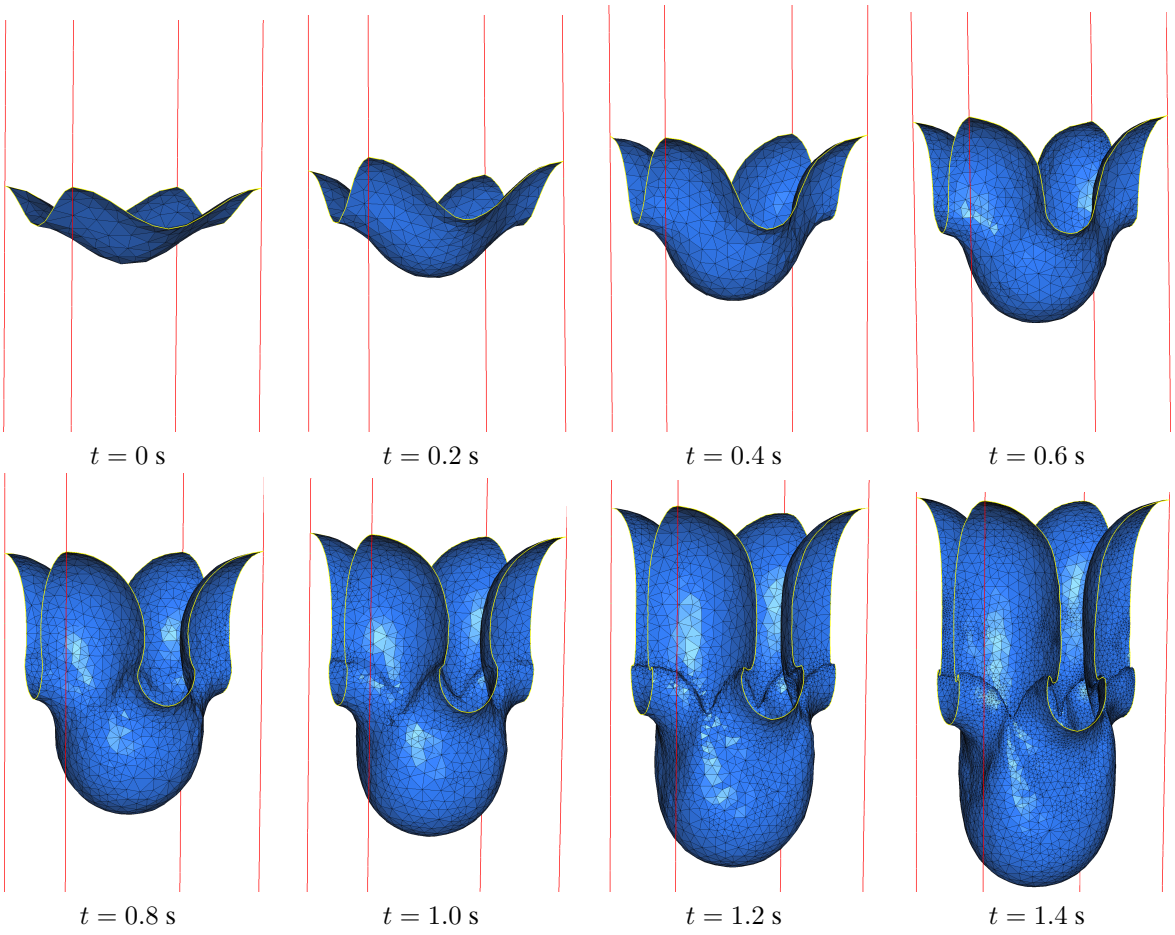


Figure 18. *Rayleigh-Taylor instability in 3D, evolution of the interface in time with At = 0.5.*

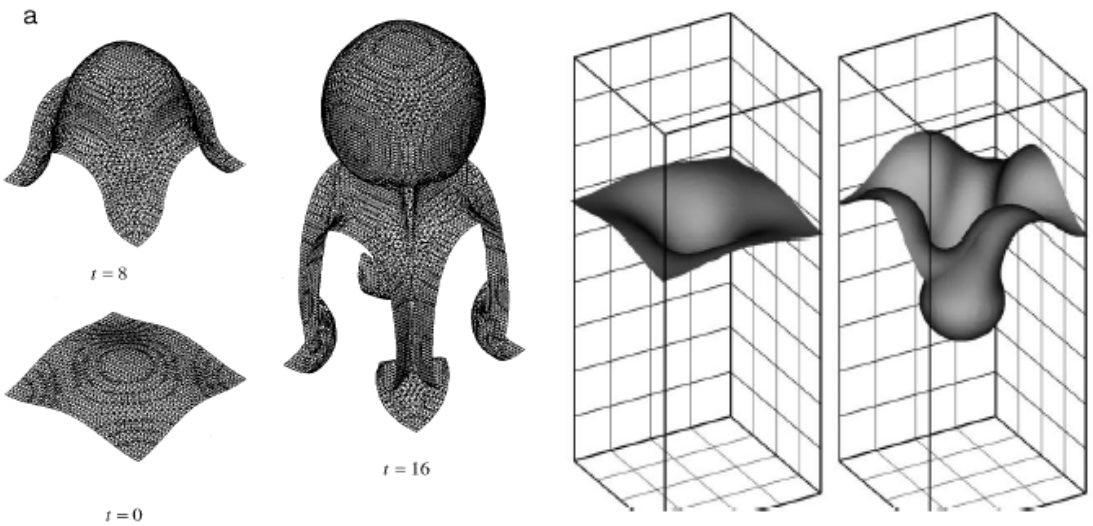


Figure 19. *Rayleigh–Taylor instability in 3D: results obtained in [LJG96] (left) and [LFX05] (right).*

an advection equation (edge-track interface locator technique). The time step we consider here is

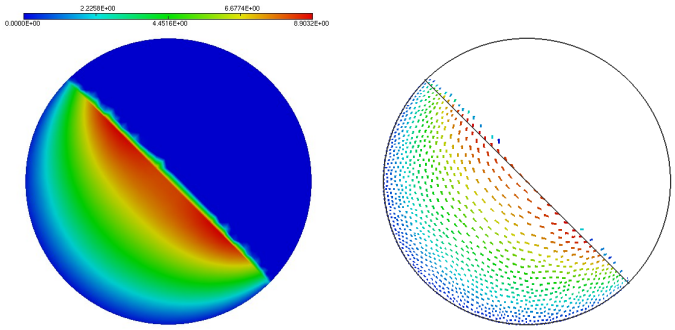


Figure 20. Norm of the initial velocity (on the left) and the associated vector field (on the right).

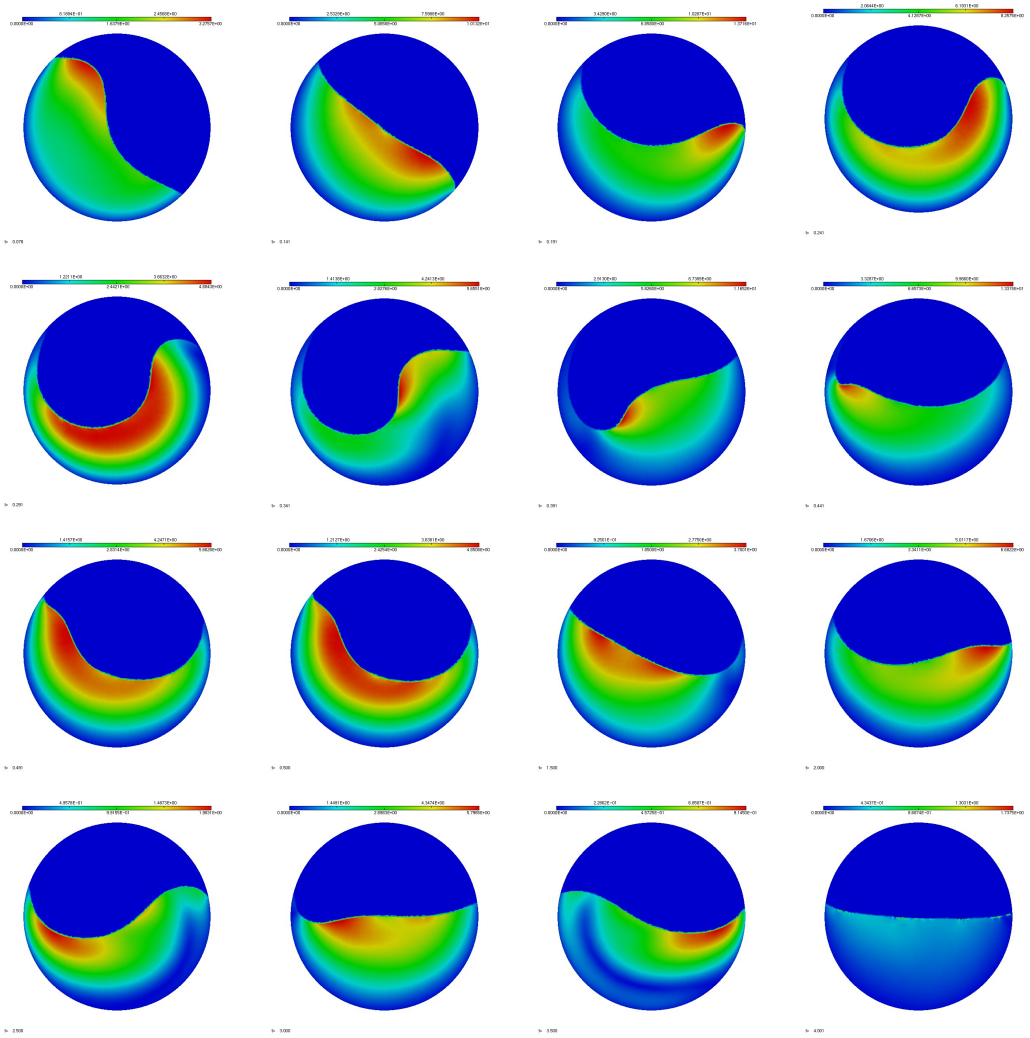


Figure 21. Evolution of a 2-dimensional viscous fluid in the unit circle. The equilibrium is reached at about time $T = 4$.

$\Delta t = 0.01$ for the first ten 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. More over $hgrad = 2.5$. The mesh

adaptation at each step leads to large elements out of the fluid and fine elements inside (see Figure 23). Let us note that the time step in [CCT07] is $\Delta t = 0.001$. Nevertheless, the results are similar.

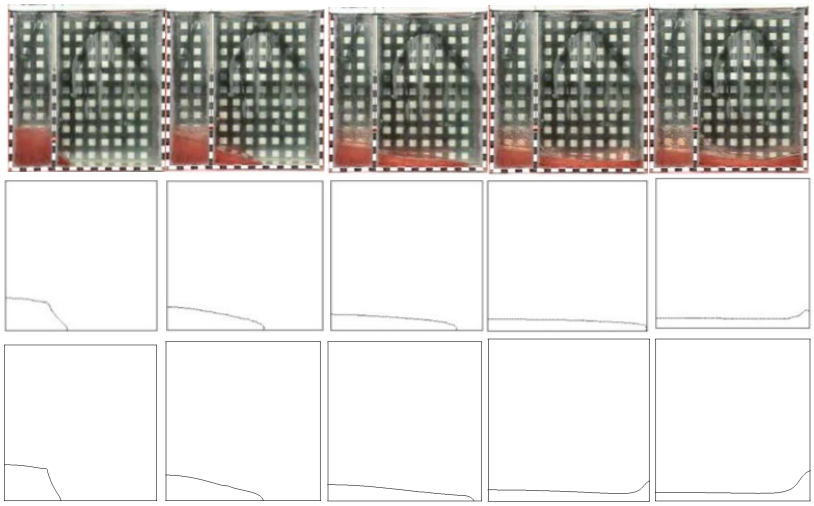


Figure 22. Comparison between physical experiments of [CCT07] (top), the numerical result of [CCT07] with wall friction (middle) and our results (bottom) at times $t = 0.1, 0.2, 0.3, 0.4, 0.5$ (from left to right).

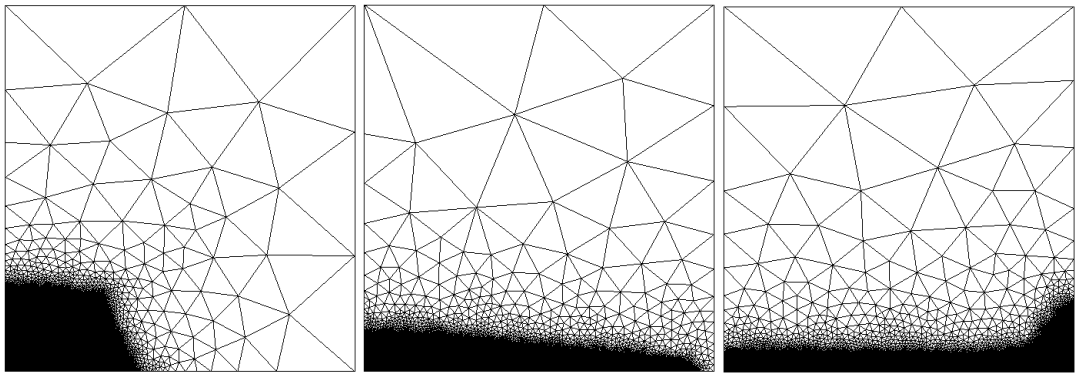


Figure 23. Adapted mesh for the dam break with shampoo at times $t = 0.1$ (left), $t = 0.3$ (middle) and $t = 0.5$ (right). We see that the mesh elements' size is small inside the fluid, very small in the vicinity of the free surface and large out of the fluid far from the surface.

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 [MP94], 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 [MP94]). The model we consider here is the model recalled in [CCT07]. 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

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

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

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

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 μ_{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 $hgrad = 2.5$.

Remark

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.

REFERENCES

[AA02] . J.F. Aujol and G. Aubert. Signed distance functions and viscosity solutions of discontinuous hamilton-jacobi equations. Technical Report 4507, INRIA Technical Report, 2002.

[AF03] . F. Alauzet and P. Frey. Estimateur d'erreur géométrique et métriques anisotropes pour l'adaptation de maillage. partie i: aspects théoriques. Rr-4759, INRIA, 2003.

[AHU58] . K.J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in linear and non-linear programming*, volume II. Stanford University Press, Stanford, Calif, 1958.

[All85] . Geneviève Allain. Un problème de Navier-Stokes avec surface libre et tension superficielle. *Ann. Fac. Sci. Toulouse Math. (5)*, 7(1):29–56, 1985.

[All87] . G. Allain. Small-time existence for the Navier-Stokes equations with a free surface. *Appl. Math. Optim.*, 16(1):37–50, 1987.

[ALTP98] . G. Agresar, J.J. Linderman, G. Tryggvason, and K.G. Powell. An adaptive, cartesian, front-tracking method for the motion, deformation and adhesion of circulating cells. *Journal of Computational Physics*, 143(346-380), 1998.

[AMW98] . D.M. Anderson, G.B. McFadden, and A.A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annual Review of Fluid Mechanics*, 30:139–165, 1998.

[BB73] . J. P. Boris and D. L. Book. Flux-corrected transport. i. shasta, a fluid transport algorithm that works. *Journal of Computational Physics*, 11(1):38 – 69, 1973.

[BdSV71] . A. J. C. Barré de Saint-Venant. 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*, 73:147–154, 1871.

[Beh01] . M. Behr. Stabilized space-time finite element formulations for free-surface flows. *Comm. Numer. Meth. Engrg.*, 11:813–819, 2001.

[BF91] . F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*, volume 15 of *Springer Series in Computational Mathematics*. Springer-Verlag, New York, 1991.

[BFM10] . C. Bui, P. Frey, and B. Maury. A coupling strategy based on anisotropic mesh adaptation for solving two-fluid flows. *International Journal for Numerical Methods in Fluids*, 66(10):1226–1247, 2010.

[BIKL80] . J.P. Benqué, B. Ibler, A. Keramsi, and G. Labadir. A finite element method for the Navier-Stokes equations. In *Proceedings of the third international conference on the finite elements in flow problems*, 1980.

[Bui09] . C. Bui. *Simulation des écoulements bifluides, une stratégie de couplage basée sur l'adaptation de maillage anisotrope*. PhD thesis, Universite Pierre et Marie Curie, 2009.

[Bur03] . M. Burger. A framework for the construction of level-set methods for shape optimization and reconstruction. *Interfaces and Free Boundaries*, 5:301–329, 2003.

[C+96] . Y.C. Chang et al. A level set formulation of eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, 124(2):449–464, 1996.

[CCG08] . C. Calgaro, E. Creus, and T. Goudon. An hybrid finite volume-finite element method for variable density incompressible flows. *Journal of Computational Physics*, 227(9):4671–4696, April 2008.

[CCT07] . M. A. Cruchaga, D. J. Celentano, and T. E. Tezduyar. Collapse of a liquid column: Numerical simulation and experimental validation. *Computational Mechanics*, 39(4):453–476, 2007.

[CDF12] . C.Bui, C. Dapogny, and P. Frey. An accurate anisotropic adaptation method for solving the level set advection equation. *International Journal for Numerical Methods in Fluids*, 70(7):899–922, 2012.

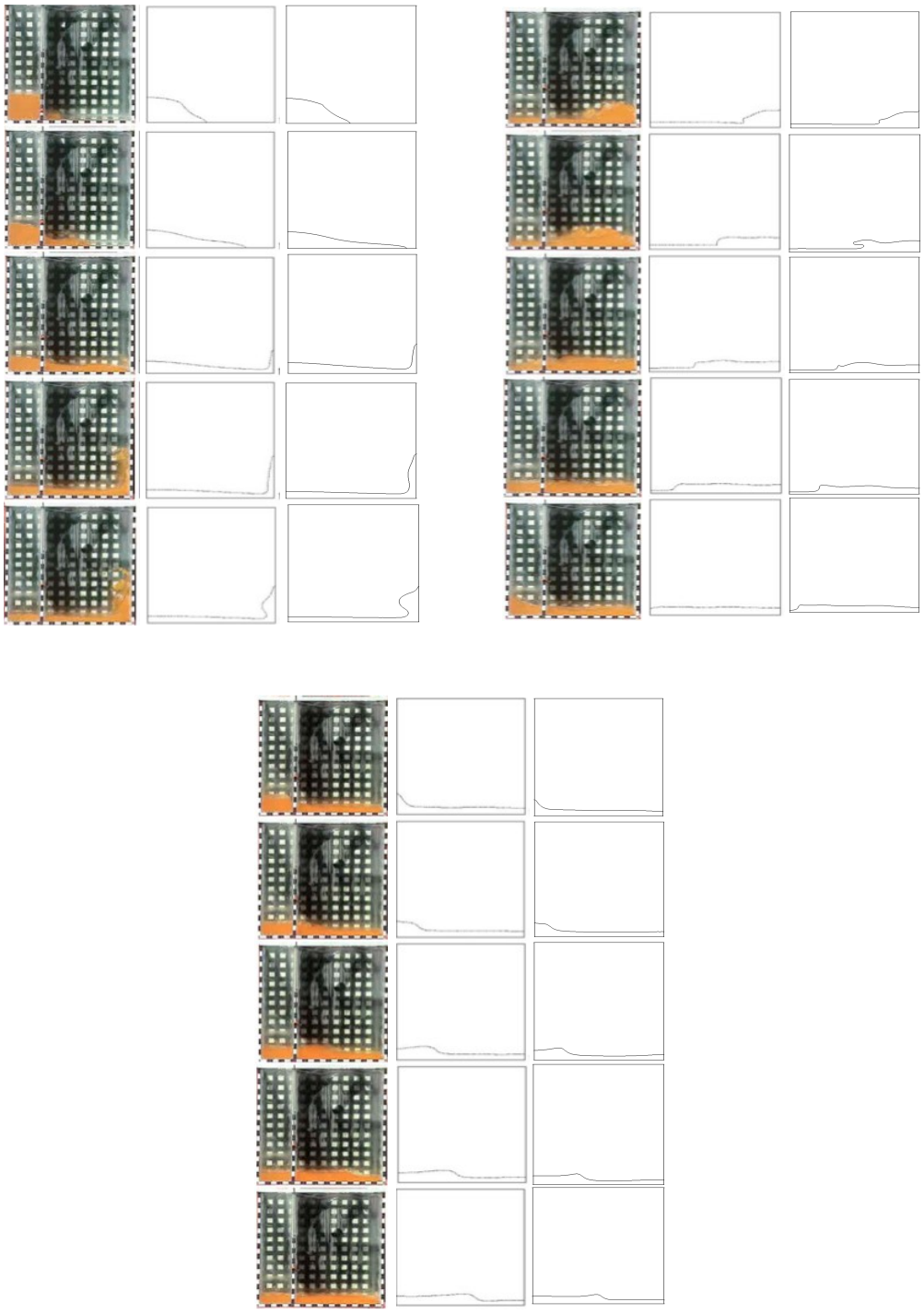


Figure 24. Comparison between physical experiments of [CCT07] (left column), the numerical result of [CCT07] with a discontinuity capturing interface dissipation (middle column) and our results (right column) at times $t = 0.1, 0.2, 0.3, 0.4, 0.5$ (top left), at times $0.6, 0.7, 0.8, 0.9, 1.0$ (top right) and at times $1.1, 1.2, 1.3, 1.4, 1.5$ (bottom). (from top to bottom).

[CIL92] . M. Crandall, H. Ishii, and P. Lions. Users guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc.*, 27:1–67, 1992.

[CR73] . M. Crouzeix and P. A. Raviart. Conforming and nonconforming finite element methods for solving the stationary stokes equations i. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 7(R3):33–75, 1973.

[Den94] . I. Denisova. Problem of the motion of two viscous incompressible fluids separated by a closed free interface. *Acta Appl. Math.*, 37:31–40, 1994.

[Des97] . Benoît Desjardins. Regularity results for two-dimensional flows of multiphase viscous fluids. *Arch. Rational Mech. Anal.*, 137(2):135–158, 1997.

[DF08] . V. Ducrot and P. Frey. Anisotropic level set adaptation for accurate interface capturing. In *In Proc. 17th Int. Meshing Roundtable*, pages 177–194, Sandia National Labs, Pittsburgh, PA, 2008.

[DF11] . C. Dapogny and P. Frey. Computation of the signed distance function to a discrete contour on adapted triangulation. *Journal of Computational Physics*, 2011.

[DF12] . C. Dapogny and P. Frey. Computation of the signed distance function to a discrete contour on adapted triangulation. *Calcolo*, 49(3):193–219, 2012.

[dG06] . F. de Gournay. Velocity extension for the level-set method and multiple eigenvalues in shape optimization. *SIAM J. on Control and Optim.*, 45(1):343–367, 2006.

[dSMN+04]. F. S. de Sousa, N. Mangiavacchi, L. G. Nonato, A. Castelo, M. F. Tomé, V. G. Ferreira, J. A. Cuminato, and S. McKee. A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces. *J. Comput. Phys.*, 198(2):469–499, 2004.

[DT80] . A. Dervieux and F. Thomasset. *A finite element method for the simulation of a Rayleigh-Taylor instablity*, volume 771 of *Lecture Notes in Mathematics*, pages 145–158. Springer-Verlag, Berlin, 1980.

[DU15] . I. Dione and J. M. Urquiza. Penalty: finite element approximation of Stokes equations with slip boundary conditions. *Numer. Math.*, 129(3):587–610, 2015.

[ECG05] . E. Erturk, T.C. Corke, and C. Gokcol. Numerical solutions of 2D steady incompressible driven cavity flow at high Reynolds numbers. *International Journal for Numerical Methods in Fluids*, 48:747–774, 2005.

[EFFM02] . D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics, 183 (2002)*, 183:83–116, 2002.

[EG04] . A. Ern and J. L. Guermond. *Theory and Practice of Finite Elements*, volume 159. Springer, 2004.

[E.n01] . E.nsch. Finite element discretization of the navier-stokes equations with a free capillary surface. *Numer. Math.*, 88:203–235, 2001.

[ET98] . A. Esmaeeli and G. Tryggvason. Direct numerical simulations of bubbly flows. Part 1. Low Reynolds number arrays. *Journal of Fluid Mechanics*, 377:313–345, 1998.

[ET99] . A. Esmaeeli and G. Tryggvason. Direct numerical simulations of bubbly flows. Part 2. Moderate Reynolds number arrays. *Journal of Fluid Mechanics*, 385:325–358, 1999.

[FA05] . P.J. Frey and F. Alauzet. Anisotropic mesh adaptation for CFD computations. *Computer Methods in Applied Mechanics and Engineering*, 194:5068–5082, 2005.

[FG08] . P. Frey and P. L. George. *Mesh generation. Application to finite elements*. J. Wiley & Sons, 2nd edition, 2008.

[FGQ01] . Y. Fraigneau, J-L Guermond, and L. Quartapelle. Approximation of variable density incompressible flows by means of finite elements and finite volumes. *Communications in numerical methods in engineering*, 17:983–902, 2001.

[Fre00] . P. Frey. About surface remeshing. In *Proc. 9th Int. Meshing Roundtable*, pages 123–136. Sandia National Labs, New Orleans, LA, 2000.

[Glo03] . R. Glowinski. *Finite Element Methods for Incompressible Viscous Flows*, volume 9 of *Handbook in Numerical Analysis*. North-Holland, 2003.

[GM99] . M. Gorazd and B. Mohammadi. NSIKE- an incompressible Navier-Stokes solver for unstructured meshes. Technical Report 3644, INRIA Rocquencourt, March 1999.

[GN76] . A. E. Green and P. M. Naghdi. A derivation of equations for wave propagation in water of variable depth. *Journal of Fluid Mechanics*, 78:237–246, 11 1976.

[GP01] . J. F. Gerbeau and B. Perthame. Derivation of viscous Saint-Venant system for laminar shallow water; numerical validation. *Discrete Contin. Dyn. Syst. Ser. B*, 1(1):89–102, 2001.

[GR11] . S. Gross and A. Reusken. *Numerical methods for two-phase incompressible flows*, volume 40 of *Springer Series in Computational Mathematics*. Springer, 2011.

[Gre70] . G. Green. *Mathematical papers*. Edited by N. M. Ferrers. Chelsea Publishing Co., Bronx, N.Y., 1970.

[GT94] . Y. Giga and S. Takahashi. On global weak solutions of the nonstationary two-phase stokes flow. *SIAM J. Math. Anal.*, 25:876–893, 1994.

[GT09] . S. Ganesan and L. Tobiska. A coupled arbitrary lagrangian-eulerian and lagrangian method for computation of free surface flows with insoluble surfactants. *J. Comp. Phys.*, 228:2859–2873, 2009.

[GTBD06] . D. Gerlach, G. Tomar, G. Biswas, and F. Durst. Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows. *International Journal of Heat anh Mass Transfer*, 49:740–754, 2006.

[GWR04] . V. Gravemeier, W.A. Wall, and E. Ramm. A three-level finite element method for the instationary incompressible Navier-Stokes equations, computer methods in applied mechanics and engineering. *Computer Methods in Applied Mechanics and Engineering*, 193(15-16):1323–1366, 2004.

[HAC97] . C.W. Hirt, A.A. Amsden, and J.L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135:203–216, 1997.

[HN81] . C.W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[HRK+10] . E. Hachem, B. Rivaux, T. Kloczko, H. Digonnet, and T. Coupez. Stabilized finite element method for incompressible flows with high Reynolds number. *Journal of Computational Physics*, 299(23):8643–8665, November 2010.

[HSL07] . J. Hua, J. F. Stene, and P. Lin. Numerical simulation of bubble rising in viscous liquid. *Journal of Computational Physics*, 222:769–795, 2007.

[HSL08] . J. Hua, J. F. Stene, and P. Lin. Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method. *Journal of Computational Physics*, 227:3358–3382, 2008.

[KHT87] . H.C. Ku, R.S. Hirsh, and T.D. Taylor. A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations. *Journal of Computational Physics*, 70(2):439–462, June 1987.

[LAB10] . T. Lee L. A. Bower. Single bubble rising dynamics for moderate reynolds number using lattice boltzmann method. *Journal of Computational Physics*, 39:1191–1207, 2010.

[Lan13] . D. Lannes. *The water waves problem*, volume 188 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2013. Mathematical analysis and asymptotics.

[LFX05] . Q. Li, S. Fu, and K. Xu. A compressible Navier-Stokes flow solver with scalar transport. *Journal of Computational Physics*, 204:692–714, 2005.

[LJG96] . X.L. Li, B.X. Jin, and J. Glimm. Numerical study for the three-dimensional Rayleigh-Taylor instability through the TVD/AC scheme and parallel computation. *Journal of Computational Physics*, 126:343–355, 1996.

[LKK10] . H.G. Lee, K. Kim, and J. Kim. On the long time simulation of the Rayleigh-Taylor instability. *International Journal for Numerical Methods in Fluids*, 00:1–25, 2010.

[LR76] . P. Lesaint and P. A. Raviart. Résolution numérique de l'équation de continuité par une méthode du type éléments finis. *Publications mathématiques et informatique de Rennes*, (S5):1–10, 1976.

[MG07] . C. Min and F. Gibou. A second order accurate level set method on non-graded adaptive cartesian grids. *Journal of Computational Physics*, 225:300–321, 2007.

[MP94] . B. Mohammadi and O. Pironneau. *Analysis of the K-epsilon turbulence model*. Recherches en mathématiques appliquées. Wiley, 1994.

[OS88] . S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:1–49, 1988.

[Pir82] . O. Pironneau. On the transport-diffusion algorithm and its applications to the Navier-Stokes equations. *Numerische Mathematik*, 38(3):309–332, 1982.

[Pir89] . O. Pironneau. *The finite element methods for fluids*. J. Wiley & Sons, 1989.

[Pir10] . O. Pironneau. Finite element characteristic methods requiring no quadrature. *Journal of Scientific Computing*, 43:402–415, 2010.

[PS10] . Jan Prüss and Gieri Simonett. On the two-phase Navier-Stokes equations with surface tension. *Interfaces Free Bound.*, 12(3):311–345, 2010.

[Qua09] . A. Quarteroni. *Numerical Models for Differential Problems*, volume 2 of *MS-A*. Springer, 2009.

[S+98] . M. Sussman et al. An improved level set method for incompressible two-fluid flows. *Computers and Fluids*, 27:663–680, 1998.

[Set99] . J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[SS03] . J.A. Sethian and P. Smereka. Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics*, 35:341–372, 2003.

[SSO94] . M. Sussman, P. Smereka, and S.J. Osher. A level set approach to computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.

[SZ99] . R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Ann. Rev. Fluid Mech.*, 31:567–603, 1999.

[T+01] . G. Tryggvason et al. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169:708–759, 2001.

[Tan93] . N. Tanaka. Global existence of two-phase nonhomogeneous viscous incompressible fluid flow. *Comm. in Partial Differential Equations*, 18:41–81, 1993.

[Tan96] . Atusi Tani. Small-time existence for the three-dimensional Navier-Stokes equations for an incompressible fluid with a free surface. *Arch. Rational Mech. Anal.*, 133(4):299–331, 1996.

[TFC+01] . M.F. Tome, A.C. Filho, J.A. Cuminato, N. Mangiavacchi, and S. Mckee. Gensmac3d: a numerical method for solving unsteady three-dimensional free surface flows. *International Journal for Numerical Methods in Fluids*, 37(7):747–796, 2001.

[Try88] . G. Tryggvason. Numerical simulation of the Rayleigh-Taylor instability. *Journal of Computational Physics*, 75:253–282, 1988.

[TT95] . Atusi Tani and Naoto Tanaka. Large-time existence of surface waves in incompressible viscous fluids with or without surface tension. *Arch. Rational Mech. Anal.*, 130(4):303–314, 1995.

[UG82] . C.T Shin U. Ghia, K.N. Ghia. High-resolution for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3):387–411, 1982.

[UT92] . S.O. Unverdi and G. Tryggvason. A front-tracking method for viscous incompressible multi-fluid flows. *Journal of Computational Physics*, 100:25–37, 1992.

[Ver85] . R. Verfürth. Finite element approximation of steady Navier-Stokes equations with mixed boundary conditions. *RAIRO Modél. Math. Anal. Numér.*, 19(3):461–475, 1985.

[Ver87] . R. Verfürth. Finite element approximation of incompressible Navier-Stokes equations with slip boundary condition. *Numer. Math.*, 50(6):697–721, 1987.

[Ver91] . R. Verfürth. Finite element approximation of incompressible Navier-Stokes equations with slip boundary condition. II. *Numer. Math.*, 59(6):615–636, 1991.

[XZZ13] . Li Xu, Ping Zhang, and Zhifei Zhang. Global solvability of a free boundary three-dimensional incompressible viscoelastic fluid system with surface tension. *Arch. Ration. Mech. Anal.*, 208(3):753–803, 2013.