Modélisation des problèmes bi-fluides par la méthode des lignes de niveau et l'adaptation du maillage : Application à l'optimisation des formes

Thi Thanh Mai Tran

Université Pierre et Marie Curie — Laboratoire Jacques-Louis Lions

Thèse de doctorat de l'Université Pierre et Marie Curie

Spécialité Mathématiques

École Doctorale de Science Mathématiques de Paris Centre

présentée par

Thi Thanh Mai TA

pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

Modélisation des problèmes bi-fluides par la méthode des lignes de niveau et l'adaptation du maillage.

Application à l'optimisation des formes.

dirigée par Pascal Frey

Soutenue le 07 Janvier 2015 devant le jury composé de :

M. Florian DE VUYST	Ecole Normal Supérieure de Cachan	Rapporteur
M. Pascal FREY	Université Pierre et Marie Curie	Directeur de thèse
M. Yvon MADAY	Université Pierre et Marie Curie	Président
M. Bijan MOHAMMADI	Université Montpellier II	Rapporteur
M. Franck PIGEONNEAU	Saint-Gobain Recherche	Invité
M. Yannick PRIVAT	Université Pierre et Marie Curie	Examinateur
M. Pierre SARAMITO	CNRS Grenoble	Examinateur

Contents

Introduction

We present here two problems related to numerical simulation of viscous incompressible fluids that will be described and analyzed in the two main parts of this thesis:

- Part 1: Numerical simulation of the two-phase flow problem.
- Part 2: An example of shape optimization in fluid mechanics.

This general introduction is aim at enlightening the structure of this manuscript, it contains neither technical details, nor references on the aforementioned topics (which are left for the detailed introduction of each part). We sketch out here the main problems, hence leading to define the content of each chapter.

The first concern of this thesis is the problem of two fluids flow or two-phase flow, i.e, we are interested in the simulation of the evolution of an interface (or a free surface) between two immiscible viscous fluids or two phases of a fluid. The main challenge in solving these problems is to define and to handle an accurate representation of the interface that separates the different fluids, even in the event of geometry or topology changes. In strongly deforming flows, this difficulty is related to maintaining a high-quality interface while conserving the total mass and allowing some extreme changes like folding, merging and breaking (see Figure 1). Another difficulty is the accurate computation or estimation of some algebraic quantities like the normal vector and the local curvature along the interface. Furthermore, a surface tension force must be considered in the model and accurately evaluated.

It seems obvious that any proposed approach for solving this problem involves two major problems, related respectively to:

i. The resolution of the geometry and the evolution of the interface.
ii. The resolution of the equation governing the "movement" of the flows in a non-homogeneous medium.

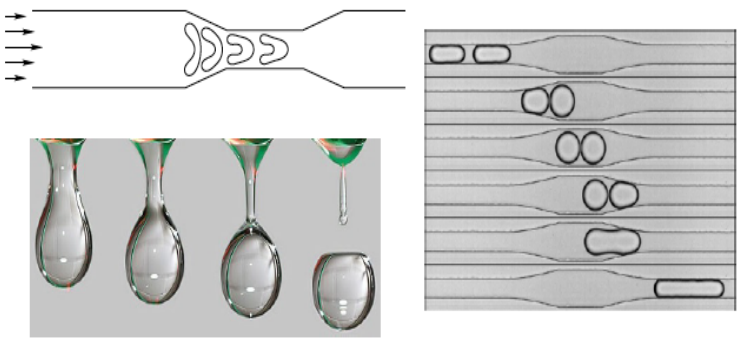


Figure 1: *Extreme changes of the interface: folding (top, left), breaking (bottom, left) and merging (right).*

Among the numerous methods that have been developed for the simulation of free surface flows, the *level set method* has been extensively used in the past few years due to its simplicity and efficiency. More specifically, we propose here a general scheme for solving two fluids flow or two-phase flows which takes advantage of the flexibility of the level set method for capturing evolution of the interfaces, including topological changes. Unlike similar approaches that solve the flow problem and the transport equation related to the evolution of the interface on Cartesian grids, our approach relies on an adaptive unstructured mesh to carry out these computations and enjoys an exact and accurate description of the interface. The explicit representation of the manifold separating the two fluids will be extracted to compute approximately the surface tension as well as some algebraic quantities like the normal vector and the curvature at the interface.

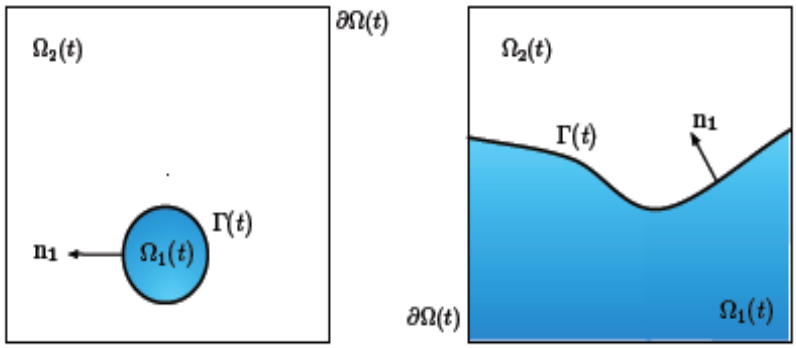


Figure 2: *Illustration of a computational domain* $\Omega(t)$ *composed of two phases* $\Omega_1(t)$ *and* $\Omega_2(t)$ *having different material properties and separated by an interface* $\Gamma(t)$ *which may evolve in time.*

More precise description, the moving interface $\Gamma(t)$ between two immiscible fluids occupying two sub-domain $\Omega^i(t)(i = 1, 2)$ (see Figure 2) is represented as zero-level set of a *level set function* $\phi(\mathbf{x}, t)$: $\Gamma(t) = \{\mathbf{x} \in \Omega(t) : \phi(\mathbf{x}, t) = 0\}$. Hence, the evolution of the interface is governed by the advection of the level set function ϕ:

$$\frac{\partial \phi}{\partial t} + \mathbf{u}.\nabla\phi = 0$$

where the advection vector $\mathbf{u}$ is the velocity of the flows governed by the incompressible Navier-Stokes equations solving:

$$\begin{cases} \rho^i \left(\dfrac{\partial \mathbf{u}^i}{\partial t} + (\mathbf{u}^i \cdot \nabla)\mathbf{u}^i \right) - \mu^i \Delta \mathbf{u}^i + \nabla p^i &= \rho^i \mathbf{f}^i \text{ in } \Omega^i \, (i = 1, 2) \\ \text{div}\mathbf{u}^i &= 0 \quad \text{ in } \Omega^i \, (i = 1, 2) \end{cases}$$

with the interfacial condition on Γ :

$$\mathbf{u}^1 - \mathbf{u}^2 = 0$$
$$(\sigma^1 - \sigma^2).\mathbf{n}^1 = -\gamma\kappa\mathbf{n}^1$$

The surface tension is taken into account as the term of a *localized force* $-\gamma\kappa\mathbf{n}^1$ at the interface.

In a nutshell, the resolution of a two-fluid problem is summarized by the steps involves the following ingredients:

- a numerical method for solving incompressible Navier-Stokes equations (Chapter 1);
- a geometrical treatment to evaluate the surface tension (Chapter 2);
- a numerical method for solving the level set advection (Chapter 2);
- the techniques of mesh adaptation (Chapter 3).

Let us finish the introduction of our first part by specifying the role of mesh adaptation in this scheme. It is obvious that no numerical method is completely exact in solving the PDE problem at hand, hence, we need a discretized computational domain. However, the accuracy of numerical solutions or the mass loss/gain can generally be improved with mesh refinement. The question that arises is related to where and how to refine the mesh. At each time, our mesh adaptation produces the adapted mesh based on the geometric properties of the interface and the physical properties of the fluid, simply speaking, only one adapted mesh at each time step to assume both the resolution of Navier-Stokes and the advection equations. It answers to the need for an accurate representation of the interface and an accurate approximation of the velocity of fluids with a minimal number of elements, then decreasing the amount of computational time.

The second part of this thesis is related to shape optimization in fluid mechanics. Broadly speaking, this problem can be formulated as the minimization of an objective function $J(\Omega)$ of the domain variable Ω, via the solution u_Ω of a mechanical problems (Stokes system in our context). By the classical way, the study of the derivative of J with respect to the domain makes it possible to compute a descent direction V for J from a given shape Ω, then Ω is updated to obtained the new shape $\Omega(t)$. However, there are a lot of difficulties in implementing this idea:

- The practical computation of the descent direction V usually involves the resolution of one, or several PDE systems posed on Ω (Stokes system and adjoint system). At this point, we need to construct the adjoint system which is also solved on Ω and leads to the resolution of V.
- The evolution of the shape Ω along the velocity field V is fairly straightforward in the theoretical framework, but unfortunately much harder in the numerical practice. In particular, it inherently depends on how Ω is parametrized. For instance, if Ω is described by a mesh, the evolution of Ω is naturally translated to the moving of the associated vertices in the direction of V. This sometimes produces an ill-shaped (or even invalid) mesh for the new shape (see Figure 3). In general, mesh evolution is a difficult issue, especially in three space dimensions.

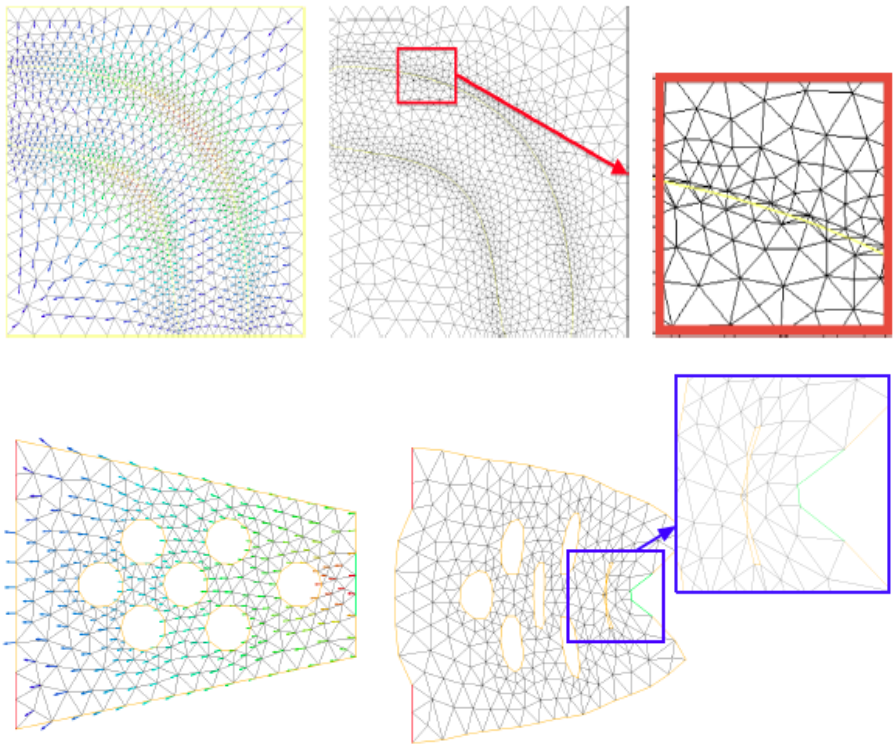


Figure 3: *Ilustration for the cases of ill-shaped mesh (top) and invalid mesh (bottom): (left) A velocity field V, defined at the vertices of a mesh; (right) deformed mesh.*

So, in order to satisfy all the requirements of the computation of a descent direction for J and of the description of the domain evolution, several authors proposed to combine the aforementioned techniques of shape sensitivity analysis with the level set method. In these approaches, the general idea is enclosing all the possible shapes in a fixed, large computational domain D equipped with a fixed mesh $\mathcal{D}$ and to describe any shape Ω from an implicit point of view, via a scalar *level set function* $\phi : D \to \mathbb{R}$ which fulfills the following properties (see Figure 4):

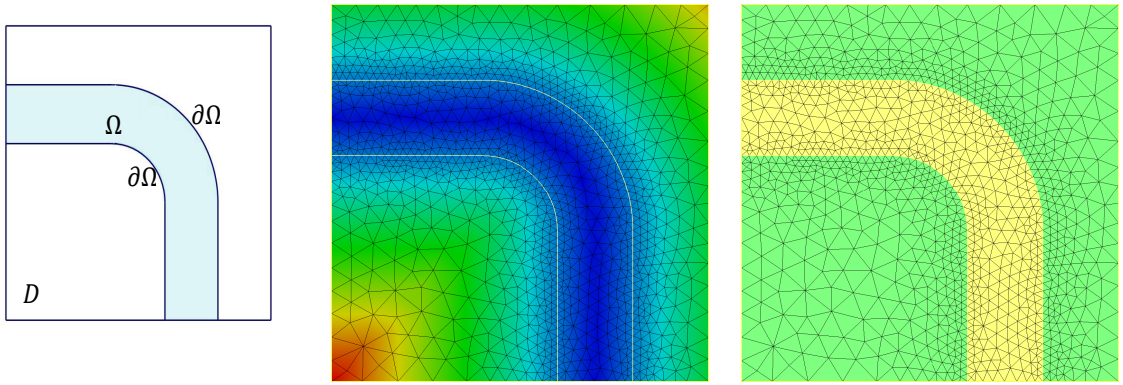


Figure 4: *(Left) A shape $\Omega \subset D$, (Center) representation of shape Ω by associated level set function and (Right) triangular mesh of D enclosing a mesh of Ω (yellow elements).*

$$\forall x \in D, \begin{cases} \phi(x) < 0 \,\text{if}\, x \in \Omega \\ \phi(x) = 0 \,\text{if}\, x \in \partial\Omega \\ \phi(x) > 0 \,\text{if}\, x \in D \setminus \Omega \end{cases}$$

The evolution of $\Omega(t)$ along the velocity field V is reformulated in terms of an associated level set function $\phi(.,t)$ as the following *level set advection* equation:

$$\frac{\partial \phi}{\partial t} + V.\nabla\phi = 0$$

which can be solved on D, using its mesh $\mathcal{D}$.

The computation of V is however not so easy in this context: we indeed evoked the fact that it requires solving PDE systems posed on Ω - a mesh of which is not available. These systems must then be approximated as PDE systems posed on the whole domain D (in the context of linearized elasticity, this is generally achieved by using the *Ersatz material* approach), and solved on the fixed mesh $\mathcal{D}$. This operation may turn out to be difficult in the study of mechanical models which require a high accuracy in the description of the boundaries of shapes.

Our approach arises from the observation that a slight modification in this methodology allows us to retain its great versatility when it comes to tracking the evolution of shapes, while benefiting from an exact description of any considered shape $\Omega \subset \mathcal{D}$.
Indeed, since a mesh of a shape Ω is needed at each time, one could imagine to modify $\mathcal{D}$ in such a way that an explicit discretization of Ω appears in it (see Figure 4). Hence, the computation of the descent direction V from Ω would become straightforward, and would not involve any approximation of the considered mechanical problem. Carrying out this idea inherently requires to be able to perform local mesh operations on $\mathcal{D}$, hence to work with fully unstructured meshes; in our case, we shall use simplicial meshes (i.e triangulation in two dimensions, tetrahedralization in three dimensions). Moreover, it implies the following ingredients:

- a numerical method for solving the mechanical problem (Stokes system and adjoint system). This is the reduced system (not containing the advection operation) of the Navier-Stokes equations that have been solved in Chapter 1.
- a numerical method for solving the level set advection equation on a simplicial computational mesh of D. This is one of the purposes of the work in Chapter 2 (Section 2.3)
- a meshing technique for discretizing explicitly a shape known via an associated level set function, on a mesh of D. This is described in Chapter 3.

Let us now outline the description of the different chapters of this thesis.

The plan of the thesis

Chapter 1: Numerical resolution of Navier-Stokes equations

Chapter 1 focuses on the numerical solving of the incompressible Navier-Stokes equations for one viscous fluid. We rely on the Lagrange-Galerkin formulation for treating the convection operator $\frac{\partial}{\partial t} + \mathbf{u}.\nabla$. By this way, at each time step the Navier-Stokes problem becomes an unsteady Stokes problem that can be solved using the finite element method. The resolution of the linear system from the finite element discretization is based on the Uzawa method or the Penalty method using Conjugate Gradient algorithm and a preconditionned Conjugate Gradient algorithm. A classical numerical test Lid-driven cavity is investigated in order to valid and assess our code with considered options in two and three dimensions spaces.

Chapter 2: The scheme for two-phase flow simulation

The second chapter begins with the introduction of the model of two-fluid flows with the interface. Our model is based on the incompressible Navier-Stokes equation for the velocity field of the fluids, and the level set advection equation for the evolution of the interface. At each time step, the level function is transported by the velocity field of the fluid to define a new position of the interface. The resolution of this equation is carried out using the method of characteristics. Here, the characteristics are approximated by piecewise affine sequences using a Euler or 4^{th}-order Runge-Kutta schemes. Due to the explicit discretization of the interface in the computational mesh, the surface tension can be estimated geometrically. As known, a drawback of the level set method is the difficulty in preserving the regularity of the interface and the conservation of the mass. To overcome these problems, a redistancing procedure and a technique of mass correction are supplemented to our scheme. A numerical scheme based on mesh adaptation for two-phase flow simulation is proposed at the end of this chapter.

Chapter 3: Mesh adaptation

In this chapter we recall the major concepts and aspects of two mesh adaptation processes that have been used successfully in the general schemes of two-fluid flows as well as for shape optimization.

- *Anisotropic mesh adaptation*: the adapted mesh is generated based on the information stored in a *metric tensor*. This metric tensor is constructed from the requirements of the accuracy of geometrical approximation as well as from a numerical approximation of the concerned problems.
- *Discrete three dimensional domain remeshing*: The adapted mesh is generated based on a *local size map*. The local size map of the adapted mesh is computed to satisfy a given geometric approximation as well as to comply with numerical approximation constraints.

In these mesh adaptation processes, the final adapted mesh should be "well-shaped" mesh with respect to some quality criterions.

Chapter 4: Numerical examples

This chapter presents several examples of numerical resolutions of incompressible Navier-Stokes equations as well as of two-fluid flows involving an interface. Each case test will be considered in two and three dimensions. The first example is the Lid-driven cavity. This test has been presented in Chapter 1 to valid all implements in our code. In this chapter, examples are exhaustively investigated

and compared with the results gathered in the references. Not only taking care of the number of the vortices, the streamlines, we compare also the position of the primary vortex and the velocity profile as well as the pressure of the fluid.
The efficiency and reliability of our numerical scheme for two-fluid flows are examined throughout several examples for which the surface tension is taken into account, namely, the rising bubble, the Rayleigh-Taylor instability, the coalescence of two rising bubbles. The results obtained are consistent with those found in the literature and show that the proposed scheme is able to manage complex interfacial movements, include topology changes.

Chapter 5: Shape optimization in fluid mechanics

The chapter begins with the shape sensitivity analysis based on Hadamard's boundary variation method in which the computation of shape derivative as well as the adjoint system are considered in the context of Stokes system. From the inferring of the descent direction, we play out a global algorithm for shape optimization in fluid mechanics. This algorithm is set in the context of level set method on unstructured meshes and mesh evolution to obtain a numerical scheme for shape optimization in fluid mechanics. A numerical example with the objective function of energy dissipation is presented to demonstrate the efficiency of the proposed scheme.

Part I

Numerical simulation of the problem two-fluid flows

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 problems. Actually, research on this topic has started in the last few years. However, there are several specific issues relevant to two-fluid 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 is certainly a difficult challenge that raises many related and yet mostly unresolved problems: the coupling between the fluid dynamics and the interface 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 this topic have been published in the engineering literature, at the noticeable exception of the monograph [GR11]; see for instance [AMW98] or [TBE$^+$01] for an overview of numerical methods for the simulation of multiphase flows.

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 2.1) have been analyzed only for special cases (e.g. unbounded domain $\Omega = \mathbb{R}^3$ and $\lim_{|x|\to\infty} \mathbf{u}(x,t) = 0$) and when the initial interface $\Gamma(0)$ is a closed manifold [Den94]. 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. 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-fluid flows.

Let us briefly recall the two most important methods for *interface representation*. 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.

1. *Interface tracking:* When the interface regularity is sufficient, then normal, curvature and immiscibility condition like V_Γ 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 $T > 0$ sufficiently small, there is a one to one mapping between $(\mathbf{x}, t) \in \Omega \times [0, T]$ and a unique $\mathbf{x_0}$ 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 $\mathbf{x} \in \Gamma(t)$ satisfying (2). 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 $t + \Delta t$. After several time steps, the equidistribution property is usually lost and marker points have to be redistributed along the new interface $\Gamma(t + \Delta t)$. 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 [Bae01, 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 $\Omega_1(t)$, one of the two subdomains delimited by the interface $\Gamma(t)$ hereby defined as $\Gamma(t) = \partial\Omega_1(t) = \partial\text{supp}(\chi_1(\cdot, t))$. 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 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, and 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 pratice. Current works in progress focus on solving these problems and carry a lot of promises [GTBD06, TCC$^+$00, dSMN$^+$04].

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:

$$\phi(\mathbf{x}) < 0, \quad \mathbf{x} \in \Omega_1(0), \qquad \phi(\mathbf{x}) > 0, \quad \mathbf{x} \in \Omega_2(0), \qquad \phi(\mathbf{x}) = 0, \quad \mathbf{x} \in \Gamma(0) \,. \tag{5}$$

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:

$$\phi(\mathbf{X}(\mathbf{x}_0,t_0;t))=\phi(\mathbf{x}_0)\,,\qquad \mathbf{x_0}\in\Omega,\,t\geq 0\,, \tag{6}$$

and when differating this advection equation with respect to t it comes:

$$\frac{\partial\phi}{\partial t}+\mathbf{u}\nabla\phi=0\,,\quad \text{for all } \frac{1}{2}\mathbf{x}\in\Omega\,,\ t\geq 0\,. \tag{7}$$

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

$$\Gamma(t)=\{\mathbf{x}\in\Omega\,;\quad \phi(\mathbf{x})=0\}\,. \tag{8}$$

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 (7) 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 $\phi=d(\mathbf{x},\Gamma(0))$ to the interface (equation (8)) at $t=0$. The velocity field is the result of Navier-Stokes equation and the transport equation (7) is discretized using suitable numerical methods in space and time (see Sections 2.1 and 2.3 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]. In the remainder of this manuscript, we will restrict our numerical investigation to only one of these, the *level set* representation.

In this part, we will describe in details our work and propose a general strategy for solving two-phase 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 numerical resolution of incompressible Navier-Stokes equations is presented in Chapter 1, our approach is based on the Lagrange-Galerkin scheme was first introduced by [BIKL80] and studied in [Pir82]. Chapter 2 presents the combining an implicit method for dealing with the domain evolution and an explicit representation of the manifold separating the two fluids. This idea is obviously not new but unlike similar approaches that solve the flow problem and the transport equation on Cartesian grids [ALTP98, GTBD06, EFFM02, MG07, BCdV14], our approach relies on an adaptive unstructured mesh to carry out these computations. Chapter 3 is devoted to the techniques of mesh adaptation as well as the construction of adapted mesh process. Many numerical tests are investigated in Chapter 4 in two and three dimensions in order to assess the efficiency and reliability of our proposed numerical schemes.

Chapter 1

Numerical resolution of Navier-Stokes equations

Contents

The purpose of this chapter is to present the numerical methods we have designed to resolve Navier-Stokes equations. Our approach is based on the combination of the method of characteristics with a finite element formulation. This method was first introduced by Benqué [BIKL80] and analyzed in [Pir82], and is also known as the Lagrange-Galerkin method. The main idea behind this method is that the convection operator (the non linear term) of the Navier-Stokes equation can be

turned into a total derivative (also called *material derivative*) by using a Lagrangian formulation. The outline of this chapter is the following: After introducing the incompressible Navier-Stokes equations for a single viscous fluid in Section 1.1, we express the operator of velocity $\frac{\partial}{\partial t}+\mathbf{u}.\nabla$ by material derivative $\frac{d}{dt}$ which allows us to discretize in time this problem in Section 1.2. In this approach, the non-linear part of Navier-Stokes equations is associated with a Cauchy problem, and this leads us to solve an unsteady Stokes problem at each time step. The following sections are concerned with the numerical resolution of a generalized Stokes problem by the finite element method: section 1.3 presents the variational formulation for different boundary conditions. Section 1.4 presents the discrete formulation corresponding to Lagrange-Galerkin finite elements of $\mathbb{P}1\text{bubble}/\mathbb{P}1$ and $\mathbb{P}2/\mathbb{P}1$. Section 1.5 describes the matrix formulation and the resolution of the resulting linear system discretized by finite elements spaces. In the last section 1.6 we detail a numerical test, the Lid-driven cavity in order to validate and to compare the numerical resolution of the incompressible Navier-Stokes equations in two and three dimensions with different parameters and settings related to: finite elements, the order of the scheme for approximating of the characteristic lines and the resolution of linear systems.

1.1 The Navier-Stokes equations for incompressible fluid

Let Ω be a bounded domain in $\mathbb{R}^d (d \in \{2,3\})$ and let denote Σ, its boundary. We consider the Navier-Stokes equations governing unsteady incompressible flows in general:

$$\begin{cases} \rho\left(\dfrac{\partial \mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{u}\right)-\mu\Delta\mathbf{u}+\nabla p &= \rho\mathbf{f} \quad \text{in}\,\Omega\times[0,T] \\ \qquad\qquad\qquad\qquad \text{div}\mathbf{u} &= 0 \quad \text{in}\,\Omega\times[0,T] \end{cases} \tag{1.1}$$

where:

- $\mathbf{u}=\mathbf{u}(\mathbf{x},t)$ is velocity of a fluid particle occupying the position $\mathbf{x}$ at the time t and $p=p(\mathbf{x},t)$ is its pressure;
- ρ and μ are the constant density and *dynamic* viscosity, respectively;
- $\mathbf{f}=\mathbf{f}(\mathbf{x},t)$ is an internal force exerted on the fluid per unit mass;
- and $(\mathbf{u}\cdot\nabla)\mathbf{u}=\Sigma_{i=1}^{d}u_i\partial_i\mathbf{u}$, u_i is the i^{th}-component of velocity field $\mathbf{u}$, $\partial_i\mathbf{u}$ is the partial derivative according to the variable x_i; $\text{div}\mathbf{u}=\Sigma_{i=1}^{d}\partial_i u_i$.

The first equation of (1.1) expresses the conservation of motion of the fluid while the second one expresses the conservation of mass for an incompressible fluid. By posing $\nu=\frac{\mu}{\rho}; p=\frac{p}{\rho}$ we obtain the formula of the incompressible Navier-Stokes equations as follows:

$$\begin{cases} \left(\dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) - \nu\Delta\mathbf{u} + \nabla p &= \mathbf{f} \quad \text{in}\,\Omega\times[0,T] \\ \qquad\qquad\qquad\qquad \text{div}\mathbf{u} &= 0 \quad \text{in}\,\Omega\times[0,T] \end{cases} \tag{1.2}$$

In these equations, ν represents the constant *kinematic* viscosity of the fluid. By giving L a characteristic length of the fluid, we can define the Reynolds number Re as:

$$Re = \frac{|\mathbf{u}|.L}{\nu} \tag{1.3}$$

This number represents the ratio between the inertial force and the viscous force. In case of large value of ν, the viscous force dominates and the convection term in the Navier-Stokes equations can be neglected. This leads to consider the unsteady Stokes equations:

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} - \nu\Delta\mathbf{u} + \nabla p &= \mathbf{f} \quad \text{in}\,\Omega\times[0,T] \\ \qquad\qquad \text{div}\mathbf{u} &= 0 \quad \text{in}\,\Omega\times[0,T] \end{cases} \tag{1.4}$$

In order to ensure the well-posedness of the problem, the equations (1.2) need to be supplemented with a set of adequate boundary conditions (see Section 1.3). They are also endowed with an initial condition: a divergence-free velocity field $\mathbf{u}_0(\mathbf{x})$ is specified over the domain Ω at time $t = 0$, i.e. $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$, $\text{div}\mathbf{u}_0 = 0$.

1.2 The method of characteristics

In this section we focus on the time discretization of the Navier-Stokes equations by the method of characteristics. This method, also known as the Lagrange-Galerkin method was introduced by Benqué [BIKL80]. It provides a time discretization without the necessity of restrictive stability conditions as in a classical discretization by an explicit scheme.

1.2.1 The Lagrange-Galerkin formulation

We are now introducing the Lagrange-Galerkin formulation to recast the convective operator of the Navier-Stokes equations. Let $\mathbf{X}(\mathbf{x}, s; t)$ be a characteristic curve associated with the velocity field $\mathbf{u}$, solution of the ordinary differential equations:

$$\begin{cases} \dfrac{d\mathbf{X}(\mathbf{x}, s; t)}{dt} &= \mathbf{u}(\mathbf{X}(\mathbf{x}, s; t), t) \\ \mathbf{X}(\mathbf{x}, s; s) &= \mathbf{x} \end{cases} \tag{1.5}$$

where the point $\mathbf{X}(\mathbf{x}, s; t)$ denotes the position at time t of a fluid particle located at position $\mathbf{x}$ at the time s.

We write the derivative of the velocity field $\mathbf{u}$ along the characteristic lines as:

$$\begin{aligned}\frac{d\mathbf{u}(\mathbf{X}(\mathbf{x}, s; t), t)}{dt} &= \frac{\partial \mathbf{u}}{\partial t} + \nabla \mathbf{u} \frac{d(\mathbf{X}(\mathbf{x}, s; t), t)}{dt} \\ &= \frac{\partial \mathbf{u}}{\partial t} + \nabla \mathbf{u}.\mathbf{u}(\mathbf{X}(\mathbf{x}, s; t), t) \\ &= \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}.\nabla)\mathbf{u} \end{aligned} \tag{1.6}$$

Hence, the operator $\frac{\partial}{\partial t} + \mathbf{u}.\nabla$ of Navier-Stokes equations can be turned into a total derivative $\frac{d}{dt}$ (also called the Lagrange derivative). The first equation of (1.2) is recast into the following form:

$$\frac{d\mathbf{u}(\mathbf{X}(\mathbf{x}, s; t), t)}{dt} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \tag{1.7}$$

It can be observed that the treatment of the nonlinear convection term is therefore reduced to a problem of searching the characteristic foot $\mathbf{X}(\mathbf{x}, s; t)$, i.e the position of the particle at the previous time. This approach allows us to avoid theoretically the constraint of CFL (Courant-Friedrichs-Levy) on the time step (for example in [PLT92] the good choice for time step is proposed as $\Delta t \approx 1.5h/|u|$). Moreover, it has been shown in [Pir82] that if the characteristic trajectory is calculated exactly and the second member is also integrated exactly, the resulting scheme is *unconditionally stable*.

1.2.2 The discretization in time

We assume that the interval $[0, T]$ is divided into a number of subintervals Δt and we denote $t^n = n\Delta t$. For the numerical solution, we only compute $\mathbf{u}(\mathbf{x}, t^n)$ for all n, so from now on, we will denote $\mathbf{u}(\mathbf{x}, t^n)$ by $\mathbf{u}^n(\mathbf{x})$.

Using the following approximation of the total derivative along the characteristic curves:

$$\frac{d\mathbf{u}(\mathbf{X}(\mathbf{x}, t^n; t), t)}{dt} \simeq \frac{\mathbf{u}(\mathbf{x}, t^n) - \mathbf{u}(\mathbf{X}^{n-1}(\mathbf{x}), t^{n-1})}{\Delta t}$$

where $\mathbf{x}$ is the position of the particle at current time t^n and $\mathbf{X}^{n-1}(\mathbf{x})$ is its foot at t^{n-1} on the characteristic curve (i.e $\mathbf{X}^{n-1}(\mathbf{x})$ stands for $\mathbf{X}(\mathbf{x}, t^n; t^{n-1})$), we find the discretization formulation of the Navier-Stokes equations (1.2):

$$\begin{cases} \dfrac{\mathbf{u}^n(\mathbf{x}) - \mathbf{u}^{n-1}(\mathbf{X}^{n-1}(\mathbf{x}))}{\Delta t} - \nu \Delta \mathbf{u}^n(\mathbf{x}) + \nabla p^n(\mathbf{x}) &= \mathbf{f}^n \\ \qquad\qquad\qquad\qquad\qquad\qquad \mathrm{div}\mathbf{u}^n(\mathbf{x}) &= 0 \end{cases}$$

which are equivalent to:

$$\begin{cases} \dfrac{\mathbf{u}^n(\mathbf{x})}{\Delta t} - \nu\Delta\mathbf{u}^n(\mathbf{x}) + \nabla p^n(\mathbf{x}) & = \mathbf{f}^n + \dfrac{\mathbf{u}^{n-1}(\mathbf{X}^{n-1}(\mathbf{x}))}{\Delta t} \\ \text{div}\mathbf{u}^n(\mathbf{x}) & = 0 \end{cases} \tag{1.8}$$

where $\mathbf{u}^{n-1}(\mathbf{X}^{n-1}(\mathbf{x}))$ represents the velocity at the point $\mathbf{X}^{n-1}(\mathbf{x})$ at the time t^{n-1}.
Hence, at each time step, the Navier-Stokes problem reduces to an unsteady Stokes problem plus a transport of the previous solution along the characteristic lines under a velocity field $\mathbf{u}$.
At each time steps, the discretization in time of the Navier-Stokes equations is carried on in two steps :

i. Resolution of the problem (1.5) in the sub-interval $[t^{n-1}, t^n]$.

ii. Resolution of the generalized Stokes equations defined by:

$$\begin{cases} k\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p & = \mathbf{g} \\ \text{div}\mathbf{u} & = 0 \end{cases} \tag{1.9}$$

where k is denoting $\frac{1}{\Delta t}$ and $\mathbf{g}$ is denoting all terms in the right-hand side and is updated in time according to formulation (1.8).

Remark 1.

1. *The advantage of the discretization of Navier-Stokes equations using this scheme lies in the fact that at each iteration, we only need to update the right-hand side, i.e the left-hand side remains constant during the resolution.*

2. *Concerning the problem of the discretisation of the characteristic lines, or more precisely the approximation of $\mathbf{X}^{n-1}(\mathbf{x})$, we forecast some difficulties:*

   i. *The first problem is how to approximate the velocity field $\mathbf{u}(t)$ in the sub-interval time $[t^{n-1}, t^n]$. In solving Navier-Stokes equations, $\mathbf{u}^n$ is unknown, and thus $\mathbf{X}^{n-1}(\mathbf{x})$ is associated with $\mathbf{u}^{n-1}$.*

   ii. *The second difficulty is that the characteristic line may cross some elements of the domain or even go out of the computational domain, hence it is necessary to locate the element which contains the last point of the advection or to find the cases in which the latter hits a boundary edge. In numerical practice, the backward tracking the characteristic points are performed by Euler's scheme and Runge-Kutta 4. These scheme will be detailed in Section 2.3 where the method of characteristics is again applied to resolve the advection equation of the interface. See figure 1.1 for illustrations the travel of given point $\mathbf{x}$ in domain Ω with substep time $\delta t \ll \Delta t$.*

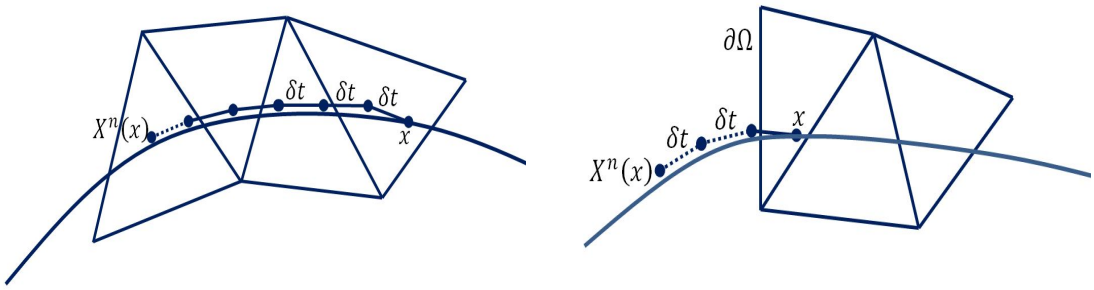


Figure 1.1: *(left) travel characteristic point with time δt and (right) characteristic point is go out to domain.*

We know that in practice the exact calculation of the characteristic curves or the second member is impossible, it must be done using numerical integration formulas. Unfortunately, the numerical approaches prevent the unconditional stability of the Lagrange-Galerkin scheme, either of order 1 or order 2. For more exhaustive studies about the stability of Lagrange-Galerkin method, we refer to [MPS88] or [Fou02] and references therein. We turn to the numerical practice where the *stopping condition* is performed when the residual of solution is small enough, one say that the solution reaches a *steady-state* (of course if it exists). More precisely, given a computational mesh, we construct a global scheme for solving Navier-Stokes equations:

Algorithm 1: Numerical scheme for solving Navier-Stokes equation over $[0, T]$

n=0, initializations $(\mathbf{u}^0, p^0)$;

Given time step Δt;

$nmax = [T/\Delta t]$;

while *((*$n \leq nmax$*) and (*$res > \varepsilon$*))* **do**

Integrate $\mathbf{X}^{n-1}(\mathbf{x})$ (equation (1.5)) for each node $\mathbf{x}$ of mesh;

Compute $\mathbf{u}^{n-1}(\mathbf{X}^{n-1}(\mathbf{x}))$ at each node $\mathbf{x}$ of mesh;

Update the right hand side $\mathbf{g}^n := \mathbf{f}^n + \mathbf{u}^{n-1}(\mathbf{X}^{n-1}(\mathbf{x}))$;

Solve Stokes equations (1.9) to obtain $(\mathbf{u}^n, p^n)$;

Compute residual at n^{th}-step: $res := \|\mathbf{u}^n - \mathbf{u}^{n-1}\|_0$;

n:=n+1;

end

where ε is a given small value introduced to define the steady-state solution in each test case.

1.3 The variational formulation

As we discussed in the previous section, at each time step, we have to solve an unsteady Stokes problem (1.9). The numerical resolution of this problem by a finite element method evidently involves a variational formulation.

Let us firstly introduce the following Sobolev spaces which will be used hereafter:

$$H_0^1(\Omega)^d = \{\mathbf{v} \in H^1(\Omega)^d : \mathbf{v}|_\Sigma = 0\}$$
$$H_{\Sigma_D}^1(\Omega)^d = \{\mathbf{v} \in H^1(\Omega)^d : \mathbf{v}|_{\Sigma_D} = 0\}$$
$$L_0^2(\Omega) = \{q \in L^2(\Omega) : \int_\Omega q = 0\}$$

1.3.1 Homogeneous Dirichlet boundary conditions

Denoting V and M for the functional spaces of velocity and pressure, respectively. We consider the variational formulation for the general Stokes problem in case of homogeneous Dirichlet boundary condition, i.e $\mathbf{u} = 0 \,\text{on}\, \Sigma$. In this case, V will be chosen to coincide with $H_0^1(\Omega)^d$ and we take $M = L_0^2(\Omega)$ for the pressure.

Given $\mathbf{v} \in V$ and using Green's formula we have:

$$-\int_\Omega \nu \Delta \mathbf{u}\mathbf{v} dx = \int_\Omega \nu \nabla \mathbf{u} : \nabla \mathbf{v} dx - \int_{\partial\Omega} \nu \nabla \mathbf{u}\mathbf{n}.\mathbf{v} ds$$
$$\int_\Omega \nabla p.\mathbf{v} dx = -\int_\Omega p \text{div}\mathbf{v} dx + \int_{\partial\Omega} p\mathbf{n}.\mathbf{v} ds$$

Using these equalities and taking the inner product in $L^2(\Omega)^d$ of the first equation of systems (1.9), we come to the following equation:

$$k\int_\Omega \mathbf{u}.\mathbf{v} dx + \int_\Omega \nu \nabla \mathbf{u} : \nabla \mathbf{v} dx - \int_\Omega p \text{div}\mathbf{v} dx = \int_\Omega \mathbf{g}.\mathbf{v} dx + \int_{\partial\Omega} (\nu \nabla \mathbf{u} - pI_d)\mathbf{n}.\mathbf{v} ds \tag{1.10}$$

Since $\mathbf{v} = 0$ on $\partial\Omega$, we have:

$$k\int_\Omega \mathbf{u}.\mathbf{v} dx + \int_\Omega \nu \nabla \mathbf{u} : \nabla \mathbf{v} dx - \int_\Omega p \text{div}\mathbf{v} dx = \int_\Omega \mathbf{g}.\mathbf{v} dx$$

Choose the test function $q \in M$, taking the inner product in $L^2(\Omega)$ of the second equation of system (1.9), we obtain finally the variational formulation of the homogeneous problem: Given the function $\mathbf{g} \in H^{-1}(\Omega)^d$, constant k and viscosity ν; find $\mathbf{u} \in V$ and $p \in M$ solution of:

$$\begin{cases} k\int_\Omega \mathbf{u}.\mathbf{v} dx + \int_\Omega \nu \nabla \mathbf{u} : \nabla \mathbf{v} dx - \int_\Omega p \text{div}\mathbf{v} dx = \int_\Omega \mathbf{g}.\mathbf{v} dx \quad \forall \mathbf{v} \in V \\ \int_\Omega q \text{div}\mathbf{u} dx = 0 \quad \forall q \in M \end{cases}$$

This problem can be written in the equivalent weak formulation: Find $(\mathbf{u}, p) \in (V \times M)$ such that:

$$\begin{cases} \forall \mathbf{v} \in V \quad a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = l(\mathbf{v}) \\ \forall q \in M \quad b(\mathbf{u}, q) \qquad\qquad = 0 \end{cases} \tag{1.11}$$

where $l(\cdot)$ is the continuous linear form defined on V:

$$l(\mathbf{v}) = \int_\Omega \mathbf{g}.\mathbf{v} dx$$

and $a(\mathbf{u}, \mathbf{v}), b(\mathbf{v}, p)$ are the continuous bilinear forms defined on $V \times V$ and $V \times M$ respectively as:

$$\begin{cases} a(\mathbf{u}, \mathbf{v}) & = k \int_\Omega \mathbf{u}.\mathbf{v} dx + \int_\Omega \nu \nabla \mathbf{u} : \nabla \mathbf{v} dx \\ b(\mathbf{v}, p) & = -\int_\Omega p \mathrm{div} \mathbf{v} dx \end{cases} \tag{1.12}$$

The existence and the uniqueness of a solution for the weak formulation of the generalized Stokes problem has been proven (see [GR86] or [EG04]). This proof involves: i. the ellipticity of the form $a(.,.)$ results from the Poincare-Friedrichs inequality; ii. the compatibility of the spaces of velocity and pressure results the satisfying Babuska-Brezzi condition, also called *inf-sup condition* on the form $b(.,.)$, i.e. existing a positive constant C such that:

$$\inf_{q \in M} \sup_{\mathbf{v} \in V} \frac{b(\mathbf{v}, q)}{\|\mathbf{v}\|_1 \|q\|_0} \geq C > 0 \tag{1.13}$$

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 $H^1(\Omega)$, $L^2(\Omega)$ respectively.

1.3.2 Case of general boundary conditions

Let us now consider the variational formulation in the case of more complicated boundary conditions. We limit ourselves to consider somme classical boundary conditions (Dirichlet, Newmann and Slip) which are used in most numerical tests. Suppose that the boundary Σ of Ω is partitioned into a finite number of components corresponding to different types of boundary condition: $\Sigma = \Sigma_D \cup \Sigma_N \cup \Sigma_S$.

i. Dirichlet conditions: $\mathbf{u} = \mathbf{u}_D$ on Σ_D.

ii. Newmann conditions: $\sigma \mathbf{n} = \varphi_N$ on Σ_N.

iii. Slip conditions: $\mathbf{u}.\mathbf{n} = 0, \alpha \mathbf{u}.\tau + \tau.\sigma \mathbf{n} = 0$ on Σ_S.

where σ is defined as: $\sigma = \nu\nabla\mathbf{u} - pI_d$; I_d is the identity matrix and $\tau, \mathbf{n}$ are the unit tangent vector and unit outward normal vector to Σ, respectively; $\alpha \geq 0$ is a constant of friction coefficient. In the case of Dirichlet boundary condition, V is usually chosen in such a way that the test function vanishes on Σ_D:

$$V = {H^1_{\Sigma_D}}^d = \{\mathbf{v} \in H^1(\Omega)^d : \mathbf{v}|_{\Sigma_D} = \mathbf{0}\}$$

Hence,

$$\int_{\Sigma_D} \sigma\mathbf{n}.\mathbf{v}ds = 0$$

It can be observed that the condition Dirichlet affects only on the selection of space functions for the velocity, it doesn't affect the variational formulation, so it is called essential boundary condition. When the non-homogeneous Dirichlet is imposed, i.e $\mathbf{u_D} \neq 0$, $\mathbf{u_D}$ is supposed to be a sufficiently smooth on Σ_D such that there exists a lifting function $\bar{\mathbf{u_D}}$ with divergence-free property. Using the expression: $\mathbf{u} = \bar{\mathbf{u_D}} + \bar{\mathbf{u}}$ where $\bar{\mathbf{u}} \in V$. The weak formulation becomes: Find $(\bar{\mathbf{u}}, p) \in (V \times M)$ such that:

$$\begin{cases} \forall \mathbf{v} \in V \quad a(\bar{\mathbf{u}}, \mathbf{v}) + b(\mathbf{v}, p) = l(\mathbf{v}) - a(\bar{\mathbf{u_D}}, \mathbf{v}) \\ \forall q \in M \quad b(\bar{\mathbf{u}}, q) \qquad\qquad\quad = 0 \end{cases} \tag{1.14}$$

In practice the Dirichlet condition is simply imposed by multiplying a very large value on the right-hand side and the diagonal matrix of matrix formulation (will be presented in the next section). Notice that V will coincide with $(H^1_0(\Omega))^d$ if $\Sigma_D \equiv \Sigma$ that means only boundary condition of Dirichlet type is imposed. In this case, the pressure only appears through its gradient, so, if $(\mathbf{u}^*, p^*)$ is a solution of Navier-Stokes equation then for any constant c, $(\mathbf{u}^*, p^* + c)$ will be also a solution. We chose

$$M = L^2_0(\Omega) = \{p \in L^2(\Omega) : \int_\Omega p = 0\}$$

to avoid this indeterminacy.

The term corresponding to a Newmann condition can be written as:

$$\int_{\Sigma_N} \sigma\mathbf{n}.\mathbf{v}ds = \int_{\Sigma_N} \varphi_N.\mathbf{v}ds$$

So, Newmann condition needs only adding this boundary term in the right-hand side of the variational formulation. This is one kind of natural boundary condition. In the case $\varphi_N = 0$, the subset Σ_N is called a free-out flow part, this integral term disappears from variational formulation. Notice that when Σ_N is not empty, this means that we have condition for pressure on Σ_N, the problem of pressure indeterminacy previously in the case only Dirichlet condition, no longer exists, so we can take $M = L^2(\Omega)$.

In the case of slip boundary condition, the test function $\mathbf{v}$ is chosen in V and satisfies $\mathbf{v}.\mathbf{n}|_{\Sigma_S} = 0$. The term of this condition can be written as:

$$\begin{aligned}\int_{\Sigma_S} \sigma\mathbf{n}.\mathbf{v}ds &= \int_{\Sigma_S} (\mathbf{n}.\sigma\mathbf{n}).(\mathbf{v}.\mathbf{n})ds + \int_{\Sigma_S} (\tau.\sigma\mathbf{n}).(\mathbf{v}.\tau)ds \\ &= \int_{\Sigma_S} -\alpha(\mathbf{u}.\tau).(\mathbf{v}.\tau)ds\end{aligned}$$

We impose the slip condition by adding the integral $\int_{\Sigma_S} \alpha(\mathbf{u}.\tau).(\mathbf{v}.\tau)ds$ in the bilinear form $a(\mathbf{u},\mathbf{v})$ of the variational formulation and the condition $\mathbf{u}.\mathbf{n}|_{\Sigma_S} = 0$ can be implemented as a Dirichlet condition on the components of $\mathbf{u}$ for the simply domain (the domain with horizontal or vertical boundary in 2D and rectangular or square boundary in 3D). In case $\alpha = 0$, this integral term vanishes and we can implement the condition $\mathbf{u}.\mathbf{n}|_{\Sigma_S} = 0$ generally by adding $\int_{\Sigma_S} A * (\mathbf{u}.\mathbf{n}).(\mathbf{v}.\mathbf{n})ds$ in the left-hand side and multiplying the components on Σ_S of right-hand side with A where A is very large number (about 10^6).

1.4 The spatial discretization

By using a Galerkin finite element approximation, the discrete problem corresponding to the formulation (1.11) writes as follows: Find $(\mathbf{u}_h, p_h) \in V_h \times M_h$ s.t

$$\begin{cases} \forall \mathbf{v}_h \in V_h & a_h(\mathbf{u}_h, \mathbf{v}_h) + b_h(\mathbf{v}_h, p_h) = l_h(\mathbf{v}_h) \\ \forall q_h \in M_h & b(\mathbf{u}_h, q_h) = 0 \end{cases} \tag{1.15}$$

with $V_h \subset V$ and $M_h \subset M$ represent two families of finite dimensional subspaces and $a_h(\mathbf{u}_h, \mathbf{v}_h)$, $b_h(\mathbf{v}_h, p_h)$, $l_h(\mathbf{v}_h)$ are defined on $V_h \times V_h$, $V_h \times M_h$ and V_h respectively as follows:

$$\begin{cases} a_h(\mathbf{u}_h, \mathbf{v}_h) = \sum\limits_{K\in T_h} k \int_K \mathbf{u}_h.\mathbf{v}_h dx + \sum\limits_{K\in T_h} \int_K \nu\nabla\mathbf{u}_h : \nabla\mathbf{v}_h dx \\ b_h(\mathbf{v}_h, p_h) = \sum\limits_{K\in T_h} \int_K -p_h \mathrm{div}\mathbf{v}_h \\ l_h(\mathbf{v}_h) = \sum\limits_{K\in T_h} \int_K \mathbf{g}_h.\mathbf{v}_h dx \end{cases} \tag{1.16}$$

It is well-known that the approximate problem also requires the discrete compatibility condition, meaning that the discrete spaces of velocity needs to be "rich" enough comparing with the one of pressure, i.e there exists a positive constant C_h s.t:

$$\inf_{q_h\in M_h} \sup_{\mathbf{v}_h\in V_h} \frac{b_h(\mathbf{v}_h, p_h)}{\|\mathbf{v}_h\|_1 \|q_h\|_0} \geq C_h > 0 \tag{1.17}$$

which is also known as the *discrete inf-sup condition.*

There are many possible choices for the discrete spaces which satisfy this condition. A description of some compatible couple of spaces for finite element spaces can be found in [Pir89], [BBD$^+$06] or [Qua09]. We have decided to retain mini elements ($\mathbb{P}_1$bubble/$\mathbb{P}_1$) and Taylor-Hood elements ($\mathbb{P}2/\mathbb{P}1$) have been implemented. We briefly recall here these elements.

The *mini elements*: All polynomial approximation of same order for both velocity and pressure

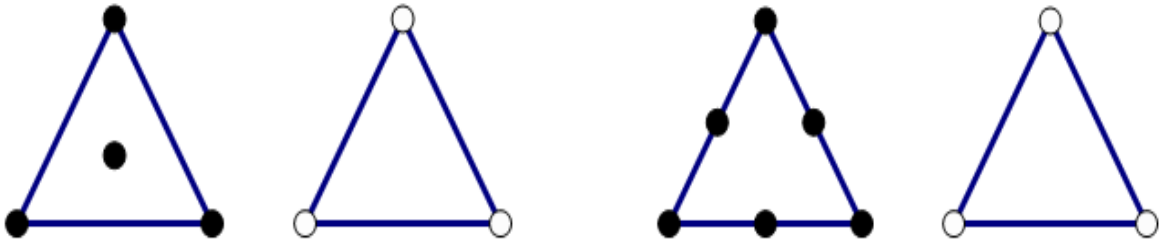


Figure 1.2: *Degree of freedom: (a) mini element ($\mathbb{P}_1$bubble/$\mathbb{P}_1$); (b)Taylor-Hood element ($\mathbb{P}2/\mathbb{P}1$)*

are ill-posed. Suppose that T_h is composed of elements $K = (P_1, ..., P_{d+1})$, from the approximation of piecewise linear $\mathbb{P}1/\mathbb{P}1$ finite elements, in order to satisfy the *inf-sup condition* (1.17), the spaces of velocity can be enriched by adding one degree of freedom b_K on each elements where b_K is so-called *bubble function* defined as: $b_K = (d+1)^{d+1} \prod_{i=1}^{d+1} \varphi_i$ with φ_i is $\mathbb{P}_1$ Lagrange basic function corresponds to vertex P_i of elements K. We have following discretizations of the spaces of velocity and pressure in this case (see figure 1.2 (a)):

$$V_h = \{\mathbf{v}_h \in (\mathcal{C}^0(\bar{\Omega}))^d : \mathbf{v}_h|_K \in (\mathbb{P}_1(K) \oplus b_K)^d \, \forall K \in T_h\} \cap V$$
$$M_h = \{q_h \in \mathcal{C}^0(\bar{\Omega}) : q_h|_K \in \mathbb{P}_1(K) \forall K \in T_h\} \cap M$$

The *Taylor-Hood elements*: Generally, this choice is to use piecewise polynomials of degree $k \geq 2$ for the velocity space V_h and of $k-1$ for pressure space M_h. In particular, we use piecewise quadratic $\mathbb{P}_2$ finite elements for each velocity component, and piecewise linear $\mathbb{P}_1$ finite elements for pressure. This is the lowest degree representative of the family of so-called Taylor-Hood elements $\mathbb{P}_k/\mathbb{P}_{k-1}, k \geq 2$ (continuous velocities and continuous pressure), that are inf-sup stable (see figure 1.2 (b)):

$$V_h = \{\mathbf{v}_h \in (\mathcal{C}^0(\bar{\Omega}))^d : \mathbf{v}_h|_K \in (\mathbb{P}_2(K))^d \, \forall K \in T_h\} \cap V$$
$$M_h = \{q_h \in \mathcal{C}^0(\bar{\Omega}) : q_h|_K \in \mathbb{P}_1(K) \forall K \in T_h\} \cap M$$

See figure 1.2 for ilustrations of these elements in two dimensions with the denoting of symbol $\bullet$ for degrees of freedom of the velocity and symbol $\circ$ for ones of the pressure.

1.5 The discrete linear systems

1.5.1 Construction of the matrix formulation

Let $\{\varphi_i\}_{i=1,...,nv}$ and $\{\phi_j\}_{j=1,...,np}$ are basis functions of V_h and M_h, respectively with $nv = \dim V_h$ and $np = \dim M_h$. Given $\mathbf{u}_h \in V_h$, $p_h \in M_h$, we have the following decompositions:

$$\begin{aligned} \mathbf{u_h} &= \Sigma_{i=1}^{nv} u_i \varphi_i \\ p_h &= \Sigma_{j=1}^{np} p_j \phi_j \end{aligned} \tag{1.18}$$

We can introduce the following matrix coefficients:

$$\begin{aligned} A_{i,j} &= a_h(\varphi_i, \varphi_j) \\ B_{i,j} &= b_h(\varphi_i, \phi_j) \\ G_i &= l_h(\varphi_i) \end{aligned}$$

where the matrices A, B correspond to the bilinear forms a_h, b_h, respectively and the vector G corresponds to the linear form l_h according to expression (1.16). Denoting the vectors $U = (u_i)_{i=1,...,nv}$ and $P = (p_j)_{j=1,...,np}$, the problem (1.15) is equivalent to solving the square linear system:

$$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} G \\ 0 \end{pmatrix} \tag{1.19}$$

The system (1.19) is sparse, symmetric but not definite and its size is $\dim V_h + \dim M_h$.

1.5.2 Matrix assembly

In practice, the assembling matrices A, B and G in section 1.5.1 are accomplished by following the mesh, or in other words, from programming point of view, by using a loop on the indices of mesh elements. All the contributions on each element K will be computed and stored in the local matrices denoted by $[Ae]_K, [Be]_K, [Ge]_K$. Matrices A, B, G are initialized by zero and add them up to the appropriate entries by finding index global corresponding to index local of $[Ae]_K, [Be]_K, [Ge]_K$. See algorithm 2 for example of the assembly matrix A.

We present here the contributions of these local matrices in two dimensions. In three dimensions, the procedure is almost identical.

The local matrix of matrix A

Recall the bilinear form of $a(\mathbf{u}_h, \mathbf{v}_h)$:

$$a_h(\mathbf{u}_h, \mathbf{v}_h) = \sum_{K \in T_h} k \int_K \mathbf{u}_h . \mathbf{v}_h dx + \sum_{K \in T_h} \int_K \nu \nabla \mathbf{u}_h : \nabla \mathbf{v}_h dx$$

We have the bilinear form $a_h(\mathbf{u_h}, \mathbf{v_h})$ on each element K of mesh as follows:

$$a_h(\mathbf{u}_h, \mathbf{v}_h)|_K = k \int_K \mathbf{u}_h . \mathbf{v}_h dx + \int_K \nu \nabla \mathbf{u}_h : \nabla \mathbf{v}_h dx \tag{1.20}$$

Using the matrical notations in the transformation of bilinear form $a_h(\mathbf{u_h}, \mathbf{v_h})$:

$$\begin{aligned} k \int_K \mathbf{u}_h . \mathbf{v}_h dx &= k \int_K \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}^t \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} dx \\ &= k \int_K \begin{pmatrix} v_1|_K \\ v_2|_K \end{pmatrix}^t \begin{pmatrix} \varphi|_K & 0 \\ 0 & \varphi|_K \end{pmatrix}^t \begin{pmatrix} \varphi|_K & 0 \\ 0 & \varphi|_K \end{pmatrix} \begin{pmatrix} u_1|_K \\ u_2|_K \end{pmatrix} dx \end{aligned} \tag{1.21}$$

$$\begin{aligned} \int_K \nu \nabla \mathbf{u}_h : \nabla \mathbf{v}_h dx &= \int_K \begin{pmatrix} \partial_1 v_1 \\ \partial_2 v_1 \\ \partial_1 v_2 \\ \partial_2 v_2 \end{pmatrix}^t \begin{pmatrix} \nu & 0 & 0 & 0 \\ 0 & \nu & 0 & 0 \\ 0 & 0 & \nu & 0 \\ 0 & 0 & 0 & \nu \end{pmatrix} \begin{pmatrix} \partial_1 u_1 \\ \partial_2 u_1 \\ \partial_1 u_2 \\ \partial_2 u_2 \end{pmatrix} dx \\ &= \int_K \begin{pmatrix} v_1|_K \\ v_2|_K \end{pmatrix}^t \begin{pmatrix} D\varphi|_K & 0 \\ 0 & D\varphi|_K \end{pmatrix}^t \begin{pmatrix} \nu & 0 & 0 & 0 \\ 0 & \nu & 0 & 0 \\ 0 & 0 & \nu & 0 \\ 0 & 0 & 0 & \nu \end{pmatrix} \begin{pmatrix} D\varphi|_K & 0 \\ 0 & D\varphi|_K \end{pmatrix} \begin{pmatrix} u_1|_K \\ u_2|_K \end{pmatrix} dx \end{aligned} \tag{1.22}$$

where: $[u_i|_K]^t = (u_i^{\varphi_1}, \ldots, u_i^{\varphi_{nev}})(i = 1, 2)$; $[\varphi|_K] = (\varphi_1, \ldots, \varphi_{nev})$; $[D\varphi|_K] = \begin{pmatrix} \partial_1\varphi_1 \ldots \partial_1\varphi_{nev} \\ \partial_2\varphi_1 \ldots \partial_2\varphi_{nev} \end{pmatrix}$ with $\varphi_1, \ldots, \varphi_{nev}$ are Lagrange basis functions of V_h on element K, nev is the number of degree of freedom of velocity on each element and $u_i^{\varphi_1}, \ldots, u_i^{\varphi_{nev}}$ are nodal values of velocity components correspond to $\varphi_1, \ldots, \varphi_{nev}$; respectively.

Hence, the contributions associated with element K of global matrix A will be stored in the local

matrix $[Ae]_K$ defined by:

$$[Ae]_K = k\int_K \begin{pmatrix} \varphi|_K & 0 \\ 0 & \varphi|_K \end{pmatrix}^t \begin{pmatrix} \varphi|_K & 0 \\ 0 & \varphi|_K \end{pmatrix} dx + \int_K \begin{pmatrix} D\varphi|_K & 0 \\ 0 & D\varphi|_K \end{pmatrix}^t \begin{pmatrix} \nu & 0 & 0 & 0 \\ 0 & \nu & 0 & 0 \\ 0 & 0 & \nu & 0 \\ 0 & 0 & 0 & \nu \end{pmatrix} \begin{pmatrix} D\varphi|_K & 0 \\ 0 & D\varphi|_K \end{pmatrix} dx \tag{1.23}$$

This integral is computed based on the quadrature formulation, we have:

$$[Ae]_K = k\sum_{p=1}^{n_p} \begin{pmatrix} \varphi|_K & 0 \\ 0 & \varphi|_K \end{pmatrix}^t \begin{pmatrix} \varphi|_K & 0 \\ 0 & \varphi|_K \end{pmatrix} |_{x_p} + \sum_{p=1}^{n_p} \begin{pmatrix} D\varphi|_K & 0 \\ 0 & D\varphi|_K \end{pmatrix}^t \begin{pmatrix} \nu & 0 & 0 & 0 \\ 0 & \nu & 0 & 0 \\ 0 & 0 & \nu & 0 \\ 0 & 0 & 0 & \nu \end{pmatrix} \begin{pmatrix} D\varphi|_K & 0 \\ 0 & D\varphi|_K \end{pmatrix} |_{x_p} \tag{1.24}$$

where x_p are Gaussian points and n_p is the number of Gaussian point in element K. It can be seen that the size of local matrix $[Ae]_K$ is $2nev \times 2nev$. ($nev = 4$ in case of $\mathbb{P}1$-bubble and $nev = 6$ in case of $\mathbb{P}2$)

The local matrix of matrix B

Similarly, recall the bilinear form of $b_h(\mathbf{v}_h, p_h)$:

$$b_h(\mathbf{v}_h, p_h) = \sum_{K\in T_h} \int_K -p_h \mathrm{div}\mathbf{v}_h dx \tag{1.25}$$

and using the matrical notations we have:

$$\begin{aligned} b_h(\mathbf{u}_h, q_h)|_K &= -\int_K q_h \mathrm{div}\mathbf{u}_h dx \\ &= \int_K \begin{pmatrix} -q \\ -q \end{pmatrix}^t \begin{pmatrix} \partial_1 u_1 \\ \partial_2 u_2 \end{pmatrix} dx \\ &= \int_K \begin{pmatrix} q|_K \\ q|_K \end{pmatrix}^t \begin{pmatrix} -\phi|_K & 0 \\ 0 & -\phi|_K \end{pmatrix}^t \begin{pmatrix} \partial_1\varphi|_K & 0 \\ 0 & \partial_2\varphi|_K \end{pmatrix} \begin{pmatrix} u_1|_K \\ u_2|_K \end{pmatrix} dx \end{aligned} \tag{1.26}$$

where: $[q|_K]^t = (q^{\phi_1}, \ldots, q^{\phi_{nep}})$, $[\phi|_K] = (\phi_1, \ldots, \phi_{nep})$, $[\partial_1\varphi|_K] = [\partial_1\varphi_1, \ldots, \partial_1\varphi_{nev}]$, $[\partial_2\varphi|_K] = [\partial_2\varphi_1, \ldots, \partial_2\varphi_{nev}]$ with $\phi_1, \ldots, \phi_{nep}$ are Lagrange basis functions of M_h on element K, nep is the

number of degree of freedom of pressure on each mesh element and $q^{\phi_1}, \ldots, q^{\phi_{nep}}$ are nodal values of test function q_h correspond to $\phi_1, \ldots, \phi_{nep}$, respectively.
The local matrix $[Be]_K$ is then determined as follows:

$$[Be]_K = \int_K \begin{pmatrix} -\phi|_K & 0 \\ 0 & -\phi|_K \end{pmatrix}^t \begin{pmatrix} \partial_1\varphi|_K & 0 \\ 0 & \partial_2\varphi|_K \end{pmatrix} dx \tag{1.27}$$

We have $[Be]_K$ is size of $2nep \times 2nev$. This integral is also computed by a quadrature formulation as in the computation of $[Ae]_K$.

The local matrix of matrix G

The last term is linear form l_h:

$$l_h(\mathbf{v}_h) = \sum_{K \in T_h} \int_K \mathbf{g}_h.\mathbf{v}_h dx$$

Hence,

$$\begin{aligned} l_h(\mathbf{v}_h)|_K &= \int_K \mathbf{g}_h.\mathbf{v}_h dx \\ &= \int_K \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}^t \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} dx \\ &= \int_K \begin{pmatrix} v_1|_K \\ v_2|_K \end{pmatrix}^t \begin{pmatrix} \varphi|_K & 0 \\ 0 & \varphi|_K \end{pmatrix}^t \begin{pmatrix} \varphi|_K & 0 \\ 0 & \varphi|_K \end{pmatrix} \begin{pmatrix} g_1|_K \\ g_2|_K \end{pmatrix} dx \end{aligned} \tag{1.28}$$

where: $[q_i|_K]^t = (q_i^{\varphi_1}, \ldots, q_i^{\varphi_{nev}})(i = 1, 2); \quad [\varphi|_K] = (\varphi_1, \ldots, \varphi_{nev})$.
We have the local vector on the right-hand side:

$$[Ge]_K = \int_K \begin{pmatrix} \varphi|_K & 0 \\ 0 & \varphi|_K \end{pmatrix}^t \begin{pmatrix} \varphi|_K & 0 \\ 0 & \varphi|_K \end{pmatrix} \begin{pmatrix} g_1|_K \\ g_2|_K \end{pmatrix} dx \tag{1.29}$$

is size of $2nev$ and computed by using a quadrature formulation.
The overall assembly algorithm of matrix A is given as following:

Algorithm 2: Assembly of matrix A, case 2D

Initialization $A = 0$.

Loop on $k = 1, 2, \ldots, ne$. Let K be the k^{th} element.

Loop on local indices $il = 1, 2, \ldots, 2nev$ of $[Ae]_K$

Find global index ig of matrix A corresponding to local index il of $[Ae]_K$:

$ig := 2 * [pt \to v[il\%nev]] + il/nev$;

(where $pt \to v[il\%nev]$ is global index corresponding to $(il\%nev)^{th}$ point of element K).

Loop on local indices $jl = il, \ldots, 2nev$.

Find global index jg corresponding to local index jl of $[Ae]_K$:

$jg := 2 * [pt \to v[jl\%nev]] + jl/nev$;

(where $pt \to v[jl\%nev]$ is global index corresponding to $(jl\%nev)^{th}$ point of element K).

If $(jg <= ig)$ then $A_{jg,ig} := +Ae_{jl,il}$;

else $A_{ig,jg} := +Ae_{il,jl}$;

The assembly matrices B, G are implemented similarly.

1.5.3 Solving the linear systems

Numerous methods have been proposed to solve the linear system (1.19). A classical alternative is *Uzawa* method [AHU58] which relies on coupling (1.19) in to two subsystems, one for the unknown velocity U and other for pressure P, which leads to solving successively:

$$BA^{-1}B^tP = BA^{-1}G, \qquad AU = G - B^tP. \tag{1.30}$$

Attentively, the so-called *Penalty method* can be used to solving U and P in an equivalent linear system as follows:

$$\begin{pmatrix} A & B^t \\ B & \epsilon I_d \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} G \\ 0 \end{pmatrix} \tag{1.31}$$

where I_d is identity matrix, its size is $\dim M_h$ and ϵ values about 10^{-6} to 10^{-4}.

So, at each time step a generalized Stokes problem leads to solve one or severals linear systems. Our code based upon solving linear system by a Conjugate Gradient (CG) algorithm and a SSOR-preconditionned CG algorithms. These techniques have been applied popularly in solving sparse linear system, see [She94, Saa03, YJ07] for details.

1.6 Numerical examples

We have developed one solver Navier-Stokes in two dimension (2D) and three dimensions (3D) with the following options:

1. The discretisation in time is based on the method of characteristics where the characteristic curves is approximated by Euler's scheme or scheme Runge-Kutta 4.
2. The disretisation in space is programmed the by $\mathbb{P}_1$bubble/$\mathbb{P}_1$ or $\mathbb{P}_2/\mathbb{P}1$.
3. The resolution of linear system is implemented by method of Uzawa or Penalty method.

In order to valid the solver with the previous options, we perform the numerical test of Lid-driven cavity 2D with $Re = 1000$. The detailed results of this test will be investigated in Chapter 4 (Section 4.1) in comparison with the other well-known references. This section aims to compare the convergence of all the scheme in our solver. The first one (the scheme of element $\mathbb{P}1b\mathbb{P}1$, Euler's scheme and Uzawa's method) is chosen as the reference. In case of this Reynolds number, all the experimental and numerical observations showed that the solution reach to a steady-state. With the same stopping condition when the residual between the solutions reaches to 10^{-6}, it has been observed that the solutions are exactly the same for all schemes. The required CPU time of these resolutions to obtain the steady solution are detailed on Table 1.1 in comparison with the reference one.

From the details of the Table 1.1 we see that: the implement of scheme Euler or Runge-Kutta 4

Finite element	Scheme of characteristic	Method	Number of iterations	CPU time
$\mathbb{P}_1b\mathbb{P}_1$	Euler	Uzawa	70	1
$\mathbb{P}_1b\mathbb{P}_1$	Euler	Penalty	76	1,2
$\mathbb{P}_1b\mathbb{P}_1$	Runge Kutta 4	Uzawa	70	1,79
$\mathbb{P}_1b\mathbb{P}_1$	Runge Kutta 4	Penalty	76	2,13
$\mathbb{P}_2\mathbb{P}_1$	Euler	Uzawa	76	4,58
$\mathbb{P}_2\mathbb{P}_1$	Runge Kutta 4	Uzawa	76	7.97

Table 1.1: *Cavity in 2D: Details on CPU time to reach the steady-state with Re = 1000, uniform mesh of 676 vertices, 1250 elements.*

(RK4) does not affect the number of iteration to reach the steady-state, on the contrary, the RK4 increases the CPU time in comparison with Euler's scheme. This increasing is explained by the

reason that in RK4, at each sub-step δt we have to pass 4 intermediate points to calculate one characteristic point while it requires only one in Euler's scheme (see section 2.3.2 for the details).

In the theorical analysis of method of characteristics for the Navier-Stokes equations, the error estimation has been determined of the form $O(h^m + \Delta t^m + h^{m+1}/\Delta t)$ in [Pir82] (a scheme of order $(h + \Delta t + h^2/\Delta t)$ for Crouzeix-Raviant element and a scheme of order $(h^2 + \Delta t^2 + h^3/\Delta t)$ for Taylor-Hood element). It has seen that the error estimate is controlled by either characteristic mesh element $h, \Delta t$ or the relation of h and Δt. An optimal error estimates for the Lagrange-Galerkin mixed finite element approximation of the Navier-Stokes equations was given by Suli [Sul98], see also [Zhi12] and references therein. More exhaustive studies about the error estimations for each finite elements lie far beyond the scope of this thesis. However, by numerical experience we propose here the choice of time step to obtain the best solution with the present scheme. In fact, the choice of time step is very important: with the different time steps we obtain the different behaviour of the steady solutions. For this illustration, we perform the test on the mesh carre7 (see figure 1.4, bottom, right) with different time steps and exhibit the profiles of velocity along horizontal and vertical lines passing the geometric center of cavity in each case. The results are shown on figure 1.3 in comparison with the profiles in two references [UG82] and [ECG05]. It is observed that the best resolution for this mesh is $\Delta t = 0.21s$.

Furthermore, we consider the case test cavity 2D with 4 different meshes: a regular triangulation

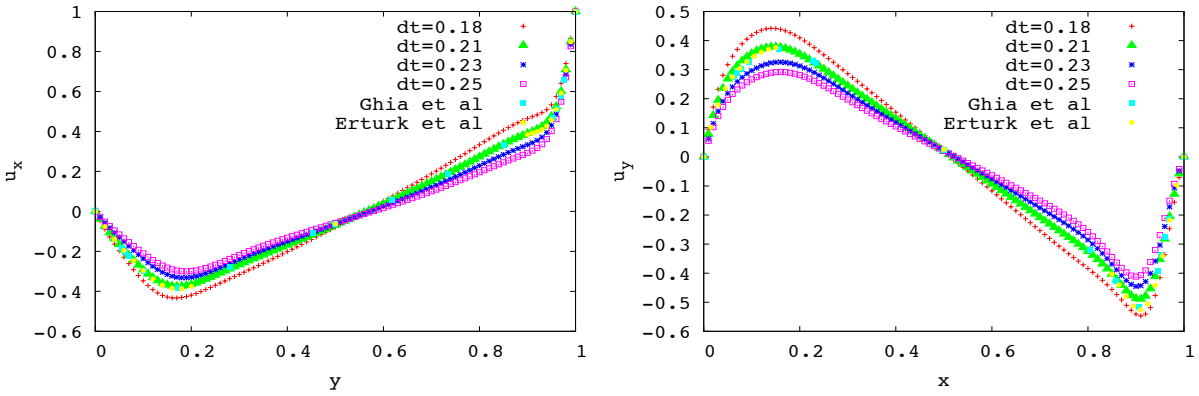
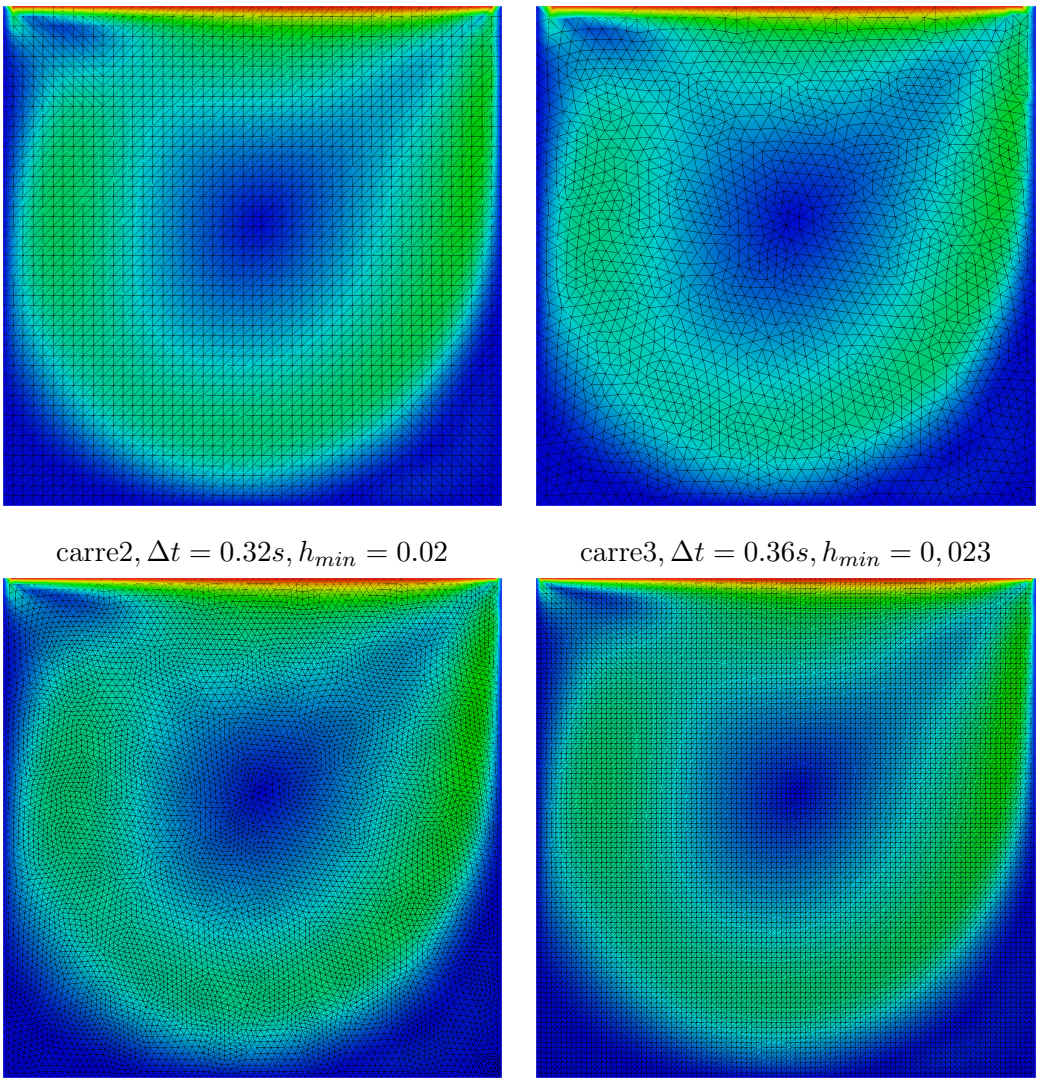


Figure 1.3: *Cavity in 2D: velocity profile for u_x and u_y in case of Re = 1000 with the different time steps.*

(carre2) with 2461 nodes, 5000 elements; an uniform triangulation (carre3) with 2143 nodes and 4136 elements; other uniform triangulation (carre4) with 8421 nodes, 16544 elements; and the last

one is a regular triangulation (carre7) with 10201 nodes, 20000 elements. It is observed that the steady-state is obtained with the different time steps, see figure 1.4 for details. Hence, we propose



carre2, $\Delta t = 0.32s, h_{min} = 0.02$ carre3, $\Delta t = 0.36s, h_{min} = 0,023$

carre4, $\Delta t = 0.23s, h_{min} = 0.012$ carre7, $\Delta t = 0.21s, h_{min} = 0.01$

Figure 1.4: *Cavity in 2D: steady-state solution in case of Re = 1000 for different meshes with the different time steps.*

the good choice for time step is $\Delta t \approx \sqrt{4.5 h_{min}}/|u|$ (where $h_{min} = \min_{K \in T_h} \text{diam}(K)$) for given mesh T_h.)

Similarly, the implement in 3D is compared by the considering the test of Cavity 3D with $Re = 400$ and $Re = 1000$. The CPU time in 3D is detailed on the Table 1.2 for the test case $Re = 400$ and the streamtraces are played out in figure 1.5 for the test case $Re = 1000$. The velocity of the flow and the streamlines in the plane $(z = 0.5)$ for $Re = 1000$ are shown in figure 1.6.

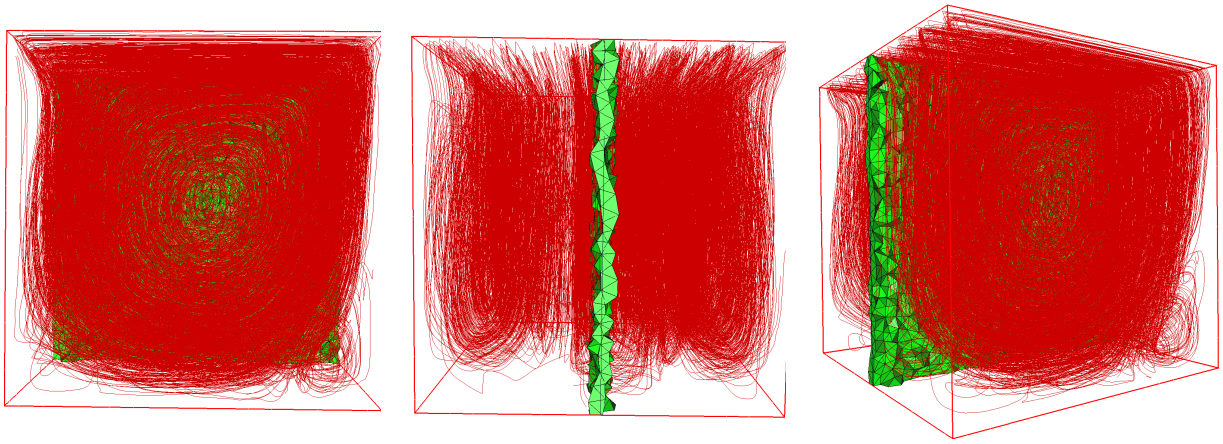


Figure 1.5: *Cavity in 3D: the streamtraces for the test case Re = 1000.*

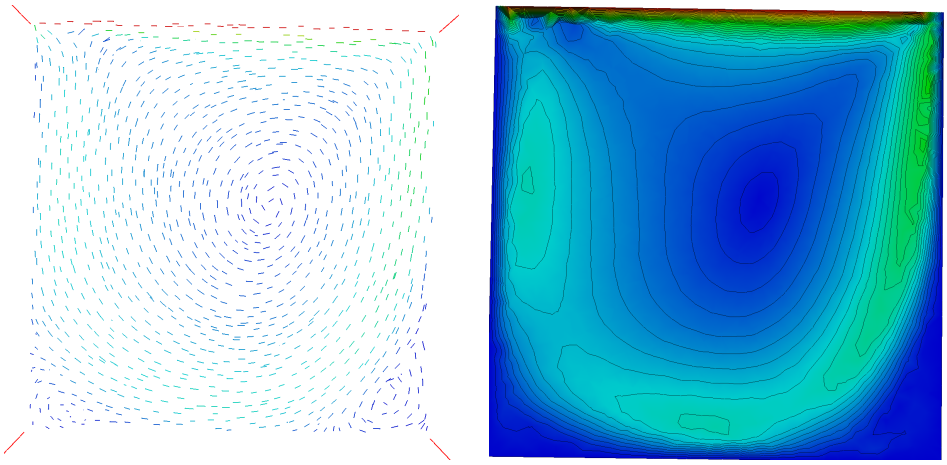


Figure 1.6: *Cavity in 3D: (left) velocity of the flow and (right) streamlines in the plane ($z = 0.5$) for $Re = 1000$.*

Finite element	Scheme of characteristic	Method	Number of iterations	CPU time
$\mathbb{P}_1 b\mathbb{P}_1$	Euler	Uzawa	41	1
$\mathbb{P}_1 b\mathbb{P}_1$	Euler	Penalty	44	0,72
$\mathbb{P}_1 b\mathbb{P}_1$	Runge Kutta 4	Uzawa	45	1,16
$\mathbb{P}_1 b\mathbb{P}_1$	Runge Kutta 4	Penalty	47	0.88

Table 1.2: *Cavity in 3D: Details on CPU time to reach the steady-state with Re = 400, uniform mesh of 4978 vertices, 23837 elements.*

The present solver can handle with the cases of high Reynold number. In fact, we have investigated the test of cavity with the number of Reynolds up to 10000 and the resultats are in good agreement with other references, (see section 4.1).

1.7 Conclusion

We have presented in detail all the features in the resolution of viscous incompressible Navier-Stokes equations in two and three dimensions. The Navier-Stokes solver is capable of dealing with high Reynold number problems. Moreover, it will be still valuable for solving bifluid problems. In fact, the resolution of Navier-Stokes equation for two fluids is one part in our simulation of bifluid flows with interface will be detailed in the next chapter.

Regarding the perspectives for this chapter, we would like to:

- Improve the Lagrange-Galerkin scheme of second order to comparer with the one of first order.
- Consider the $\mathbb{P}1\mathbb{P}1$ elements in combination with a stable scheme that is hoped that dramatically reduce the computational time of solving linear systems, either solving by Uzawa or Penalty method.
- Develope the present solver with a technique of mesh adaptation to treat the more complex test of Navier-Stokes problem.

Chapter 2

Two-phase flow simulations

Contents

In the previous chapter, all the resolution of Navier-Stokes for a single fluid has been described. This chapter continues with the numerical resolution of bifluid flow. In section 2.1 we introduce the model of bifluid flow with the continuous condition on the interface between two fluids. This condition induces effect of surface tension as a localized force added to the right-hand side of the momentum equation. This model for a bifluid incompressible flow problem is often used in the literature, see [GLM06, GRR06] and many references therein. Hence, the motion of two fluids can be achieved by the solver presented in Chapter 1 to obtain the velocity field of the flows.

The appearance of localized surface tension force and its geometrical approximation are described in section 2.2. In our approach, the interface is implicitly captured as zero-level set of a level-set function. The advection of level-set functions by the velocity of the flows (may be extended and regularized) obeys a Hamilton-Jacobi equation will be handled numerically in section 2.3. As known before, in the problem involving incompressible flows, the level-set function does not assure the regularity during the time of evolution leading the diffusion on the interface and difficulties in preserving the mass conservation. In order to overcome these problems, some numerical techniques are proposed in section 2.4 to ameliorate the numerical resolution. A numerical scheme for two-phase flow is proposed in the last section 2.5.

2.1 Introduction the model of two-phase flow

2.1.1 The equation of motion of bifluid flow

We consider the problem of two immiscible fluids illustrated by figure 2.1 (in 2D). At each time

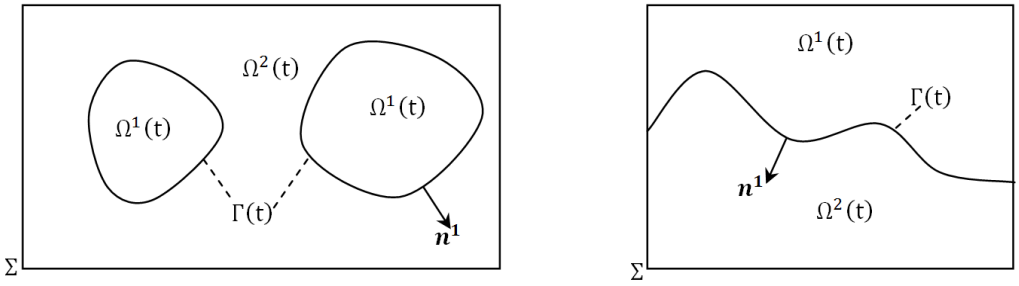


Figure 2.1: *Two examples of computational domains for bifluid flow simulations; left: disconnected components, right: connected components.*

$t \in [0, T]$, the domain Ω in $\mathbb{R}^d$ (d=2;3) is decomposed into two moving subdomain $\Omega^1(t)$ and $\Omega^2(t)$ and the interface between the two fluids is $\Gamma(t) = \overline{\Omega^1(t)} \cap \overline{\Omega^2(t)}$, and we suppose that:

$$\overline{\Omega^1(t)} \cup \overline{\Omega^2(t)} = \overline{\Omega}, \quad \Omega^1(t) \cap \Omega^2(t) = \emptyset$$

Supposing $\rho(\mathbf{x}, t) = \rho(\mathbf{x})$ and $\mu(\mathbf{x}, t) = \mu(\mathbf{x})$, meaning that the physical characteristics of each fluid remain constant during each time step of the resolution. For the sake of simplicity, we omit the dependence on t, write simply Ω^i and Γ instead of $\Omega^i(t)$ and $\Gamma(t)$, at each time $t \in [0, T]$, the motion

of bifluid flow governed by incompressible Navier-Stokes equations can be written as follows:

$$\begin{cases} \rho^i \left(\dfrac{\partial \mathbf{u}^i}{\partial t} + (\mathbf{u}^i \cdot \nabla)\mathbf{u}^i \right) - \mu^i \Delta \mathbf{u}^i + \nabla p^i & = \rho^i \mathbf{f}^i \text{ in } \Omega^i \ (i = 1, 2) \\ \text{div}\mathbf{u}^i & = 0 \quad \text{in } \Omega^i \ (i = 1, 2) \end{cases} \tag{2.1}$$

We introduce the functions μ and ρ for the viscosity and the density of the fluid, respectively defined on the whole domain Ω as follows:

$$\mu = \chi^1 \mu^1 + \chi^2 \mu^2, \qquad \rho = \chi^1 \rho^1 + \chi^2 \rho^2$$

where χ^i is the characteristic function of the subdomain Ω^i and μ^i,ρ^i are the constant dynamic viscosities and densities of each fluid (i=1,2), respectively.

Using these notations and based on the definition ν of the constant *kinematic* viscocity of the fluid for all subdomains, equations (2.1) are rewritten then in reduced form:

$$\begin{cases} \left(\dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) - \nu \Delta \mathbf{u} + \nabla p & = \mathbf{f} \quad \text{in each } \Omega^i \ (i = 1, 2) \\ \text{div}\mathbf{u} & = 0 \quad \text{in } \Omega \end{cases} \tag{2.2}$$

where $\mathbf{u} = \mathbf{u}^i$, $p = p^i$ and $\mathbf{f} = \mathbf{f}^i$ in Ω^i (i=1,2).

These equations need to be endowed by adequate boundary and initial conditions which have been discussed in the previous chapter for single fluid. Besides, they are also endowed with interface conditions imposing: the continuity of the velocity and the balance of the normal stress with the surface tension across the interface $\Gamma(t)$, namely:

$$[\mathbf{u}]_\Gamma = 0, \qquad [\sigma]_\Gamma . \mathbf{n}^1 = -\gamma \kappa \mathbf{n}^1 \tag{2.3}$$

where σ here is the stress tensor defined as: $\sigma = \nu \left(\nabla \mathbf{u} + {}^t\nabla \mathbf{u} \right) - pI_d$, $\mathbf{n}^1$ is the unit normal vector to $\Gamma(t)$, exterior to $\Omega^1(t)$; $[.]_\Gamma$ denotes the jump of quality across Γ in the normal direction of $\mathbf{n}^1$, i.e. $[.]_\Gamma = .|_{\Omega^1} - .|_{\Omega^2}$ and $\gamma > 0$ is the constant surface tension coefficient along the interface; κ is the algebraic mean curvature of the interface, being positive if the interface curve/surface bends towards Ω^1 and negative otherwise.

Remark 2. *The effect of the surface tension here is expressed in term of a localized force $-\gamma\kappa\mathbf{n}^1$ at the interface which is also mentioned as the continuum surface force (CSF) technique in references [BKZ92, CHMO96]. We will show that this localized force of surface tension is added to the right-hand side of the variational formulation in case of bifluid flow (in Section 2.2).*

2.1.2 The interface capturing by level set method

The level set method, introduced by Osher and Sethian in [OS88], has become very popular in the past few years for describing boundaries of domains or evolution of free surfaces and interfaces. This approach appeared in Computational Fluid Dynamics with the study of the motion of two compressible gases, separated by a sharp interface in [MOS92] and was used in [SSO94] for describing the interface between two immiscible incompressible fluids, driven by the Navier-Stokes equations. Since these seminal works, level set method has been extensively used to treat problems involving free surfaces or interfaces between the fluids and strongly promoted in two books by Sethian [Set96] and [Set99].

Following [Set99], we introduce the signed distance function to the interface $\Gamma(t)$ as follows:

$$\phi(\mathbf{x},t) = \pm d(\mathbf{x},\Gamma(t)), \qquad \forall \mathbf{x} \in \Omega(t) \tag{2.4}$$

where $d(\mathbf{x},\Gamma(t))$ denotes the usual Euclidean distance function from $\mathbf{x}$ to $\Gamma(t)$ and:

$$\begin{cases} \phi(\mathbf{x},t) > 0 \,\text{if}\, \mathbf{x} \in \Omega^1(t) \\ \phi(\mathbf{x},t) < 0 \,\text{if}\, \mathbf{x} \in \Omega^2(t) \\ \phi(\mathbf{x},t) = 0 \,\text{if}\, \mathbf{x} \in \Gamma(t). \end{cases} \tag{2.5}$$

See figures 2.2, 2.3 for illustrations of computational domain with the level-set function.

The function ϕ is initialized by the signed distance function to Γ^0:

$$\phi^0(\mathbf{x}) = \pm d(\mathbf{x},\Gamma^0), \qquad \forall \mathbf{x} \in \Omega(0)$$

where Γ^0 is the initial position of the interface and indicates the initial shape of each subdomain $\Omega^i(0)$. We use here the approach suggested in [SSO94] for the incompressible two-phase flows. The interface between two fluids at each time t is then captured as zero-level set of $\phi(\mathbf{x},t)$ which is solution to *advection equation*:

$$\begin{cases} \dfrac{\partial \phi}{\partial t}(\mathbf{x},t) + \mathbf{u}(\mathbf{x},t)\nabla\phi(\mathbf{x},t) & = 0 \quad \forall (\mathbf{x},t) \in \Omega \times \mathbb{R}^+ \\ \phi(\mathbf{x},0) & = \phi^0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega \end{cases} \tag{2.6}$$

where $\mathbf{u}$ is the solution of (2.2) and (2.3). Notice that in these equations, the density and the viscosity are constant in each fluid and take different values depending on the characteristic function χ^i of each subdomain Ω^i (hence, depending on the sign of the level set function):

$$\mu = \chi^1\mu^1 + \chi^2\mu^2, \qquad \rho = \chi^1\rho^1 + \chi^2\rho^2$$

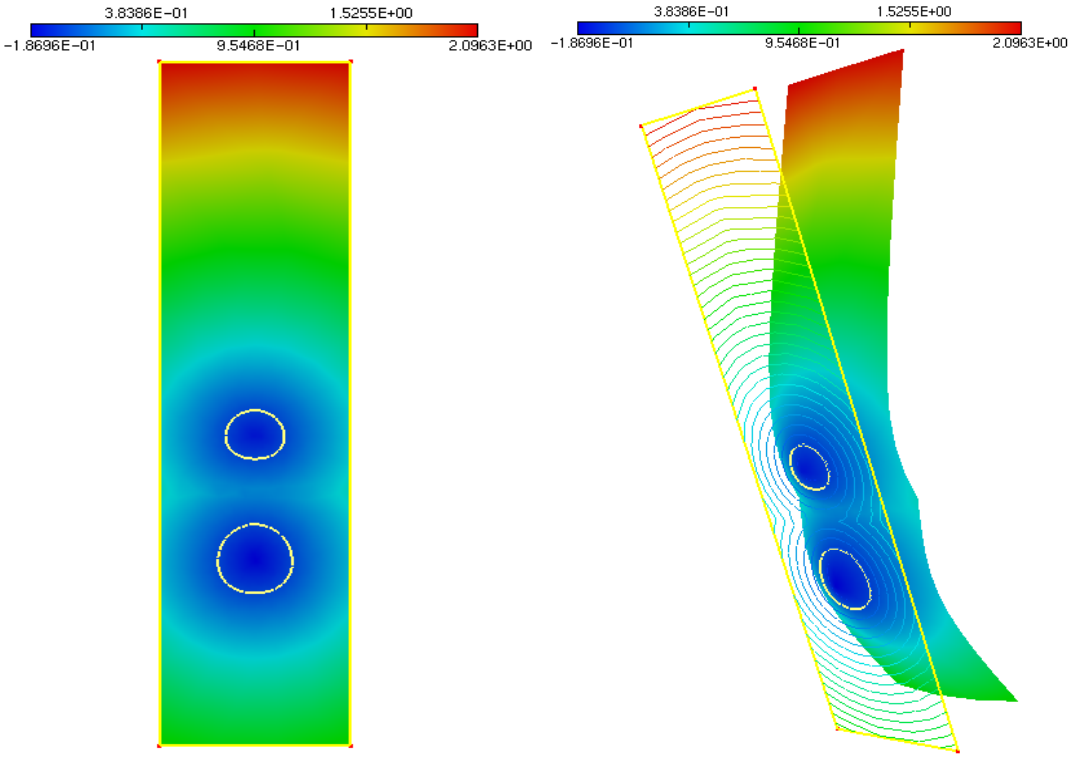


Figure 2.2: *Examples of computational domains with level-set function in 2D.*

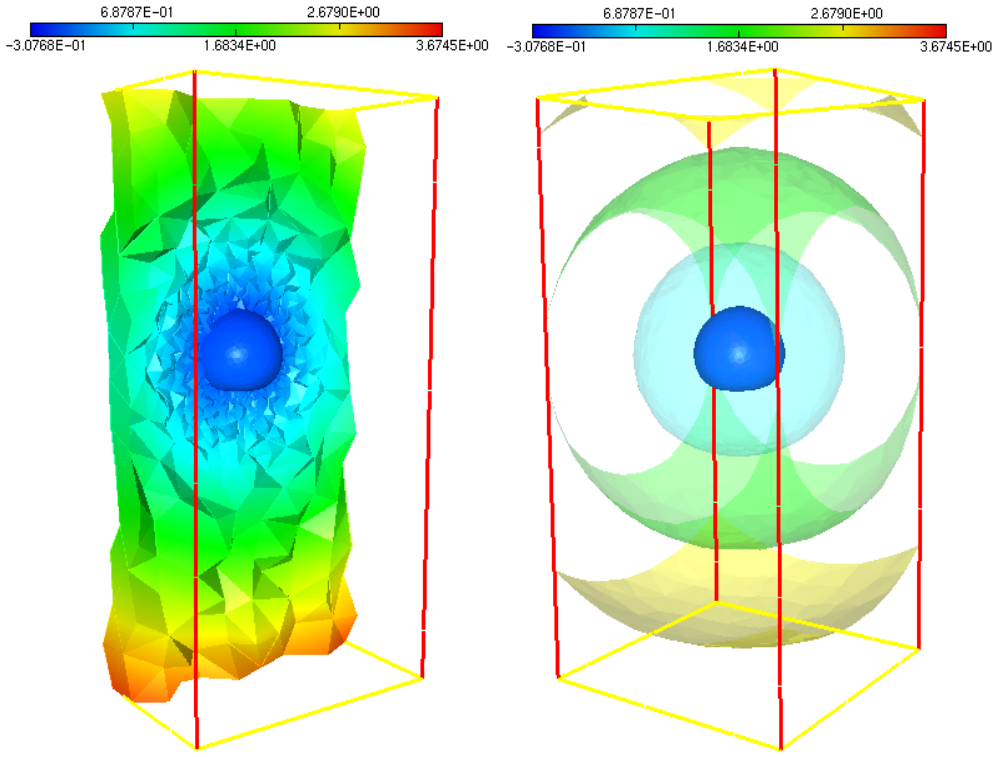


Figure 2.3: *(left) Cutting plane through a computational domains with level-set function and (right) several iso-surfaces in 3D.*

Furthermore, in the level set framework, the unit normal vector $\mathbf{n}$ to the interface and the mean curvature κ at the interface are classically computed via the function ϕ as:

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

However, we will introduce below in section 2.2 another geometric technique to approximate $\mathbf{n}$ and κ, that revealed less sensitive to numerical artifacts when the shape of $\Gamma(t)$ becomes more complex.

2.1.3 The extension and regularity of flow fields

One should note that the evolution of the interface depends strongly only on the flow field in its vicinity. Moreover, in case of complex deformation, the large velocity may cause uncontrolled oscillation and jeopardize the numerical stability. This is the reasons we propose to extend and regularize the velocity of flows $\mathbf{u}$ (defined only on the interface Γ) into vector field $\widetilde{\mathbf{u}}$ (defined on the whole Ω) for advection of the interface. There are many approachs for extension and regularity velocity, for example from suggest in [Bur03] the vector $\widetilde{\mathbf{u}}$ can be seached in a regular space V by solving the following problem:

$$\begin{cases} -\alpha\Delta\widetilde{\mathbf{u}} + \widetilde{\mathbf{u}} &= 0 \quad \text{in}\Omega \\ \widetilde{\mathbf{u}} &= 0 \quad \text{on}\Sigma \\ \widetilde{\mathbf{u}} &= \mathbf{u} \quad \text{on}\Gamma \end{cases} \tag{2.7}$$

where $\alpha > 0$ can be interpreted as a regularization lengthscale. This leads us to the variational problem: Find $\widetilde{\mathbf{u}} \in V$ such that:

$$\alpha\int_\Omega \nabla\widetilde{\mathbf{u}}.\nabla\mathbf{v} + \int_\Omega \widetilde{\mathbf{u}}\mathbf{v} = \int_\Gamma \mathbf{u}\mathbf{v} \quad \forall\mathbf{v} \in V \tag{2.8}$$

It can be seen that the left-hand side of (2.8) is a coercive bilinear form on V which is close to I (so $\widetilde{\mathbf{u}}$ is expected close to $\mathbf{u}$). This equation can be solved easily by a finite element method.

2.1.4 Description of the numerical scheme

The bifluid model is represented by the equations (2.2)-(2.7). In summary, the general scheme to solve this system is the following:

1. Initialization of the level set function ϕ_0 and velocity $\mathbf{u}_0$.
2. At each time step:
   - update the interface Γ, as well as the densities ρ and the viscosities μ of fluids;
   - solve the Navier-Stokes equations to find $\mathbf{u}, p$;

- extend and regularize the velocity **u** to obtain a vector field for solving the advection equation;
- solve the advection equation to obtain ϕ in the next time.

3. Repeat step 2 until the final time.

The reduced scheme is hiding some complex and necessary numerical routines like: solution interpolation, error estimate and mesh adaptation. All these numerical techniques will be detailed in chapter 3. However, let us say few words about mesh adaptation, an important routine. Mesh adaptation aims at improving the accuracy of numerical solution and saving the memory resource. In our approach, the adapted mesh will be generated in order to assure the minimal error estimation of velocity and geometrical approximation of the interface, it plays the role of a computational mesh for both the Navier-Stokes and advection equation in each time step. A more precise scheme of the whole three parts (Navier-Stokes, advection, mesh adaptation) will be presented in section 2.5.

2.2 The surface tension

Surface tension is very important in the simulation of fluid flows with free surface or interface. In the case of interfaces between two fluids, the surface tension contributes a surface pressure that is a normal force per unit interfacial area. Actually, this surface tension balances the normal stress at the interface (see equation (2.3)). Intuitively, the surface tension balances the interaction between two fluids, if the interface is concave in one domain then the surface tension tends to reduce this concavity.

For an illustration of the effect of the surface tension, we consider the evolution of a rising bubble in a fluid. The bubble with lower density than surrounding fluid rises and final at the top of the domain. It has been observed the difference behavior of shape bubble when the surface tension term effects or not in figure 2.4. In the first case, without surface tension ($\gamma = 0$), the bubble rapidly becomes deformed in time while in the second case, under the effect of surface tension, the bubble is almost circular during the evolution.

As mentioned in the previous section, due to the condition on the interface (2.3) the surface tension appears in the right-hand side of the momentum equation as a localized force concentrated at the interface. This section will consider the effect of this term in the equation of bifluid flow (2.2). After, we will treat the geometric approximation of this term using the explicit discretization of the interface in the computational mesh.

Figure 2.4: *Rising bubble in 2D: evolution of the interface in time: (top) without surface tension; (bottom) with surface tension.*

2.2.1 The surface tension term in the variational formulation

In solving Navier-Stokes equations for two fluids flow, the discretization in time the leads us solve the following generalized Stokes equations in each time:

$$\begin{cases} k\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p &= \mathbf{g} \quad \text{in each } \Omega^i \ (i = 1, 2) \\ \text{div}\mathbf{u} &= 0 \quad \text{in } \Omega \end{cases} \tag{2.9}$$

where $\mathbf{u} = \mathbf{u}^i$, $p = p^i$ and $\mathbf{f} = \mathbf{f}^i$ in Ω^i (i=1,2) and k is denoting of $1/\Delta t$.
By introducing $\mathbf{D}(\mathbf{u}) = \dfrac{\nabla\mathbf{u} + {}^t\nabla\mathbf{u}}{2}$, the symmetric gradient of $\mathbf{u}$, also called the *rate of deformation tensor*, and thanks to the condition of incompressibility we have: $\text{div}(2\mathbf{D}(\mathbf{u})) = \Delta\mathbf{u}$.
Using Green's formula we have:

$$\begin{aligned} -\int_{\Omega^i} \text{div}(2\mu\mathbf{D}(\mathbf{u}^i))\mathbf{v}^i dx &= \int_{\Omega^i} 2\mu\mathbf{D}(\mathbf{u}^i) : \nabla\mathbf{v}^i dx - \int_{\partial\Omega^i} 2\mu\mathbf{D}(\mathbf{u}^i)\mathbf{n}^i.\mathbf{v}^i ds \\ &= \int_{\Omega^i} 2\mu\mathbf{D}(\mathbf{u}^i) : \mathbf{D}(\mathbf{v}^i) dx - \int_{\partial\Omega^i} 2\mu\mathbf{D}(\mathbf{u}^i)\mathbf{n}^i.\mathbf{v}^i ds \\ \int_{\Omega^i} \nabla p^i.\mathbf{v}^i dx &= -\int_{\Omega^i} p^i \text{div}\mathbf{v}^i dx + \int_{\partial\Omega^i} p^i\mathbf{n}^i.\mathbf{v}^i ds \end{aligned}$$

Given a test function $\mathbf{v}$, taking the inner product in $L^2(\Omega)^d$ of the first equation of system (2.9) and summing on i, we have following equation:

$$\begin{aligned} k\int_\Omega \mathbf{u}.\mathbf{v} dx + \int_\Omega 2\mu\mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) dx - \int_\Omega p\text{div}\mathbf{v} dx &= \int_\Omega \mathbf{g}.\mathbf{v} dx \\ &+ \sum_{i=1}^{2} \int_{\partial\Omega^i} 2\mu\mathbf{D}(\mathbf{u}^i)\mathbf{n}^i.\mathbf{v}^i ds - \sum_{i=1}^{2} \int_{\partial\Omega^i} p^i\mathbf{n}^i.\mathbf{v}^i ds \end{aligned} \tag{2.10}$$

Due to the interface condition (2.3) and the geometric condition $\mathbf{n}^2 = -\mathbf{n}^1$ on Γ and noticing also that $\mathbf{v} = 0$ on Σ we have:

$$\begin{aligned} \sum_{i=1}^{2} \int_{\partial\Omega^i} 2\mu\mathbf{D}(\mathbf{u}^i)\mathbf{n}^i.\mathbf{v}^i ds - \sum_{i=1}^{2} \int_{\partial\Omega^i} p^i\mathbf{n}^i.\mathbf{v}^i ds &= \sum_{i=1}^{2} \int_\Gamma (2\mu\mathbf{D}(\mathbf{u}^i) - p^i I)\mathbf{n}^\mathbf{i}\mathbf{v}^i ds \\ &= \int_\Gamma (\sigma^1\mathbf{n}^1\mathbf{v}^1 - \sigma^2\mathbf{n}^1\mathbf{v}^2) ds \\ &= \int_\Gamma -\gamma\kappa\mathbf{n}^1.\mathbf{v} ds \end{aligned}$$

This term corresponds to the effect of the surface tension on the interface in the variational formulation. As can be seen that the surface tension is taken into account as a localized force $-\gamma\kappa\mathbf{n}^1$ added to the right-hand side of the momentum equation and proportioned to the local mean curvature of the interface.

2.2.2 The discretisation of the surface tension term

In the context of level set method, the level-set function allows to calculate the unit normal vector and the mean curvature at the interface by following formulas:

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}|_{\phi=0}; \quad \text{and} \quad \kappa = \text{div}\mathbf{n} = \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)|_{\phi=0} \tag{2.11}$$

Introducing δ_Γ is a Dirac δ-function with support on Γ acting on a smooth test function φ as follows:

$$\int_\Omega \delta_\Gamma \varphi dx = \int_\Gamma \varphi ds \tag{2.12}$$

The surface tension force is then represented by, see also in [SSO94] :

$$\begin{aligned} l_\Gamma(\mathbf{v}) : &= -\int_\Gamma \gamma\kappa\mathbf{n}^1.\mathbf{v}ds \\ &= -\int_\Omega \gamma\kappa\delta_\Gamma(\phi)\nabla\phi.\mathbf{v}dx \end{aligned}$$

Hence, we have the expression of the surface tension in the discrete formulation:

$$l_{\Gamma_h}(\mathbf{v_h}) = \sum_{K\in T_h} -\int_K \gamma\kappa\delta(\phi)\nabla\phi.\mathbf{v_h}dx \tag{2.13}$$

However, this approach requires an approximation of the gradient of ϕ which is known that error-prone on unstructured triangulations. In addition, the main challenge is to compute the mean curvature κ which is related to a *second* order derivative of the level-set function.
Another approach has been used to compute the effect of surface tension based on the so-called Laplace-Beltrami operator. The Laplace-Beltrami operator Δ_Γ is a function of the tangential gradient ∇_Γ and is defined for a given function f by:

$$\Delta_\Gamma f = \nabla_\Gamma.\nabla_\Gamma f \tag{2.14}$$

with ∇_Γ is the tangential derivative (along Γ):

$$\nabla_\Gamma f = \nabla f - \mathbf{n}_\Gamma {\mathbf{n}_\Gamma}^t \nabla f = (I - \mathbf{n}_\Gamma {\mathbf{n}_\Gamma}^t)\nabla f \tag{2.15}$$

then the second order derivatives of curvature can be eliminated by the following formulation (see [GR11, GRR06, Hys06]):

$$-\Delta_\Gamma id_\Gamma(\mathbf{x}) = \kappa(\mathbf{x})\mathbf{n}(\mathbf{x}), \mathbf{x}\in\Gamma \tag{2.16}$$

Using this result, we see that the surface tension term can be rewritten as follows:

$$l_\Gamma(\mathbf{v}) = -\gamma\int_\Gamma \nabla_\Gamma id_\Gamma.\nabla_\Gamma\mathbf{v} \tag{2.17}$$

Given an approximation Γ_h of Γ, the localized force term $l_\Gamma(\mathbf{v_h})$ can be approximated by:

$$l_{\Gamma_h}(\mathbf{v_h}) = -\gamma \int_{\Gamma_h} \nabla_{\Gamma_h} id_{\Gamma_h} . \nabla_{\Gamma_h} \mathbf{v}_h \tag{2.18}$$

The discretization of the surface tension by this approach has been detailed in [GR11], however the computation is rather complex as requires also the approximation of the normal vector according to the first formula of (2.11).

In our approach the interface is explicitly discretized in the triangulation T_h, this representation gives us an alternate technique to approximate the interface via a set of connected segments (faces in three dimensions). Denoting $(\mathbf{x_i})_{1\leq i\leq ns}$ 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 $\mathbf{x_0} \equiv \mathbf{x_{ns}}, \mathbf{x_1} \equiv \mathbf{x_{ns+1}}$ in case Γ_h is closed curve, see figure 2.5. Using quadrature formula along each edge E of Γ_h, the surface tension

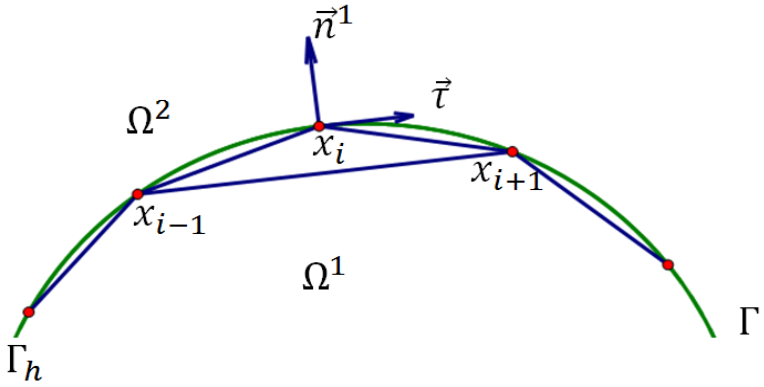


Figure 2.5: *Ilustration of geometric discretization of interface in 2D.*

term can be rewritten as follows, for all $\mathbf{v}_h \in V_h$:

$$\begin{aligned} l_{\Gamma_h}(\mathbf{v}_h) :&= -\int_{\Gamma_h} \gamma\kappa \mathbf{n}_h^1 . \mathbf{v}_h ds \\ &= \sum_{E\subset\Gamma_h} \frac{|E|}{2} \sum_{\mathbf{x}_i\in E} \gamma\kappa(\mathbf{x}_i)\mathbf{n}_h^1(\mathbf{x}_i).\mathbf{v}_h(\mathbf{x}_i) \\ &= \sum_{\mathbf{x}_i\in\Gamma_h} \gamma\kappa(\mathbf{x}_i)\mathbf{n}_h^1(\mathbf{x}_i).\mathbf{v}_h(\mathbf{x}_i) \sum_{E\ni\mathbf{x}_i} \frac{|E|}{2} \end{aligned} \tag{2.19}$$

where the unit normal vector $\mathbf{n}^1$ is computed from the approximation of the unit tangent vector $\tau = (\tau_1, \tau_2)^t$ at each vertex $\mathbf{x}_i \in \Gamma_h$: $\tau(\mathbf{x_i}) = \overrightarrow{\mathbf{x_{i+1}x_{i-1}}} / \|\overrightarrow{\mathbf{x_{i+1}x_{i-1}}}\|$, hence $\mathbf{n}(\mathbf{x_i}) = (\tau_2(\mathbf{x_i}), -\tau_1(\mathbf{x_i}))^t$. 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]:

$$r(\mathbf{x_i}) = \frac{1}{4}\left(\frac{\langle \overrightarrow{\mathbf{x_i x_{i-1}}}, \overrightarrow{\mathbf{x_i x_{i-1}}} \rangle}{\langle -\mathbf{n^1}(\mathbf{x_i}), \overrightarrow{\mathbf{x_i x_{i-1}}} \rangle} + \frac{\langle \overrightarrow{\mathbf{x_i x_{i-1}}}, \overrightarrow{\mathbf{x_i x_{i+1}}} \rangle}{\langle -\mathbf{n^1}(\mathbf{x_i}), \overrightarrow{\mathbf{x_i x_{i+1}}} \rangle} \right)$$

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 $\mathbf{x}_i$. Other formula can be used to approximate $r(\mathbf{x})$ or $\kappa(\mathbf{x})$ due to explicit discretization of geometric of interface, see for instances [TBE+01] or [GLM06].

2.3 Numerical resolution of the level set advection equation

We sketch out in this section numerical resolution of the evolution of the interface. We recall that, at each time step, the moving of the interface between two fluids is represented by the zero isocontour of the level set function ϕ that is convected by the velocity field to find the new position of the interface. The advection of the level set function by the flow velocity $\mathbf{u}$ (may be extended and regularized) obeys the so-called *advection equation*:

$$\frac{\partial \phi}{\partial t}(\mathbf{x},t) + \mathbf{u}(\mathbf{x},t).\nabla\phi(\mathbf{x},t) = 0 \,\forall\, (\mathbf{x},t) \in \Omega \times [O,T]. \tag{2.20}$$

2.3.1 The discretisation in time by the method of characteristics

Consider the Cauchy problem for advection equation has been introduced in (2.20) : Given an initial function $\phi^0(\mathbf{x}) : \Omega \to \mathbb{R}$ and a velocity field $\mathbf{u}(\mathbf{x},t) : \Omega \to \mathbb{R}^d$ defined on Ω, find $\phi(\mathbf{x},t) : \Omega \times [O,T] \to \mathbb{R}$ such that:

$$\begin{cases} \dfrac{\partial \phi}{\partial t}(\mathbf{x},t) + \mathbf{u}(\mathbf{x},t)\nabla\phi(\mathbf{x},t) & = 0 \qquad \forall (\mathbf{x},t) \in \Omega \times (0,T) \\ \phi(\mathbf{x},0) & = \phi^0(\mathbf{x}) \qquad \forall \mathbf{x} \in \Omega \end{cases} \tag{2.21}$$

This problem is solved by following the characteristic curves of the fluid particules. Given a particule $\mathbf{x} \in \Omega$ at the time s, its curve is described by the following equations:

$$\begin{cases} \dfrac{d\mathbf{X}(\mathbf{x},s;t)}{dt} & = \mathbf{u}(\mathbf{X}(\mathbf{x},s;t),t) \;\forall t \in (0,T) \\ \mathbf{X}(\mathbf{x},s;s) & = \mathbf{x} \end{cases} \tag{2.22}$$

where $\mathbf{X}(\mathbf{x},s;t)$ is the position of $\mathbf{x}$ at the time t.
The first equation of (2.21) implies that $\phi(\mathbf{x},t)$ is constant along the characteristic lines $\mathbf{X}(\mathbf{x},s;t)$. Hence the solution of the Cauchy problem (2.21) can be expressed as:

$$\phi(\mathbf{x},t) = \phi^0(\mathbf{X}(\mathbf{x},t;0),0) \quad \forall (\mathbf{x},t) \in \Omega \times [0,T] \tag{2.23}$$

Assume that the interval $[0,T]$ is divided into a finite number intervals Δt of the form (t^{n-1},t^n) with $t^n = n\Delta t$, we have the discretization in time of the equations (2.22) for all n:

$$\begin{cases} \dfrac{d\mathbf{X}(\mathbf{x},t^n;t)}{dt} &= \mathbf{u}(\mathbf{X}(\mathbf{x},t^n;t),t)\ \forall t \in (t^{n-1},t^n) \\ \mathbf{X}(\mathbf{x},t^n;t^n) &= \mathbf{x} \end{cases} \tag{2.24}$$

We only compute $\phi(\mathbf{x},t^n)$ for all n, so if denote $\phi(\mathbf{x},t^n)$ by $\phi^n(\mathbf{x})$, by substituting the time interval $[t^{n-1},t^n]$ into (2.23), we obtain the following result:

$$\phi^n(\mathbf{x}) = \phi^{n-1}(\mathbf{X}(\mathbf{x},t^n;t^{n-1})) \quad \forall \mathbf{x} \in \Omega \tag{2.25}$$

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

The expression (2.25) follows us to calculate ϕ^n from ϕ^{n-1}. In numerical solution, this problem can be solved by a Galerkin numerical scheme which involves the resolution of a linear system with using the quadrature formulas for approximating the integrals in the right-hand side (see [BNV06] for example). Another alternative is simply combination with a Lagrange interpolation, i.e $\phi^n(\mathbf{x})$ is computed by values of $\phi^{n-1}(\mathbf{x})$ at the degrees of freedom of element K which locates $\mathbf{X}(\mathbf{x},t^n;t^{n-1})$. In fact, the first approach is more expensive than the second one. Moreover, the second approach brings an estimation to control the geometric error of the interface by the interpolation error (see section 3.3) that is very efficient in generation of mesh adaptation, so, its computation is applied for present work.

Algorithm 3: Numerical scheme for advection equation

n=0, start with the initializations (ϕ^0,Γ^0);

for *n=1,...,N* **do**

1. Solve the ODE (2.24) to search $\mathbf{X}^{n-1}(\mathbf{x})$ for each node $\mathbf{x}$ of mesh;
2. Compute $\phi^{n-1}(\mathbf{X}^{n-1}(\mathbf{x}))$ for each node $\mathbf{x}$ of mesh;
3. Update the new level set values $\phi^n(\mathbf{x})$ according to (2.25).

end

2.3.2 Spatial approximation of the characteristic points

Let us now explain the approximation of the characteristic lines, or more precisely the approximation of $\mathbf{X}(\mathbf{x},t^n;t^{n-1})$ in space. We notice that in numerical resolution only an approximation $\mathbf{u}_h$ of $\mathbf{u}$ is known at each degree of freedom of computational mesh T_h of the domain, and we need to

compute an approximation of $\mathbf{X}_h(\mathbf{x}, t^n; t^{n-1})$ where $\mathbf{X}_h(\mathbf{x}, t^n; t^{n-1})$ is solution at the time t^{n-1} of the approximated characteristic curve:

$$\begin{cases} \dfrac{d\mathbf{X}_h(\mathbf{x}, t^n; t)}{dt} & = \mathbf{u}_h(\mathbf{X}_h(\mathbf{x}, t^n; t), t) \\ \mathbf{X}_h(\mathbf{x}, t^n; t^n) & = \mathbf{x} \end{cases} \tag{2.26}$$

that implies the "formal" expression:

$$\mathbf{X}_h(\mathbf{x}, t^n; t) = \mathbf{x} - \int_t^{t^n} \mathbf{u}_h(\mathbf{X}_h(\mathbf{x}, t^n; t), t)dt \tag{2.27}$$

There are many approaches to approximate $\mathbf{X}_h(\mathbf{x}, t^n; t^{n-1})$. For example the simplest algorithm for computing the characteristics is obtained directly from (2.27) as:

$$\mathbf{X}_h(\mathbf{x}, t^n; t^{n-1}) = \mathbf{x} - \Delta t \mathbf{u}_h(\mathbf{x}) \tag{2.28}$$

In this way, the characteristic curve is considered to be a straight line from the starting point $\mathbf{x}$ to its foot $\mathbf{X}_h(\mathbf{x}, t^n; t^{n-1})$. But in fact, this algorithm should be used only when the time step is small enough, one should not travel too many elements mesh at once to ensure the numerical stability. So, we prefer to use Runge Kutta 4 or Euler's scheme, especially in case the time step is relatively large. Given a substep δt of each interval $[t^{n-1}, t^n]$ and suppose that existing an integer number M such that: $\Delta t = M\delta t$, we have the backing forward schemes as follows:

1. Euler's scheme:

$$\begin{cases} \mathbf{X}_h(\mathbf{x}, t^n; t^n) & = x \\ \mathbf{X}_h(\mathbf{x}, t^n; t^n - m\delta t) & = \mathbf{X}_h(\mathbf{x}, t^n; t^n - (m-1)\delta t) - \delta t \mathbf{u}_h(\mathbf{X}_h(\mathbf{x}, t^n; t^n - m\delta t)); \\ & (m = 1, ..., M) \end{cases} \tag{2.29}$$

2. Runge-Kutta 4 scheme:

$$\begin{cases} \mathbf{X}_h(\mathbf{x}, t^n; t^n) & = x \\ \mathbf{X}_h(\mathbf{x}, t^n; t^n - m\delta t) & = \mathbf{X}_h(\mathbf{x}, t^n; t^n - (m-1)\delta t) - \dfrac{\delta t}{6}(v_1 + 2v_2 + 2v_3 + v_4), \text{with :} \\ v_1 & = \mathbf{u}_h(\mathbf{X}_h(\mathbf{x}, t^n; t^n - m\delta t)) \\ v_2 & = \mathbf{u}_h(\mathbf{X}_h(\mathbf{x}, t^n; t^n - m\delta t) - \dfrac{\delta t}{2} v_1) \\ v_3 & = \mathbf{u}_h(\mathbf{X}_h(\mathbf{x}, t^n; t^n - m\delta t) - \dfrac{\delta t}{2} v_2) \\ v_4 & = \mathbf{u}_h(\mathbf{X}_h(\mathbf{x}, t^n; t^n - m\delta t) - \delta t v_3); (m = 1, ...M) \end{cases} \tag{2.30}$$

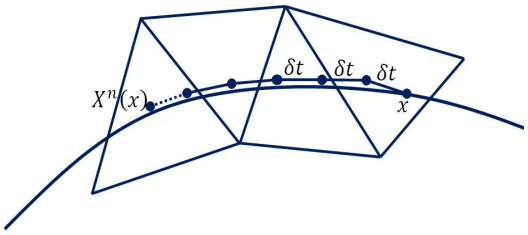


Figure 2.6: *Ilustration of characteristic curve approximation.*

See figure 2.6 for illustration of characteristic curve approximation.

Remark 3.

1. *At now, two different time steps have been used. The first one, $\Delta t = t^n - t^{n-1}$, is a "large" time step on a generic period of time $[0, T]$, that is the period of time for which we assume the velocity and physical properties of the fluids are unchanged. Time step Δt is then used for both resolution of Navier-Stokes and advection equation. The second one, $\delta t < \Delta t$ is a sub-time step involved the integration of the characteristic curves. In fact, the approximation of characteristic curves with the hold time step Δt means that the curves $(\mathbf{x}, \mathbf{X}_h(\mathbf{x}, t^n; t^{n-1}))$ is approximated by straight line from the starting point $\mathbf{x}$ to its foot $\mathbf{X}_h(\mathbf{x}, t^n; t^{n-1})$ and directed only by $\mathbf{u}(\mathbf{x})$, this may lead to inaccuracy dramatically. By using δt we achieve a polygonal approximation with the velocity field is updated more frequently, so we can improve the approximation of the characteristic curve.*
2. *The approximation by these schemes needs the information of the velocity at the immediate points as $\mathbf{u}_h(\mathbf{X}_h(\mathbf{x}, t^n; t^n - m\delta t))$, ... that always requires the exact location of these points. Normally, given a point $\mathbf{x}$ located at element K, the immediate points is somewhere in K or adjacents of K. Therefore, the efficiency of the algorithm is related to the fast identification of the triangle adjacent to a given triangle. To this end, we use a specific data structure in which the three adjacent triangles are associated with the informations of each triangle (the so-called adjacency matrix [FG08]).*
3. *Some time, the procedure of searching the characteristic point can not implemented for all the interval (t^{n-1}, t^n), especially with the vertices next to the boundary of domain. This implies that during the procedure, one of immediate points is go out to domain (see figure 2.7). Although, in that case these vertices need to be traveled with the maximal time that possible, may be this leads to the last point hit the boundary. In fact, this requires one algorithm of calculating the time dt that is necessary for one point to hit an edge or go out the element contains it, see*

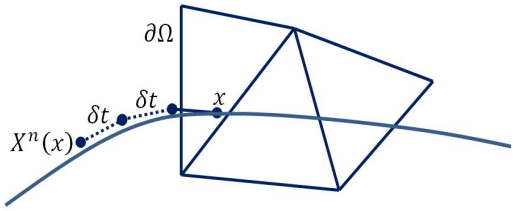


Figure 2.7: *Ilustration of characteristic points is go out to domain.*

figure 2.8. In case of $dt < \delta t$ and we can't find the next point, the last point of characteristic

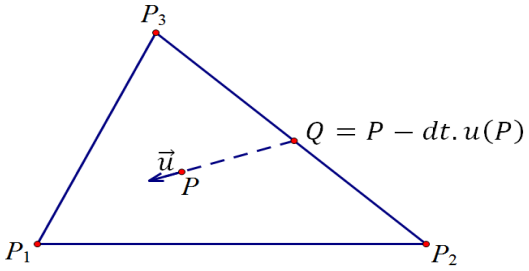


Figure 2.8: *Ilustrations of time dt to go out an element.*

curve will be obtained by travel with the time dt.

Algorithm 4: Calculation time go out the containing element of a point

Input: starting point P in element K, the velocity $\mathbf{u}$

Output: time dt such that $P - dt * \mathbf{u}(P)$ is on an edge of K

Initialisation: $dt = 10e6$

1. Calcul the barycentric $\lambda_{i(1\leq i\leq 3)}$ of point P.
2. Calcul the barycentric $\eta_{i(1\leq i\leq 3)}$ of point $P - \mathbf{u}(P)$.
3. **for** *i=1,...3* **do**
   If ($\lambda_i > \eta_i$ and $dt > \frac{\lambda_i}{\lambda_i - \eta_i}$) then $dt := \frac{\lambda_i}{\lambda_i - \eta_i}$

end

2.4 Redistancing procedure and conversation the mass

It has been known that, in the context of level set method, the level set function must usually satisfy:

$$|\nabla \phi| = 1 \tag{2.31}$$

Unfortunately, when ϕ is transported by a physical velocity field, this property is not preserved as the isolines do not travel at the same speed and the level set does not remain a distance function. A natural choice to reinitialize the level set function is the signed distance function to the interface:

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

In our scheme, this signed distance function is computed approximatively by *redistancing procedure* studied in [DF11], we only briefly recall here the implement of this procedure:

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

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

– Step 2: Calculation numerically ϕ as *steady solution* of so-called *Eikonal equation*:

$$\begin{cases} \dfrac{\partial \phi}{\partial t}(\mathbf{x}, t) + sgn(\phi_0)(|\nabla \phi| - 1) = 0 & \forall (\mathbf{x}, t) \in \Omega \times (0, T) \\ \phi(\mathbf{x}, 0) = \phi^0(\mathbf{x}) & \forall \mathbf{x} \in \Omega \end{cases} \tag{2.32}$$

Its purposes are to remain the behaviour in "vicinity" of the zero isoline, i.e. the position of interface Γ is not modified and to ensure the constraint (2.31).

As pointed out, other drawbacks of this method in case of free surface problems involving incompressible flows is the lack of mass conservation. In this work no particular treatment for mass conservation during the simulation, this problem is remedied by replacing zero-level set by ϵ-level set with ϵ is appropriate negative constant and depends strongly on quality of computational mesh. It has been observed that without this correction of level set function, the derivation of mass is not notable when our mesh adaptation asks a lot of elements at each iteration, however, for the efficiency of computational time especially intending to the problem in 3D, this cure is applied in our approach.

2.5 The proposed scheme

In this section we will describe the general scheme for all the time $[0, T]$. Suppose that $[0, T]$ is divided by N subintervals $[t^{n-1}, t^n]$. At n-iteration, we have to solve the Navier-Stokes equation

and the advection equation approximatively, emphasize that they are solved on the same mesh T^n to obtain the approximative solution $(\mathbf{u}^n, p^n)$ and ϕ^n.

Our scheme is an iterative procedure based on the mesh adaptation process. At each time t^n, given mesh T^n, initialization $\mathbf{u}_0^n$ for velocity and ϕ_0^n for level-set function. After solving Navier-Stokes problem we have the new velocity field $\mathbf{u}^n$, this velocity (may be extended and regularized) is used for solving the advection equation to obtain a new level-set ϕ^n, the new interface Γ^n is captured as the zero-level of ϕ^n. The mesh T^n is adapted with $\mathbf{u}^n, \phi_0^n, \phi^n$ in order to create a new mesh T^{n+1}. The generation of adapted mesh at each time step will be detailed in chapter 3. The projection of $\mathbf{u}^n$ and ϕ^n on T^{n+1} are implemented to have initialisation $\mathbf{u}_0^{n+1}$ and ϕ_0^{n+1} for the $(n+1)$-iteration. The overall algorithm is given as following:

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

1. Start with mesh T^1 and initialization $\mathbf{u}_0^1, \phi_0^1$
2. **For** n = 1, ..., N **do**:

	Problem	Input	Output
2.1	Solving Navier Stokes	$(T^n, \mathbf{u}_0^n)$	$\mathbf{u}^n$
2.2	Solving Advection	$(T^n, \mathbf{u}^n, \phi_0^n)$	ϕ^n
2.3	Extension and regularity	$(T^n, \mathbf{u}^n\|_\Gamma)$	$\mathbf{u}^n$
2.4	Adaptation	$(T^n, \phi^n, \phi_0^n, \mathbf{u}^n)$	T^{n+1}
2.5	Redistancing	(T^{n+1}, ϕ^n)	ϕ^n
2.6	Projection	$(\mathbf{u}^n, \phi^n, T^{n+1})$	$\mathbf{u}_0^{n+1}, \phi_0^{n+1}$

3. **Return** $(\mathbf{u}^N, \phi^N, T^N)$

In comparison with previous study in [BFM10] this scheme has been much reduced. In the before approach mesh adaptation needed two mesh independent, one for the resolution of the fluid and other for the advection of the interface, so, the interpolations is always necessary to correspond to each solving of the problem. The simplicity of present scheme is due to the requirement only one adapted mesh at each time of the simulation, yet all difficulties have been driven for mesh adaptation. On the one hand, the adapted mesh has to respond the capturing of the interface with high quality, especially in the vicinity of interface. On the other hand, the solving Navier-Stokes equation requires the accuracy of numerical solution of velocity of fluids and the efficiency of the numerical scheme depends strongly on the quality of the computational mesh. However, we are only interested in the vicinity of the interface, so mesh adaptation is only much concerned in this "important"

region of the computational mesh, resulting actually our mesh adaptation is implemented rapidly and efficiently.

2.6 Conclusion

We have presented all the numerical scheme for bifluid flows simulation. Our scheme is based on unstructured adapted mesh at each time step. This strategy has several assets; on the one hand, no projection is needed between different meshes. In addition, the method does not present any theoretical difficulty to the extension from the two-dimensional case to the three-dimensional case. This is a tremendous feature insofar as mesh adaptation is concerned; in fact 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. We go to the next chapter to detail the mesh adaptation process for the previous scheme.

Chapter 3

Mesh adaptation

Contents

This chapter is devoted to mesh adaptation. It is known that in numerical simulation based on finite element methods, the accuracy of the solution depends strongly on the quality of the computational mesh. The purpose of mesh adaptation is to increase the efficiency of the numerical schemes and improve the accuracy of numerical solutions. This process relies on:

- the control of a priori/ posteriori error approximations (interpolation error, geometry error,...) of numerical solutions.
- the quality assessment of mesh elements respect to an appropriate quality function.
- the concentration of more degrees of freedom in important regions (such as boundary/interface or regions of large solution variations) than in any other region of the computational domain.

Consequently, the adapted mesh enjoys a nice quality with fewer elements that ultimately results in a diminution of the computational cost to achieve the desired accuracy.

We will introduce hereafter two mesh adaptation procedures which have been used in our numerical experiment.

1. *Anisotropic mesh adaptation.* This is the inheritance of a mesh adaptation technique that has been developed for many years, see [AF03, FA05, Fre08] and applied in many previous works for the model of level set advection equations, see [CDF08, CDF12] and the model of Stokes bifuid flow in 2D [BFM10]. In this thesis, this anisotropic mesh adaptation is ameliorated to solve the model of Navier- Stokes bifuid flow in two dimensions.
2. *Discrete three dimensional domain remeshing.* This is a tremendous developing of mesh adaptation since 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. Thanks to the results of this mesh adaptation process (see [DDF14]) several test cases of bifluid flows in three dimensions have been implemented efficiently, as will be seen later in chapter 4.

The outline of this chapter is the following: In section 3.1 we recall some basic notations and definitions of mesh adaptation. Section 3.2 and section 3.3 describe the two previous mesh adaptation procedures and at the end of each section we will explain how to construct the adapted mesh for two fluids flow problem in chapter 2.

3.1 Basic notations and definitions

3.1.1 Isotropic and anisotropic mesh adaptation

Definition 1. *Let Ω be a open, bounded, polygonal domain in $\mathbb{R}^d$ (d=2,3) and given its simplicial mesh T (i.e triangulation in two dimensions, tetrahedralization in three dimensions). We assume that each element K in T is a closed d-simplex (triangle in two dimensions, tetrahedra in three dimensions) and $\overline{\Omega} = \bigcup_{K \in T} K$. T is called a conforming mesh if satisfies the so-called* conformity condition *such that the intersection of any two different simplices, if not empty, is reduced to:*

– *either a point, or a common edge in two dimensions,*
– *either a point, or a common edge or a common (triangular) face in three dimensions.*

See figure 3.1 for examples of conforming and non conforming meshes.

Given T is a conforming mesh, then T is:

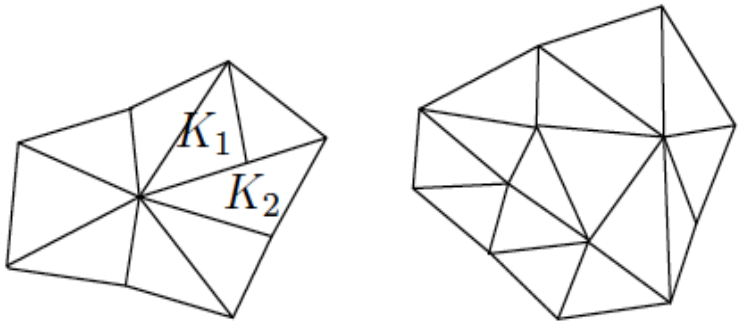


Figure 3.1: *Examples of non conforming mesh (left) and conforming mesh (right).*

– a *uniform* mesh, if all its simplices are equally sized and regular (equilateral),
– a *quasi-uniform* mesh, if the variation of its simplices size is bounded and there exists a constant c such that $\frac{h_K}{\rho_K} \leq c$ for all elements K in T with h_K is diameter of longest edge of K and ρ_K is radius of its inscribed circle or sphere.

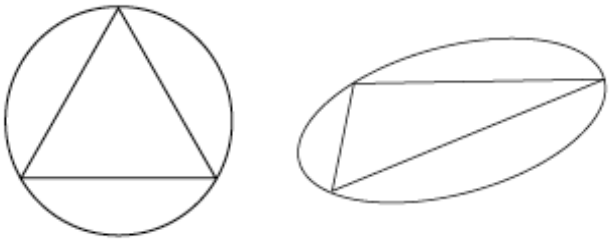


Figure 3.2: *Example of an element in isotropic (left) and anisotropic (right) mesh adaptation with circumscribed circle and ellipse.*

Definition 2. *We introduce the following notations:*

– Isotropic *mesh adaptation generates the mesh where almost elements are regular (equilateral) and only adjusted in* size *based on an error estimate.*
– Anisotropic *mesh adaptation based on a metric tensor consists in controlling the* size, *the* shape *and the* orientation *of the elements altogether.*

See figure 3.2 and figure 3.3 for illustrations of isotropic and anisotropic mesh adaptation.

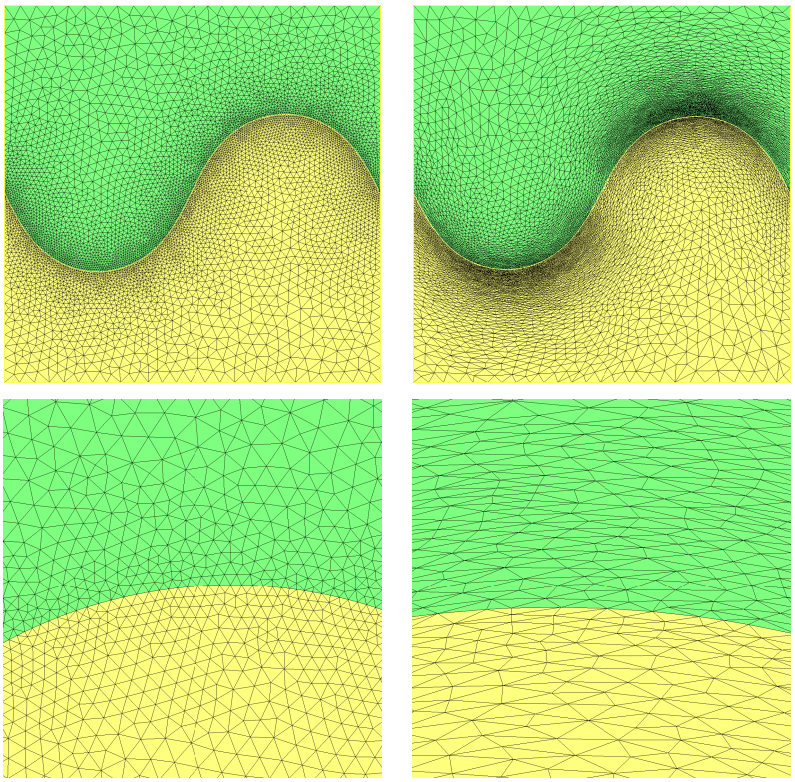


Figure 3.3: *Example of mesh adaptation about 7000 vertices: isotropic mesh (top, left) and anisotropic mesh (top, right); zoom in the vicinity of the interface triangulations (bottom).*

Remark 4.

- *Both anisotropic and isotropic mesh adaptation have been applied in practice. Traditionally, isotropic mesh adaptation has received much attention, however, in regions of large solution gradient, adaptive isotropic meshes usually contain too many elements. Moreover, in several cases elements with high aspect ratio are needed for better represent of the solution variations or boundary layers in fluid dynamics, thus it is desirable to adjust the element size as well as the element shape, resulting in an anisotropic mesh.*
- *From the geometrical point of view, isotropic mesh adaptation can be considered as a special case of anisotropic one with a special metric tensor (see below in section 3.2.1).*

3.1.2 The quality of a mesh

In addition to the requirement of adapting mesh to approximate the continuous domain Ω, we also need to define criteria to evaluate the quality of an element K of a simplicial mesh T.

There are many definitions of the quality of a simplex $K \in T$. For instance one can rely on the ratio $\frac{\rho_K}{h_K}$, see [Cia78]. This measure only depends on the shape of the considered simplex and not on its size. From practical point of view, this implies that the accuracy of a finite element computation performed on T is of course influenced by the size of its elements, but also by their "well-shapeness". However, we will rather rely on the following quality function $\mathcal{Q}(K)$ of a d-simplex K:

$$\mathcal{Q}(K) := \alpha \frac{V(K)}{(\Sigma_{i=1}^{na} l(e_i))^{d/2}} \tag{3.1}$$

where $V(K)$ is the area/volume of K, $na = d(d+1)/2$ is the number of edges of K denoted by $e_1, \ldots, e_{na}$ and $l(e_i)$ stands for the length of e_i. The quality function in (3.1) retains the same theoretical meaning as $\frac{\rho_K}{h_K}$, but shows a better numerical ability when it comes to discriminating "good" from "average", or "bad" elements. One can show that, for any tetrahedron K, $\mathcal{Q}(K) \leq 1$, and equality holds if and only if K is regular.

3.2 Anisotropic mesh adaptation

Research on anisotropic adaptation is based on the idea of metric-based mesh adaptation: the local desired size, shape and orientation of the elements are prescribed using a metric tensor field. Usually, an anisotropic tensor field can be defined from an error indicator or an error estimate that relates the approximation error for a size, shape and orientation prescriptions, see [BA76, VHM91, Ape99, FP01, DVB$^+$02, Hua05] and the references therein for examples. Theoretical analysis of this approach in our program of mesh adaptation have been detailed in [AF03, Fre08, FG08]. We only recall here some basic notations related to adaptation process in this thesis.

3.2.1 Metric tensor fields

Definition 3. *Let M be a metric tensor over $\mathbb{R}^d$, i.e at each point $x \in \mathbb{R}^d, M(x)$ is a symmetric, positive definite $d \times d$ matrix. The length of a curve $\gamma : [0,1] \to \mathbb{R}^d$, the volume $V_M(K)$ of a simplex K, and the distance $d_M(x,y)$ between two points $x, y \in R^d$ with respect to M an be defined as follows, respectively:*

$$l_M(\gamma) = \int_0^1 \sqrt{< M(\gamma(t))\gamma'(t), \gamma'(t) >} dt$$

$$V_M(K) = \int_K \sqrt{det(M(x))} dx$$

$$d_M(x,y) = \inf_{\substack{\gamma \in \mathcal{C}^1([0,1],\mathbb{R}^d) \\ \gamma(0)=x, \gamma(1)=y}} l_M(\gamma)$$

From the geometrical viewpoint, the metric defining an element can be represented by an ellipsoid whose volume, ratios between the lengths of semi-axes and the principal axis vectors are associated respectively with notations of the size, the shape and the orientation of the element. Indeed, given a metric tensor M, let T a unit mesh with respect to M. Given x_0 is a vertex of T such that $M(x) = M$ is almost constant around x_0, then every simplex K of T lying in the ball $B(x_0)$ (implies that K sharing x_0) is inscribed in the ellipsoid $\mathcal{B}_M(x_0)$) defined as:

$$\mathcal{B}_M(x_0) = \{x \in \mathbb{R}^d : d_M(x, x_0) = 1\} = \{x = \Sigma_{i=1}^{d} x_i e_i \in \mathbb{R}^d : \lambda_1 {x_1}^2 + \ldots + \lambda_d {x_d}^2 = 1\} \tag{3.2}$$

where $e_1, \ldots, e_d$ denote the normalized eigenvectors of $M(x_0)$ and $\lambda_1, \ldots, \lambda_d$ are the associated eigenvalues.

The set in (3.2) represents an ellipse (in 2D case) or an ellipsoid (in 3D) centered at x_0, its axes are the eigenvectors e_i and the length of the semi-axes are $1/\sqrt{\lambda_i}$ (see figure 3.4). Notice that when all

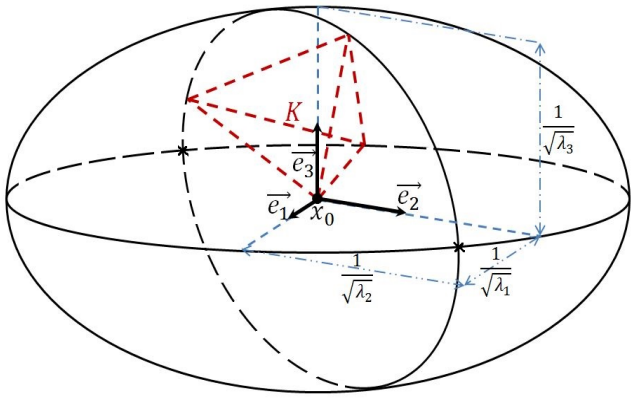


Figure 3.4: *Geometrical illustration of a unit ellipsoid $\mathcal{B}_M(x_0)$ with an inscribed element K associated to a metric tensor in 3D.*

the λ_i are equal, $M(x_0) = \lambda I_d$, where I_d is the identity matrix, then the ellipsoid $\mathcal{B}_M(x_0)$ becomes a sphere with radius $1/\sqrt{\lambda}$, thus it accounts for *isotropic* mesh. On the contrary, in case of *anisotropic* mesh, size, shape and orientation notions of an element are associated with the volume, the ratios between the lengths of semi-axes and the principal axis vectors of ellipsoids, respectively. In cases of very stretched elements may be desired which do not fulfill the standard quality requirements follow definition 3.1, these quality functions must be traded for their anisotropic counterparts, obtained by using the same expressions except that the distance and volume notions are those supplied by metric tensor M, see quality function in expression (3.5).

3.2.2 Metric definition based on interpolation error

We present in this section the determination of metric tensor such that the according anisotropic mesh guarantees a given interpolation error. This determination will be used to construct a metric tensor for adapted mesh in the scheme of two fluids flow (see section 3.2.6).
Classically, the approximation error is bounded from above by the interpolation error, thus it is natural to get an upper bound of the interpolation error. Let us now recall the L^∞ error estimate for the Lagrange finite element $\mathbb{P}^1$-interpolation in [FA05]. For a function $\varphi : \mathbb{R}^d \to \mathbb{R}$, its linear interpolation is denoted by $\Pi\varphi$, for all mesh element K we have following error estimate:

$$\begin{aligned} ||\varphi - \Pi\varphi||_{L^\infty(K)} &\leq c_d \max_{x\in K} \max_{\overrightarrow{v}\subset K} < \overrightarrow{v}, |\mathcal{H}(\varphi)(x)|\overrightarrow{v} > \\ &\leq c_d \max_{x\in K} \max_{\overrightarrow{e}\subset E_K} < \overrightarrow{e}, |\mathcal{H}(\varphi)(x)|\overrightarrow{e} > \end{aligned} \tag{3.3}$$

where $\mathcal{H}(\varphi)$ is the Hessian matrix of function 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 indicates that controlling the size of element edges allows to control the interpolation error. Given $\varepsilon > 0$ is desired tolerance, we aim at defining a metric tensor such that: $||\varphi - \Pi\varphi||_{L^\infty(K)} \leq \varepsilon$ for every element K in the adapted mesh. As suggested in [AF03], we define an anisotropic metric as follows:

$$M_\varphi = P^t \widetilde{\Lambda} P; \quad \widetilde{\Lambda} = \begin{pmatrix} \widetilde{\lambda_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \widetilde{\lambda_d} \end{pmatrix} \tag{3.4}$$

$$\text{with} \quad \widetilde{\lambda_i} = \min\left(\max\left(\frac{c_d|\lambda_i|}{\epsilon}, \frac{1}{h_{max}^2}\right), \frac{1}{h_{min}^2}\right)$$

where P is the eigenvector matrix and the coefficients λ_i are the eigenvalues of the Hessian matrix $\mathcal{H}(\varphi)$, h_{min} and h_{max} are the prescribed minimal and maximal edge size of the mesh. This metric construction is related to the Hessian of the exact solution φ, that is however not known. Therefore, a procedure for approximating the Hessian of the solution from the discrete solution is necessary, see [AF03].

3.2.3 Metric intersection

In several applications, we need to adapt a mesh at the same time to several priori independent informations, supplied by two (or more) metric tensor fields M_1, M_2 (for examples the mesh adaptation for two fluids flow, see below in section 3.2.6). This is classically achieved by a so-called *metric*

intersection procedure: operating on the simultaneous reductions of $M_1(x)$ and $M_2(x)$ at any point $x \in R^d$. Suppose that

$$M_1(x) =^t P(x) \begin{pmatrix} \lambda_1(x) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_d(x) \end{pmatrix} P(x) \text{ and } M_2(x) =^t P(x) \begin{pmatrix} \nu_1(x) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \nu_d(x) \end{pmatrix} P(x)$$

where $P(x)$ is an invertible matrix; $\lambda_i > 0, \nu_i > 0 (i = 1, \dots, d)$ then the *intersected metric* is determinated by:

$$M_1 \cap M_2(x) :=^t P(x) \begin{pmatrix} \max(\lambda_1(x), \nu_1(x)) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \max(\lambda_d(x), \nu_d(x)) \end{pmatrix} P(x)$$

Geometrically, this intersected metric implies that the ellipsoid $B_{M_1 \cap M_2}(x)$ is the largest one in-

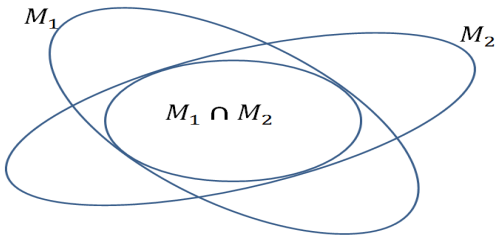


Figure 3.5: *Intersection of two metric tensors.*

scribed in both the ellipsoids $B_{M_1}(x)$ and $B_{M_2}(x)$ (see figure 3.5). More details can be found in [FG08].

3.2.4 The generation of anisotropic mesh

We assume that a metric tensor M is defined at the mesh vertices of a given simplicial T (note that in practice, $M(x)$ is defined only at the nodes of any background structure and interpolated from these values, see [FG08]). Several techniques have been devised for generating anisotropic meshes according to a metric tensor, such as Delaunay based triangulation methods (see [She12] for instances) and advancing-front methods where the construction of mesh elements is from the surface mesh of its boundary, see [FWE70, Geo71]. Another approach is the local mesh modification method which starts from an existing non-adapted mesh and adapt it so that it fits at best conditions, see [DF08] for a description of mesh procedure. In our strategy, the generation of an adapted mesh is

obtained using a Delaunay-based local mesh modification procedure (described in [DF08]) include following classical operations:

- splitting of mesh edges,
- collapsing of mesh edge endpoints,
- swapping of mesh edges or faces,
- vertex relocation.

Anisotropic mesh adaptation aims at modifying T iteratively by local operations in order to get a *uniform* (or a *quasi-uniform*) mesh with respect to this metric: in the desired mesh, all its simplices have edges lengths equal to 1 (respectively, lying in $[l_{min}, l_{max}]$) in the sense of Definition 3 and the *anisotropic quality* measure:

$$\mathcal{Q}_M(K) := \alpha \frac{V_M(K)}{(\Sigma_{i=1}^{na} l_M(e_i))^{d/2}} \tag{3.5}$$

of its simplices are as close to 1 as possible.

3.2.5 Explicit discretization of the interface

In the context of anisotropic level set adaptation, an interface Γ is approximated by a piecewise affine interface Γ_T such that $d(\Gamma, \Gamma_T) < \varepsilon$ on the adapted mesh T. We recall that Γ is defined basing on zero level set of function ϕ: $\Gamma_\phi = \{x \in \Omega : \phi(x) = 0\}$. For practical reasons, we need to modify T in order to obtain a new triangulation $\widetilde{T}$ in which $\Gamma_{\widetilde{T}}$ is explicitly discretized. This is achieved by inserting the set of intersection points of Γ_ϕ with the mesh edges/surfaces of T into the set of mesh vertices of T. The level set metric is updated on the resulting mesh and this mesh is then modified as previously to obtain final mesh $\widetilde{T}$ with respect to the updated metric.

3.2.6 Anisotropic mesh adaptation process for two-phase flow

We are now in position to construct an adapted mesh based on anisotropic mesh adaptation for the problem two fluid in section 2.5. In each time step, the initial mesh T have to solve two following problems:

1. Solving Navier-Stokes equation to have the vector field $\mathbf{u}$. This resolution is based on method of characteristic which is known that *unconditional stable* with the following error estimate, see [Pir82]:

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)} \leq c(h^m + \Delta t^m + h^{m+1}/\Delta t) \tag{3.6}$$

2. Find the new position of the interface Γ as zero iso-contour of level set function, solution to

advection equation:

$$\begin{cases} \dfrac{\partial \phi}{\partial t}(\mathbf{x},t) + \mathbf{u}(\mathbf{x},t)\nabla\phi(\mathbf{x},t) & = 0 \quad \forall (\mathbf{x},t) \in \Omega \times \mathbb{R}^+ \\ \phi(\mathbf{x},0) & = \phi^0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega \end{cases}$$

We aim to alter the initial mesh T to obtain a new mesh $\widetilde{T}$ such that:

i. The interface Γ is explicitly discretized the mesh $\widetilde{T}$.

ii. The geometry error of the zero-level set and its approximation in the context of Hausdorff distance is small enough, i.e $d^H(\Gamma, \Gamma_{\widetilde{T}}) < \varepsilon$ with ε is the desired error and:

$$d^H(\Gamma, \Gamma_{\widetilde{T}}) := \max\left(\sup_{x\in\Gamma} \inf_{y\in\Gamma_{\widetilde{T}}} |x-y|, \sup_{y\in\Gamma_{\widetilde{T}}} \inf_{x\in\Gamma} |x-y| \right)$$

iii. The mesh $\widetilde{T}$ is nice quality according to quality function defined in (3.5) and responds the resolution of the Navier-Stokes equation at the next iteration with small error approximation.

It can be interpreted that:

i. The first requirement is assured following the lines in section 3.2.5.

ii. The second requirement is implemented by the construction a metric tensor as the following: We have show in [CDF12] the following error estimates analysis for advection equation:

– The interpolation error of level set function and its numerical solution:

$$\|\phi - \phi_h\|_{L^\infty(\Omega)} \leq \|\phi - \Pi_h\phi\|_{L^\infty(\Omega)} + \|\phi_0 - \Pi_h\phi_0\|_{L^\infty(\Omega)} + c_1\|\mathbf{u} - \mathbf{u_h}\|_{L^\infty(\Omega)} + c_2 e^{\delta t}\delta t. \quad (3.7)$$

where c_1, c_2 are constants depending on initial datas ϕ_0 and velocity $\mathbf{u}$.

– The geometry error of zero-level set and its approximation in the context of Hausdorff distance:

$$d^H(\Gamma, \Gamma_h) \leq \sup\left(\frac{\sup_{x\in\Omega}\|\nabla\phi(x)\|}{\inf_{x\in\Omega}\|\nabla\phi(x)\|^2}, \frac{\sup_{K\in T_h}\|\nabla\phi_h(x)|_K\|}{\inf_{K\in T_h}\|\nabla\phi_h(x)|_K\|^2} \right) \|\phi - \phi_h\|_{L^\infty(\Omega)} \quad (3.8)$$

where: $d^H(\Gamma, \Gamma_h) := \max\left(\sup_{x\in\Gamma} \inf_{y\in\Gamma_h} |x-y|, \sup_{y\in\Gamma_h} \inf_{x\in\Gamma} |x-y| \right)$.

The estimations (3.7) and (3.8) permit us to control the Hausdoff distance $d^H(\Gamma, \Gamma_h)$ by the interpolation errors $\|\phi - \Pi_h\phi\|_{L^\infty(\Omega)}$, $\|\phi_0 - \Pi_h\phi_0\|_{L^\infty(\Omega)}$ and the approximation error $\|\mathbf{u} - \mathbf{u_h}\|_{L^\infty(\Omega)}$. In other words, the initial mesh is adapted such that these errors are as small as necessary. Otherwise, it is well-known that in case of approximation by Galerkin finite element, thanks to Cea's lemma the approximation error $\|\mathbf{u} - \mathbf{u_h}\|_{L^\infty(\Omega)}$ is bounded by the interpolation error $\|\mathbf{u} - \pi\mathbf{u}\|_{L^\infty(\Omega)}$. In summary, in order to perform a good discrete approximate Γ_h of Γ

the desired mesh should be adapted to interpolation of functions: ϕ, ϕ_0 and $\mathbf{u}$.

The last term in the right-hand side of (3.7) is related to substep δt, this means that the geometry error can be controlled by the adjusting of this substep.

In the section 3.2.3 we have detailed how to construct a metric tensor in such a way that the generated mesh satisfies the given interpolation error. In our scheme, we intend to construct only one mesh adapted to both Lagrange interpolation of function ϕ_0, ϕ and $\mathbf{u}$. Denoting M_{ϕ_0}, M_ϕ and $M_{\mathbf{u}}$ are respectively the resulting metric tensor according to these interpolations. By using *intersection metric* (see section 3.2.4) two times we obtain a tensor metric $M = (M_{\phi_0} \cap M_\phi) \cap M_{\mathbf{u}}$. The desired mesh is built according to metric tensor M.

iii. The third condition is responded rely on the estimation (3.6). It relates to characteristic mesh elements size h and the time step Δt. This implies that the approximation error of Navier-Stokes problem can be decreased by diminution h as well as chosen the appropriate time step Δt.

Hence, the final adapted mesh is an anisotropic mesh generated according to metric tensor M following the lines in section 3.2.4 with the appropriate characteristic element size h adjusted by $[l_{min}, l_{max}]$. See figure 3.6 for the generation of adapted mesh at each time step. This illustration is one step of the simulation of the rising bubble in 2D (section 4.2).

3.3 Discrete three dimensional domain remeshing

We now address the problem of isotropic three-dimensional domain remeshing which has been used in mesh adaptation of some numerical tests in 3D. As in the previous section, we keep always the notations Ω of continue domain and its three-dimensional simplicial mesh T associated with "surface triangulation" Γ_T which is an approximation of surface part Γ of Ω. The aim is to improve mesh T to obtain a resulting mesh $\widetilde{T}$ which is expected to be nice approximation of domain Ω follows some criterions, for examples:

- The surface triangulation $\Gamma_{\widetilde{T}}$ is the better approximation of Γ.
- $\widetilde{T}$ is well-shaped according an evaluating of quality mesh, for instances in this strategy we use the following nomalized quality mesh function element for any tetrahedron $K \subset \mathbb{R}^3$:

$$\mathcal{Q}(K) := \alpha \frac{V(K)}{(\Sigma_{i=1}^{6} l(e_i))^{3/2}}, \text{ with } \alpha = 144\sqrt{3} \tag{3.9}$$

where $e_i, i = 1, \ldots, 6$ are the edges of K and $l(e_i)$ stands for the length of e_i.

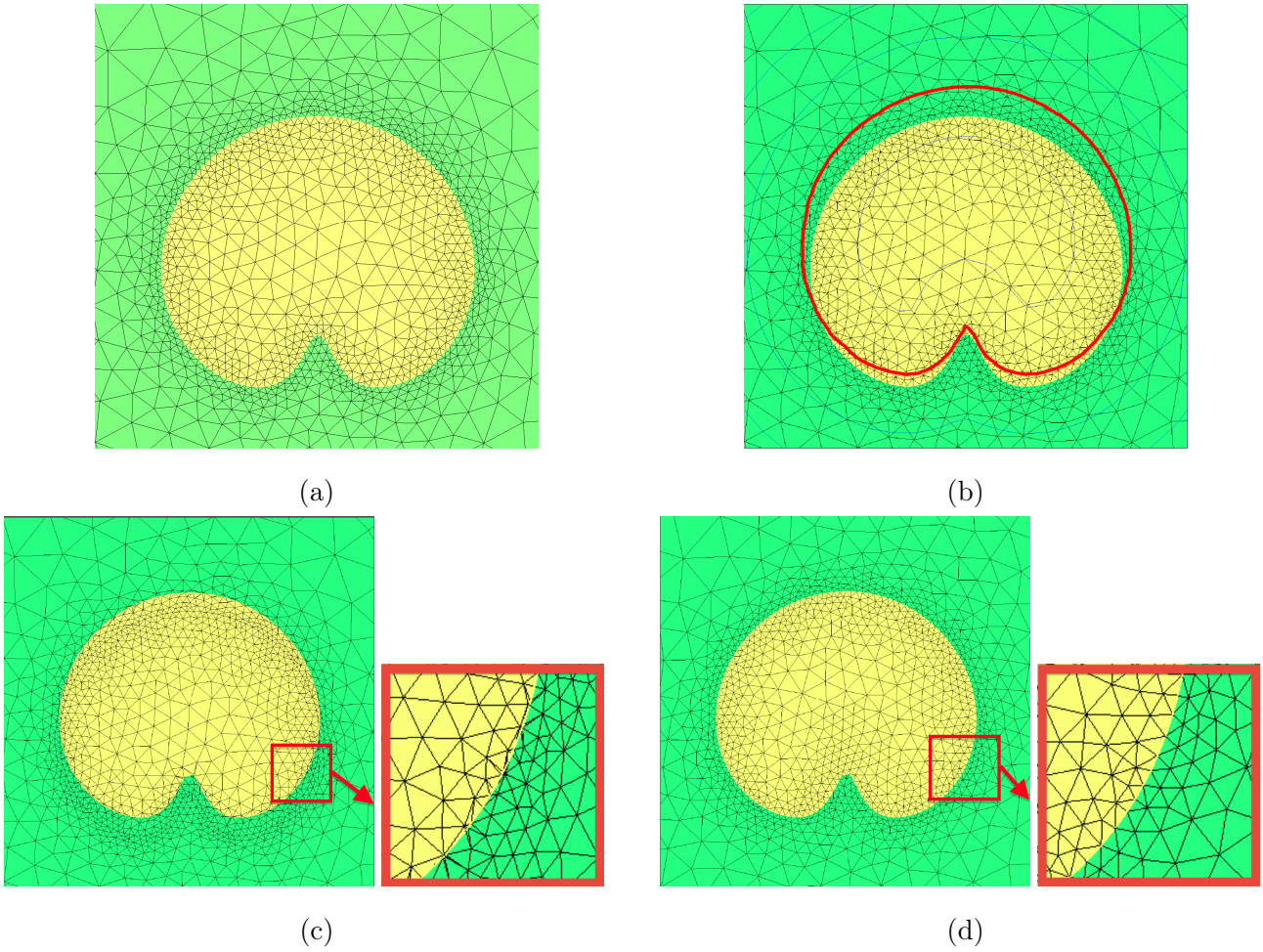


Figure 3.6: *Illustration for mesh adaptation in n^{th} iteration for two fluids flow. (a): Initial mesh T^n (used to obtain $\mathbf{u}^n$), (b): The new level set fuction ϕ^n with the red line is the zero-level set , (c): Explicit discretization of zero-level set of ϕ^n, the obtained mesh is not well-shaped, (d): High-quality adapted mesh in which the zero-level set of ϕ^n is explicitly discretized.*

.

3.3.1 The local size map

The remeshing process of T is based on the classical local remeshing operators (edge split, edge collapse, edge swap, and node relocation) which enjoy two forms, depending on whether they are applied to a surface configuration or to a purely internal one, see [FG08, DDF14]. Although, we need a global vision to drive the remeshing strategy, that is to identify (or classify) those edges of T that should be split, collapsed, or swapped. Since [VHM91, Fre00], a very convenient means to encode such information has been rely on a size map $h : \Omega \to \mathbb{R}$ such that for any $x \in \Omega$, $h(x)$ accounts for the local desired size for the edges of T surrounding x. The final aim of the process is then to produce a new mesh $\widetilde{T}$ of Ω, whose edges pq have (as far as possible) unit length $l_h(pq)$ respect to h, where:

$$l_h(pq) = \int_0^1 \frac{|pq|}{h(p + t(q - p))} dt \tag{3.10}$$

In numerical practice, h is defined and stored at the vertices of T then interpolated from these data whenever needed elsewhere.

3.3.2 Map size determination adapted to the geometric approximation

This section will present the definition of size map h such that the adapted mesh according to h satisfies the desired geometry error of a given surface.

Consider T is a conforming tetrahedral mesh of a continue domain Ω with the interface surface Γ. Let denote Γ_T is approximation of Γ. Our attempt to modify T into a new mesh $\widetilde{T}$ in which the approximating interface surface $\Gamma_{\widetilde{T}}$ of surface Γ is demanded should be close to Γ up to a given tolerance ε, that is:

$$d^H(\Gamma_{\widetilde{T}}, \Gamma) \leq \varepsilon \tag{3.11}$$

where $d^H(\Gamma_{\widetilde{T}}, \Gamma)$ is Hausdorff distance between $\Gamma_{\widetilde{T}}$ and Γ.

In practice, $\widetilde{T}$ will be generated rely on the information of the size map h which is defined as follows:

i. *at a surface vertex* $x \in \Gamma_T$, the size prescription $h(x)$ in a neighborhood of x based on the following theorem (see [Dap13] for a proof, Theorem 8.2):

Theorem 1. *Let* $\Omega \in \mathbb{R}^d$ *a domain, and* T *a mesh, whose associated surface mesh* Γ_T *is "close" from* Γ. *Denote as* d_Γ *the signed distance function to* Γ, *and* $\mathcal{H}(d_\Gamma)$ *its Hessian matrix. Then*

$$d^H(\Gamma, \Gamma_T) \leq \frac{1}{2}\left(\frac{d-1}{d}\right)^2 \max_{K \in \Gamma_T} \max_{x \in K} \max_{\overrightarrow{e} \subset E_K} \langle \overrightarrow{e}, |\mathcal{H}(d_\Gamma)(x)| \overrightarrow{e} \rangle \tag{3.12}$$

Hence, to satisfy (3.11) it is enough to ask that, for each triangle $K \in \Gamma_T$, one has:

$$\frac{2}{9}\max_{x\in K}\max_{\overrightarrow{e}\subset E_K}\langle\overrightarrow{e},|\mathcal{H}(d_\Gamma)(x)|\overrightarrow{e}\rangle \tag{3.13}$$

Now, let $K \in \Gamma_T$ be a triangle which is "very close" to a point $x \in \Gamma$, so that we can assume $\mathcal{H}(d_\Gamma)$ is nearly constant around K. An approximation of the latter sufficient condition is then:

$$\forall\overrightarrow{e}\subset E_K,\ \frac{2}{9}\langle\overrightarrow{e},|\mathcal{H}(d_\Gamma)(x)|\overrightarrow{e}\rangle \leq \varepsilon$$

On the other hand, introducing $\kappa_i(x), (i = 1, 2)$ the two principal curvatures of Γ at x and $e_i(x)$ the two associated principal directions, the matrix $\mathcal{H}(d_\Gamma)(x)$ writes in the orthonormal basis $(e_1(x), e_2(x), n(x))$ of $\mathbb{R}^d$:

$$\mathcal{H}(d_\Gamma)(x) = \begin{pmatrix} \kappa_1(x) & 0 & 0 \\ 0 & \kappa_2(x) & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.14}$$

Then, we have the upper bound $\sqrt{\frac{9\varepsilon}{2\max(|\kappa_1(x)|,|\kappa_2(x)|)}}$ of any edges of K.
We are led to the definition of metric size map h on the surface as $h : \Gamma \to \mathbb{R}$

$$\forall x \in \Gamma, h(x) = \min\left(h_{max}, \max\left(h_{min}, \sqrt{\frac{9\varepsilon}{2\max(|\kappa_1(x)|, |\kappa_2(x)|)}}\right)\right) \tag{3.15}$$

where h_{min} and h_{max} are respectively lower and upper bounds on the authorized lengths of edges in T.

ii. *at an internal point x,* no constraint from the geometry, i.e we only need apply the maximal authorized size $h(x) = h_{max}$ for an edge of the resulting mesh $\widetilde{T}$.

Remark 5. *The size map h defined above only accounts for the local size feature associated to the geometric approximation of surface Γ. In many case, the local size feature need to adapted another size map $k : \Omega \to \mathbb{R}$ arise from a posteriori error analysis associated to some numerical resolution of a problem (see the requirement of mesh adaptation in chapter 2 of this thesis or the problem of [AO97] for instances). In the problem of two fluids flows, the local size map need to adapted size map k arise from the error analysis associated to numerical resolution of Navier-Stokes equation problem. In such a case, this additional information is taken into account by trading h for a new size map $\widetilde{h}$ defined as: $\forall x \in \Omega : \widetilde{h}(x) = min(h(x), k(x))$.*

3.3.3 Gradation of the size map

Unfortunately, conforming to the size prescription discussed above is not sufficient in itself to guarantee the resulting mesh will enjoy a nice mesh quality. Indeed, the computed size map h may vary very sharply from one point to one of its neighbors, because the computation of h suffered from noise one the input data Γ_T, or because the ideal surface Γ itself shows sharp variations of curvatures. This shock of size prescriptions between close areas may urge the formation of undesired ill-shaped elements during the remeshing process. For this reason, it may be desirable to drive the remeshing operators in such a way that the resulting triangulation Γ_T from the process shows smooth variations in edge lengths. More accurately, we expect the resulting mesh to be such that two edges ap and bp sharing a common vertex p have Euclidean lengths satisfying:

$$\frac{1}{r} \leq \frac{|b-p|}{|a-p|} \leq r \tag{3.16}$$

where r is a user-defined bound (typically, we use $r = 1.1, 1.2, 1.3, 1.4, 1.5$).
To achieve this, the seminal work [BFH98] proposed to enforce a gradation on the size map h. More precisely, once the size map h has been computed in such a way that:

$$\forall \text{ edge } pq \in T, \frac{|h(p) - h(q)|}{|p-q|} \leq h_{grad} \tag{3.17}$$

This criterion is imposed on the values of h stored at the vertices of T by traveling repeatedly the edges $pq \in T$ and decreasing the largest of the two values $h(p)$ and $h(q)$ so that (3.17) holds.

3.3.4 Explicit discretization of the zero level set

As mentioned, in our approach we need an explicit discretization of the zero level set of function ϕ: $\Gamma_\phi = \{x \in \Omega : \phi(x) = 0\}$ into mesh T. This is achieved through the following *marching tetrahedra* procedure, which is a well-known variation of the marching cubes algorithm, see [LC87, DK91, FB96]:

i. Identify the set $\mathcal{K}$ of elements $K \in T$ intersecting Γ_ϕ: a tetrahedron $K = (P_0P_1P_2P_3)$ belongs to $\mathcal{K}$ if and only if there exists $i \neq j \in \{0, 1, 2, 3\}$ with $\phi(P_i) \leq 0$, and $\phi(P_j) \geq 0$.

ii. For an element $K = (P_0P_1P_2P_3) \in \mathcal{K}$, the intersection of $\Gamma_\phi \cap K$ is a plane portion of surface. Identify the edges P_iP_j of K which intersect Γ_ϕ (i.e. such that $\phi(P_i)$ and $\phi(P_j)$ have different signs), and compute the coordinates of the associated intersection points M_{ij}.

iii. Travel all elements $K \in \mathcal{K}$, and split them, introducing the pre-computed points M_{ij} then using splitting patterns. Up to permutations, there are four possible configurations, depending on the relative signs of the $\phi(P_i)$

This procedure creates a new conforming mesh $\widetilde{T}$ of T but it is likely very ill-shaped since the intersections of the elements of T with Γ_ϕ are quite arbitrary.

3.3.5 The complete adapted mesh strategy for two-phase flows in 3D

Now, starting from an initial simplicial mesh T, with associated surface triangulation Γ_T, and given the four parameters $h_{hausd}, h_{min}, h_{max}$ and h_{grad}, with $h_{min}, h_{max}, h_{grad}$ have the same meaning as in previous subsections and h_{hausd} is the desired geometry error between the interface and its approximation in the context of Hausdoff distance. The proposed remeshing algorithm reads as follows:

i. The explicit discretisation of the interface is completed following the lines in the section 3.3.4, hence we obtain a new mesh $\widetilde{T}_1$ may be still of poor quality.

ii. *Construction of the size map.* Although it may still be of poor quality, the new mesh $\widetilde{T}_1$ accounts for a suitable discretisation of the geometry of Γ. A size map $h : \Omega \to \mathbb{R}$ dedicated to the geometric approximation of within tolerance h_{hausd} in terms of Hausdorff distance is computed as the section 3.3.2 and stored on $\widetilde{T}_1$. This size map is then graded as detailed in section 3.3.3.

iii. *Mesh modifications with respect to the size map.* In this step, mesh modifications aims at producing another intermediate mesh $\widetilde{T}_2$ of Ω which is a nice geometric approximation, with respect to the prescribed tolerance: $d^H(\Gamma_{\widetilde{T}_2}, \Gamma) < h_{hausd}$. Furthermore, we rely now on lengths measured with respect to map size h by formula (3.10). Aiming at getting a new mesh whose edges have length 1, we impose that all the edges of the mesh should lie in $[l_{min}, l_{max}]$ where l_{min}, l_{max} are rough bounds around the target size 1.

iv. *"Fine" mesh modifications with respect to the size map.* Mesh $\widetilde{T}_2$ should now be good in terms of geometric approximation of Γ and of better quality, and in this last stage, we perform delicately driven operations so as to get the final mesh $\widetilde{T}$. Lengths of edges are still evaluated with respect to h, except we now impose $\widetilde{T}$ have no edge with length lying outside a sharper interval as in step iii. We are also even stricter as far as the authorized degradation in mesh quality entailed by our operators is concerned. We eventually use the vertex relocation operator to help improving the quality of the mesh.

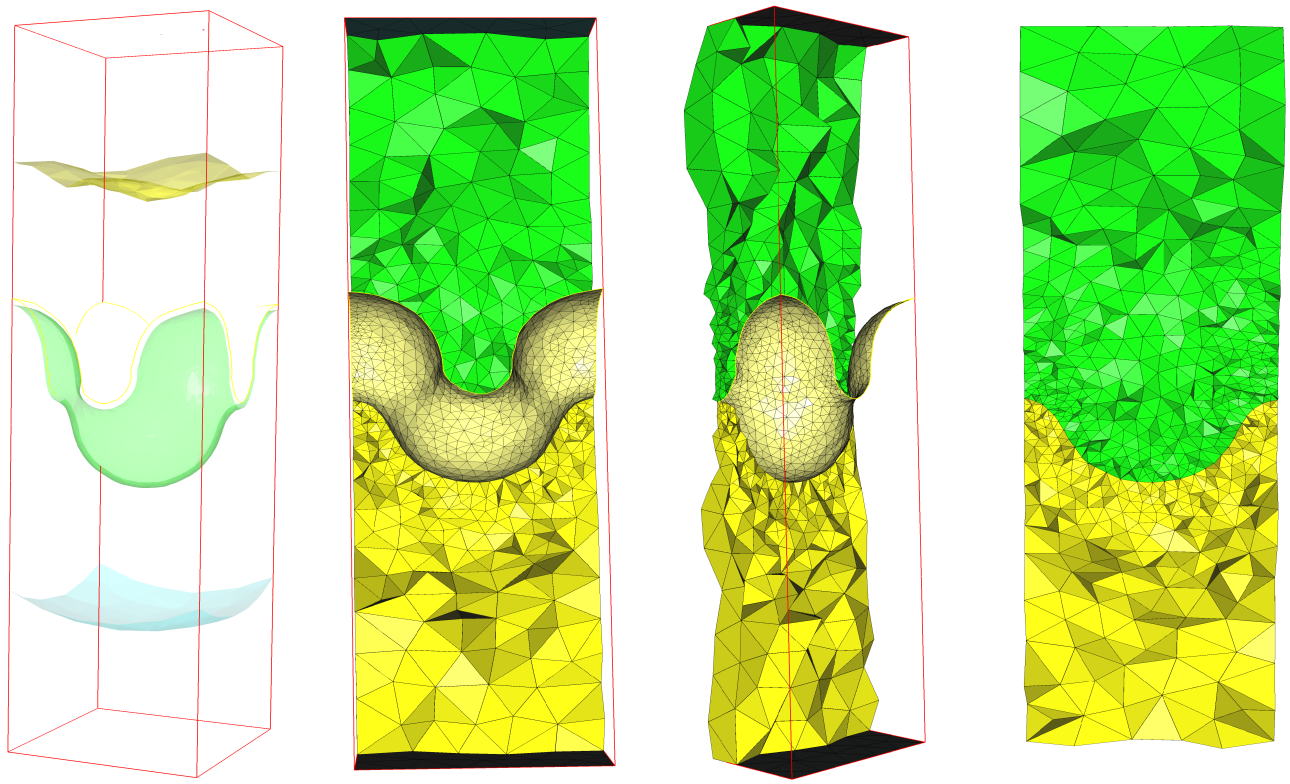
It is clear that the final resulting $\widetilde{T}$ responds all the requirements of an adapted mesh in each time step for resolution of two-phase flows in section 2.5:

– The explicit discretization of the interface in $\widetilde{T}$.

– The nice geometrical approximation for the interface as well as the desired error approximation of the velocity.

– The quality assessment of mesh elements.

For illustration, we show in the figure 3.7 the generation of adapted mesh in one iteration of the simulation of Rayleigh Taylor instability in 3D (detailed in section 4.3)



(a) (b) (c) (d)

Figure 3.7: *Illustration for mesh adaptation in 3D* ($h_{min} = 1.e-2, h_{max} = 0.2, h_{grad} = 1.3, h_{hausd} = 5e-3$). *(a): isosurfaces of level set function, (b) and (c): Final adapted mesh where zero-surfaces is explicitly discretized , (d): a cut of final mesh.*

.

3.4 Conclusion

We have presented in this chapter some basic features of mesh adaptation. Together with chapter 2 our approach for the bifluid flows simulation with interface is fully described.

This chapter completes the theoretical presentation of the first part which contains two main problems:

1. Numerical resolution of the Navier-Stokes equation for single viscous fluid. (chapter 1)
2. Numerical simulation of two-phases flow. (chapter 2 and chapter 3).

The next chapter is devoted to the numerical application to emphasize the efficiency of the proposed numerical schemes for both single fluid and two fluids flows.

Chapter 4

Numerical examples

Contents

In this chapter, we present several numerical results obtained with our method. Firstly, the Navier-Stokes solver for monofluid has been valided by the Lid-driven cavity test. Next, the test of a rising bubble and an instability Rayleigh-Taylor are investigated to evaluate the ability of the scheme of two fluids problem. The results in both two dimensions (2D) and three dimensions (3D) of these simulations are given in comparison with some results in other references.

4.1 The Lid-driven cavity problem

The Lid-driven cavity problem is known as standard benchmark for Navier-Stokes solver in numerical methods and there is also a great deal of reference to compare with. The problem corresponds to the flow confined in the unit domain $\Omega = [0,1]^d (d = 2;3)$ (a domain in three dimension is given by extending the two dimensional one in z-direction with a unit width) and the homogeneous 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.

4.1.1 Two-dimensional lid-driven cavity

In two dimensions we investigated the simulations for Reynolds number 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; other uniform triangulation (carre4) with 8421 nodes,16544 elements; and the last one is a regular triangulation (carre7) with 10201 nodes, 20000 elements for the test of high Reynolds numbers ($Re = 10000$). This problem involves a primary vortex at the cavity center and the vortices at the corners as Re increases. It is known that in the increasing of Reynolds number the number of vortices increases and the position of the center of primary vortex has the tendency to move from the bottom right corner towards the center of cavity. In the table 4.1, we resume the positions of the center of 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 the figure 4.1 for different Reynolds numbers are in good accordance with those in many references, for examples [UG82, GM99, ABPV12, APV14]. We also compute the profiles of velocity along horizontal and vertical lines passing the geometric center of cavity, see figure 4.2. 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.

We have also obtained a good agreement of the pressure solution with the result showed in [HRK$^+$10], see figure 4.3.

4.1.2 Lid-driven cavity in 3D

Two mesh are employed to simulate 3D problem: the coarse one consists of 11037 vertices, 56244 tetrahedrals for the cases of $Re = 100$, $Re = 400$ and the other one consists of 35723 vertices, 193586

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 4.1: *Cavity in 2D: comparison of the positions of the main vortex for different Reynolds.*

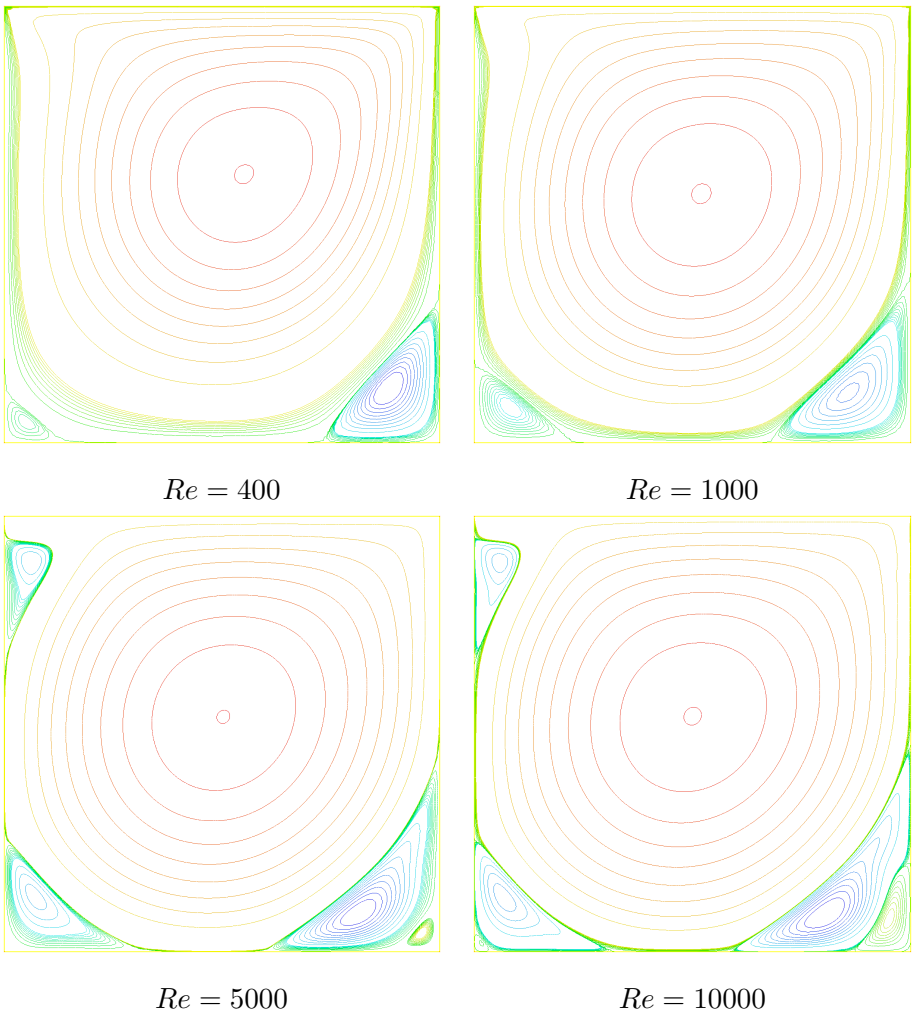


Figure 4.1: *Cavity in 2D: Streamlines for different Reynolds number.*

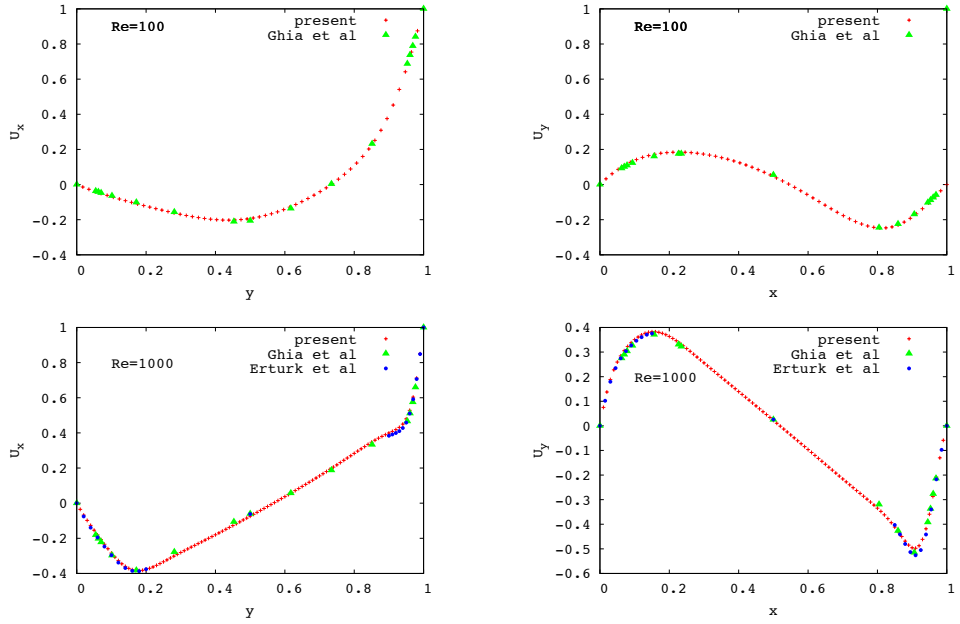


Figure 4.2: *Cavity in 2D: velocity profile for u_x and u_y in cases of Re=100 and Re =1000.*

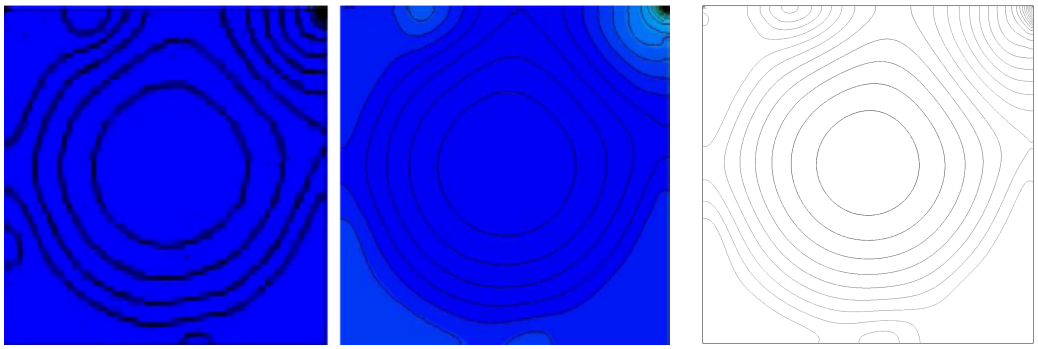


Figure 4.3: *Cavity in 2D: Isolines of pressure with Re = 10000. Left: result in [VG04]. Center: result in [HRK+10]. Right: present result.*

tetrahedrals for $Re = 1000$. In many references the results in 3D is examined by plane in 2D, so we simulate 3D problem with the same number of Reynolds in 2D: $Re = 100$, $Re = 400$, $Re = 1000$. As expected, we obtain the streamlines in the each plane $z = const$ are correspondent to those in 2D, see the figure 4.4 for example.

We would like to compare our numerical solutions with the results in [KHT87] because the Reynolds

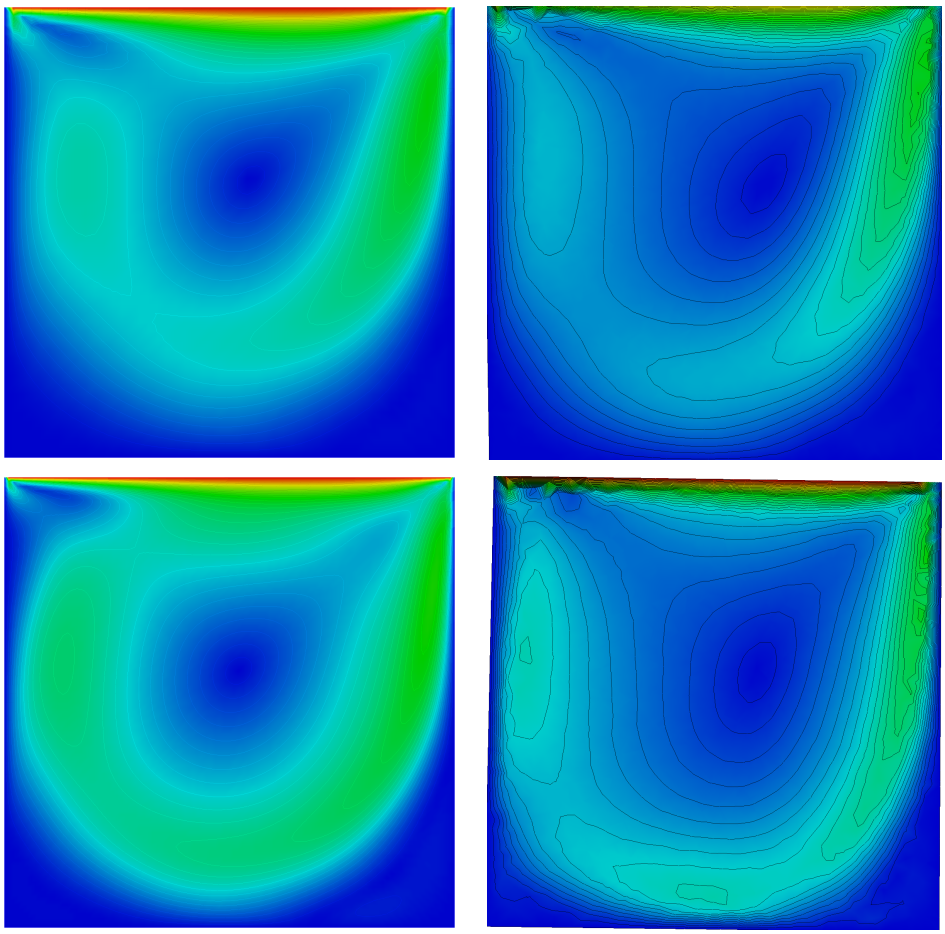


Figure 4.4: *Cavity in 3D: from left to right, streamlines in 2D and in the plane ($z = 0.5$) of 3D for Re = 400 (top), Re = 1000 (below).*

numbers are exactly the same in our tests, but the data of velocity profiles has not been detailed there, so the comparison is on the images, see figure 4.5 for a good agreements of our velocity profiles in 2D, 3D and those obtained in given reference.

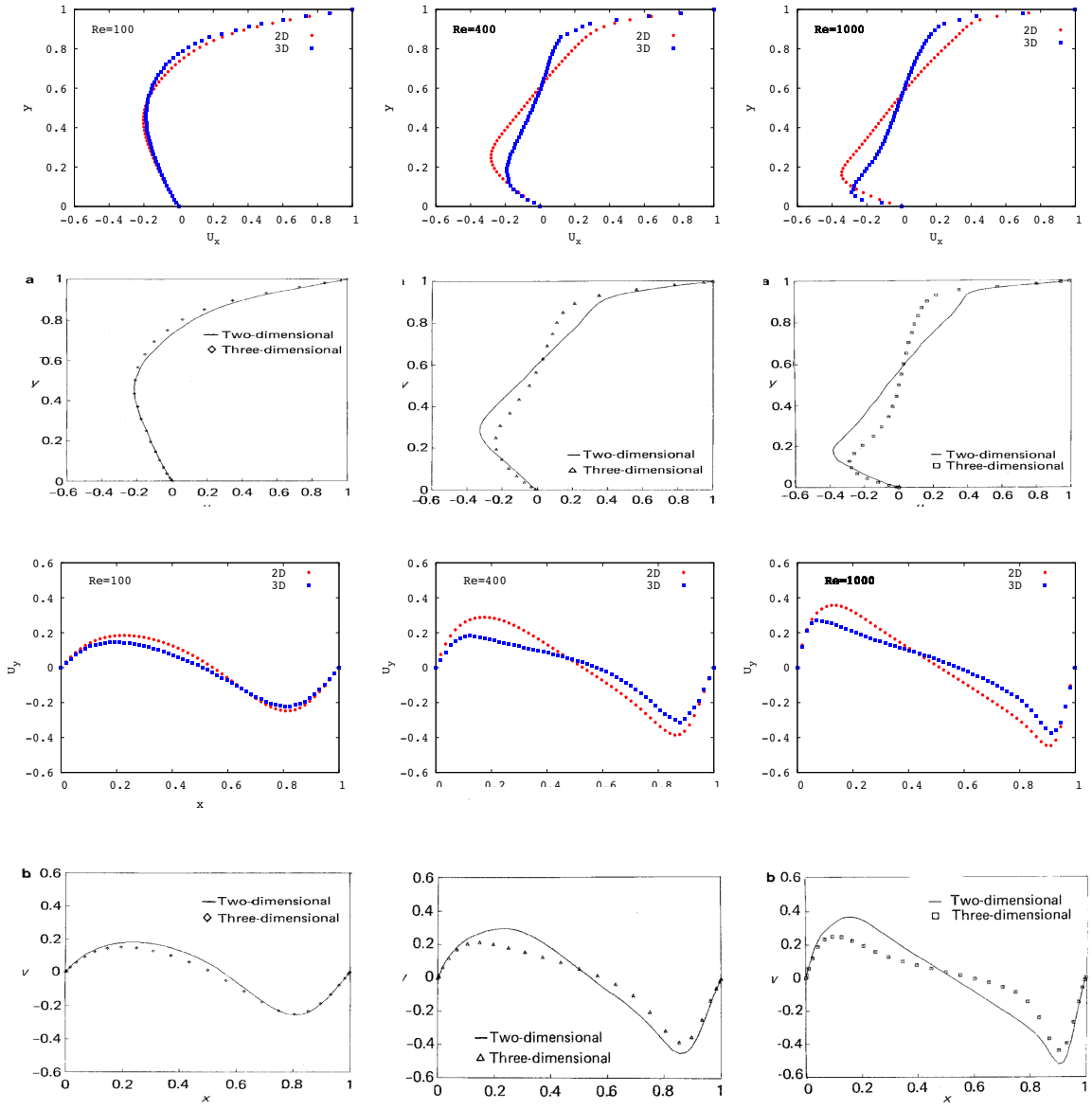


Figure 4.5: *Cavity in 3D: velocity profiles on vertical centerline. The first and third rows: present results. The second and fourth rows: results obtained in [KHT87].*

4.2 Rising bubble

We consider the rising and the deformation of single bubble under gravity in fluid contained in a vertical, rectangular domain. The bubble with lower density than the surrounding fluid rises and final at the top of the domain.

4.2.1 Rising bubble in 2D

The initial configuration consists of a circular bubble of radius $r = 0.5$ centered at $[2, 1.5]$ in a $[4, 10]$ domain of 2494 nodes. For the boundary condition, we impose no-slip condition ($\mathbf{u}$=0) on the horizontal walls and free-slip ($\tau.\sigma\mathbf{n} = 0$ and $\mathbf{u}.\mathbf{n} = 0$) on the vertical walls. See Fig 4.6 for the illustration of initialization and boundary conditions.

In many references, the different simulation is classified by Reynolds number and Bond number

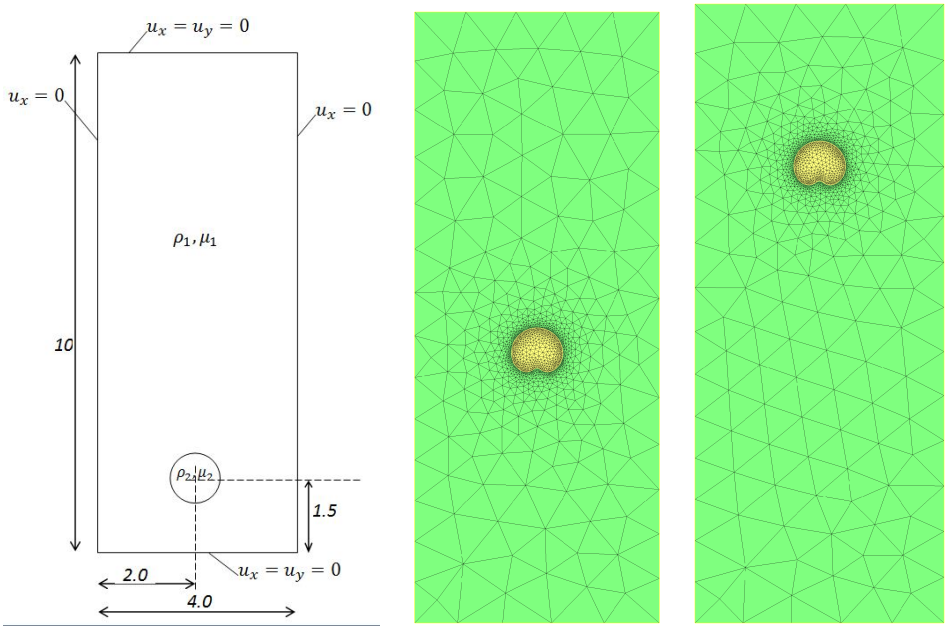


Figure 4.6: *Rising bubble in 2D: initialisation (left) and evolution in time: t = 5.0s (center), t = 10.0s (right).*

(also known as Eotvos number) defined as follows:

$$Re = \frac{\rho_1 \sqrt{g}(2r)^{3/2}}{\mu_1}, \quad Bo = \frac{4\rho_1 g r^2}{\gamma} \tag{4.1}$$

The problem has been set up with the constant densities and viscosities are: $\rho_1 = 100\text{kg.m}^{-3}, \mu_1 = 0.1\text{kg.m}^{-1}.s^{-1}, \rho_2 = 1.0\text{kg.m}^{-3}, \mu_2 = 0.01\text{kg.m}^{-1}.s^{-1}$. The gravity is: $g = 9.81e-3\text{m.s}^{-2}$. The evolution of the bubble with the coefficient of surface tension $\gamma = 6.e-3N.m^{-1}$ and adaptive meshes

with $h_{min} = 0.02, h_{max} = 1.0$ is showed in the figure 4.6. It can be seen that the bubble shape deforms during rising and the terminal bubble shape is slightly dimpled at the bottom.

In order to impress the effect of surface tension, we investigate this simulation with different coefficients of surface tension. It can be seen in the figure 4.8 when surface tension is rather small here ($\gamma = 6e-5$), the bottom of the bubble becomes more dimpled while it is flat in cases of more important coefficient ($\gamma = 2.5e-2$) and the bubble remains almost circular with the further increase in this coefficient ($\gamma = 9.e-2$). This result of bubble shapes is in good agreement with figure 6 of [HSL07] in cases of low Reynolds number and Bond number is from 10 to 200 extracted in figure 4.8, corresponding with our differences of coefficient surface tension γ. (see figure (4.1))

We have known that during the long simulation, the mass of bubble can't be guaranteed. The

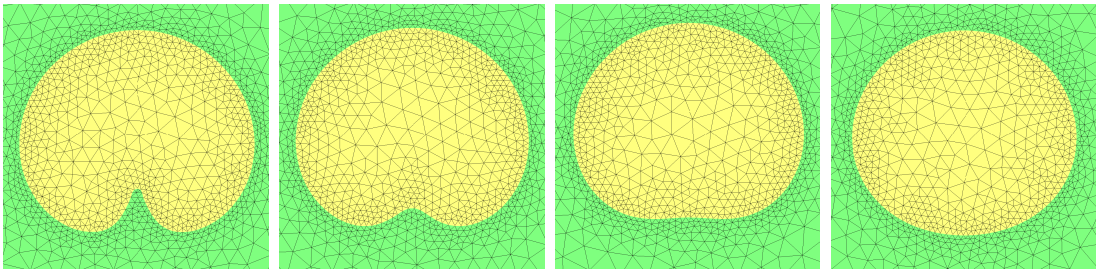


Figure 4.7: *Rising bubble in 2D: interaction of surface tension on the final bubble (at time t=10s) with different tension coefficients, from left to right:* $\gamma = 6e-5$*;* $\gamma = 6e-3$*;* $\gamma = 2.5e-2$*;* $\gamma = 9e-2$.

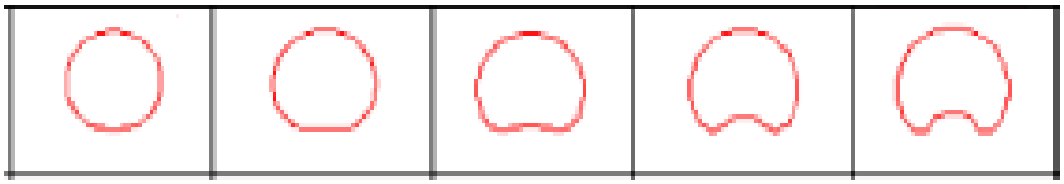


Figure 4.8: *Rising bubble in 2D: interaction of surface tension in case of Re = 5 and from left to right: Bo = 10; Bo = 20 ; Bo = 50; Bo = 100; Bo = 200 in [HSL07].*

advection of bubble front and the interpolation in the each iteration may lead to the loss of mass. We measure the variation of mass in each time step and show in the figure 4.9. The variation of mass can be seen as about $2.E-3(0,1\%)$ in each time step.

4.2.2 Rising bubble in 3D

Extending to simulate bubble rising under gravity in 3D. We consider the problem with the same conditions in 2D: a bubble with diameter of 0.5m initialized at $[0.75, 0.75, 1.0]$ in the domain

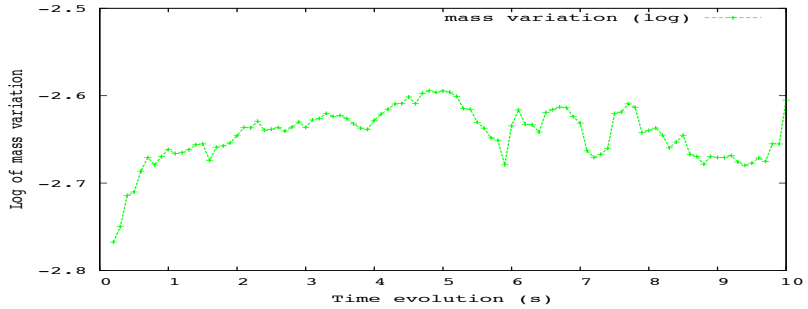


Figure 4.9: *Rising bubble in 2D: variation of mass correction with time evolution.*

of $[0, 1.5] \times [0, 1.5] \times [0, 4.5]$. The aims of this simulation is validation our code of two-phase fluid in 3D by examination the bubble shapes during the evolution and the correction of volume in each iteration. As the simulation with low Reynolds number, we observe that the shape of bubble deforms slowly from the beginning, it becomes dimpled ellipsoidal and more distorted in time. This result is similar to those in many references with corresponding Reynolds and Bond number, see [LAB10], [HSL08] for examples. In figure 4.10 we represent the evolution of bubble from $t = 0$ to $t = 10$. It can be seen that when the bubble is very close to upper wall of the domain, its shape is rapidly distorted.

4.3 Rayleigh-Taylor instability

In this section, we carry out the simulation of more interesting problem, named Rayleigh-Taylor instability. This instability would occur along the interface of two layers where the heavy fluid is superposed the light one ($\rho_2 > \rho_1$) in the gravity field g. Starting from the results concerning the inviscid flow of [Try88], this problem has been considered by many references for viscous regime, see [FGQ01] and[CCG08], for examples.

In our knowledge, the difficulty of the problem concerning viscous fluid depends on many criterions, among those we impress two 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:

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

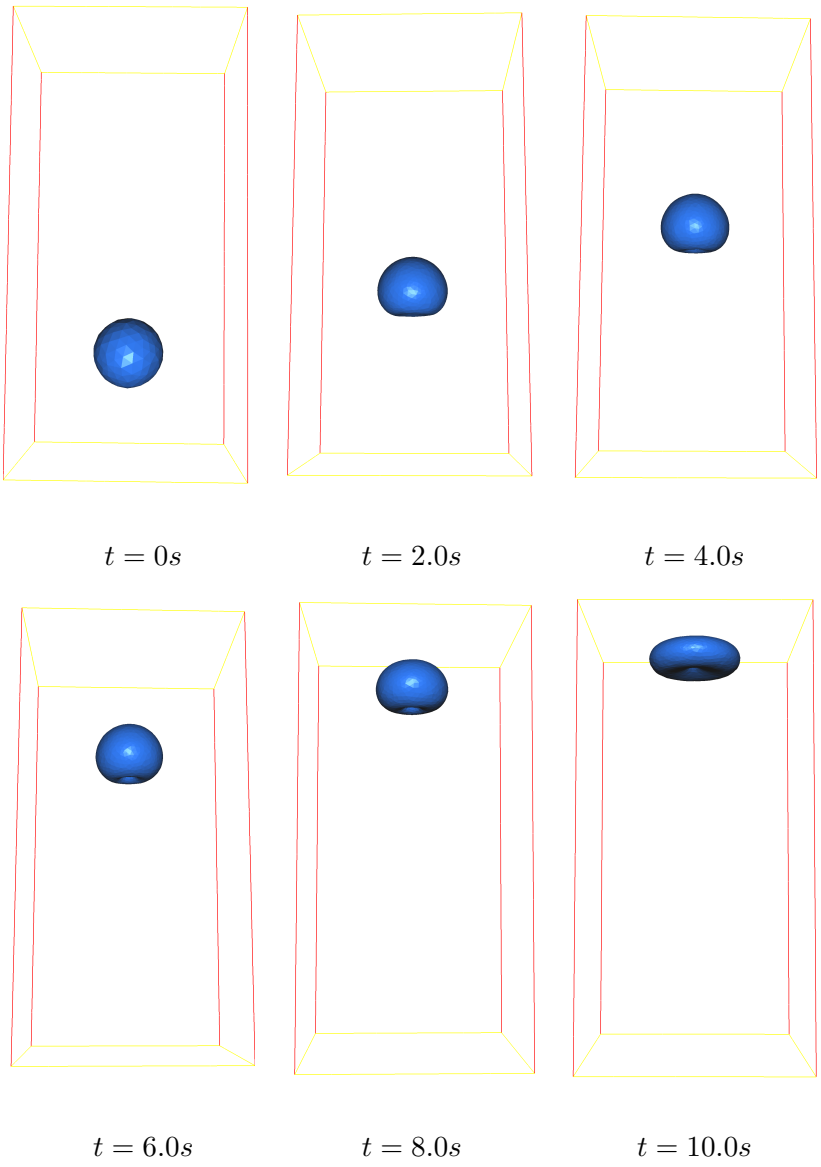


Figure 4.10: *Rising bubble in 3D: evolution of the interface in time.*

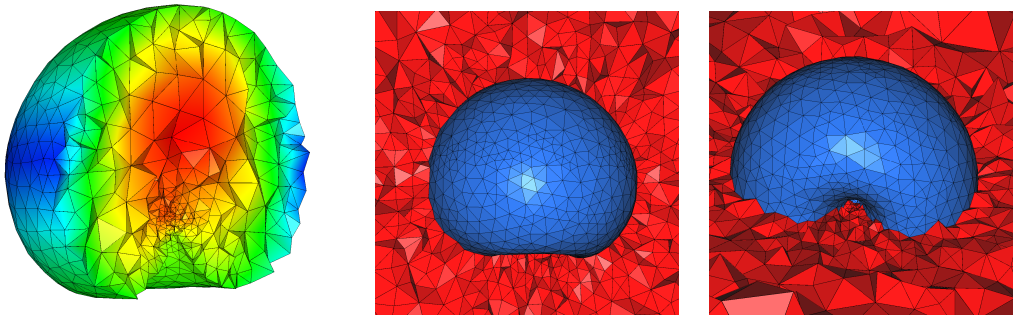


Figure 4.11: *Rising bubble in 3D: zoom of and adapted mesh at time $t = 6.0s$.*

where d is the width of the computational domain.

4.3.1 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 as in by:

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

We obtain the results is in good symmetry during the time evolution with the Atwood number is 0.3. The results display in figure 4.12 can be compared with figure 8 of [LKK10], notice that the mushroom shapes are not identical because the given comparison showed the simulation with $At = 0.5$ and higher Reynolds number. We will investigate this case of the test in reduced domain in which the computational time is much cheaper and we can also verify boundary conditions for this type of domain as in [FGQ01].

Figure 4.13 shows the results according to different Atwood numbers at the same Reynolds number.

Figure 4.12: *Rayleigh-Taylor instability in 2D: evolution of the interface in time with* $At = 0.3$.

We can observe that the mushroom shape is more roll-up in the increasing of Atwood number. This results suggest the effect of this number on the ratio of the width of bubble and spike fluid is in good agreement with the measuring played in the figure 9 in [LKK10].

Figure 4.13: *Rayleigh-Taylor instability in 2D: evolution of the interface with different Atwood numbers.*

Secondly, assuming the symmetry of the initial condition is maintained during the time evolution, we consider the problem with the reduced domain with a width of $d/2$ and a height of $2d$. We see again the same configurations with different Atwood numbers on the reduced domain in figure 4.14

As mentioned, we also validate our code by the same value of the test in [FGQ01]. The evolution of the interface is plotted in figure 4.15 in time scale according to [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], also in [CCG08], there are in good agreement of the global characteristic of the flow in the early stage and some slight discrepancies at the large times of the evolution.

4.3.2 Rayleigh-Taylor instability in 3D

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 the proposed scheme. The results performed in the figure 4.18 are very close to the results in figure 7 of [LJG96], or figure 18 of [LFX05]. The slight difference of geometric configurations is the reason that our fluids are incompressible while the comparisons are compressible. On the other hand, the density ratio is not identical, so more detailed comparison with these results may be not interesting.

In this simulation, we begin with the mesh of 9.901 vertices (50429 tetrahedras) and the final

Figure 4.14: *Rayleigh-Taylor instability in 2D: evolution of the interface with different Atwood numbers and reduced domain.*

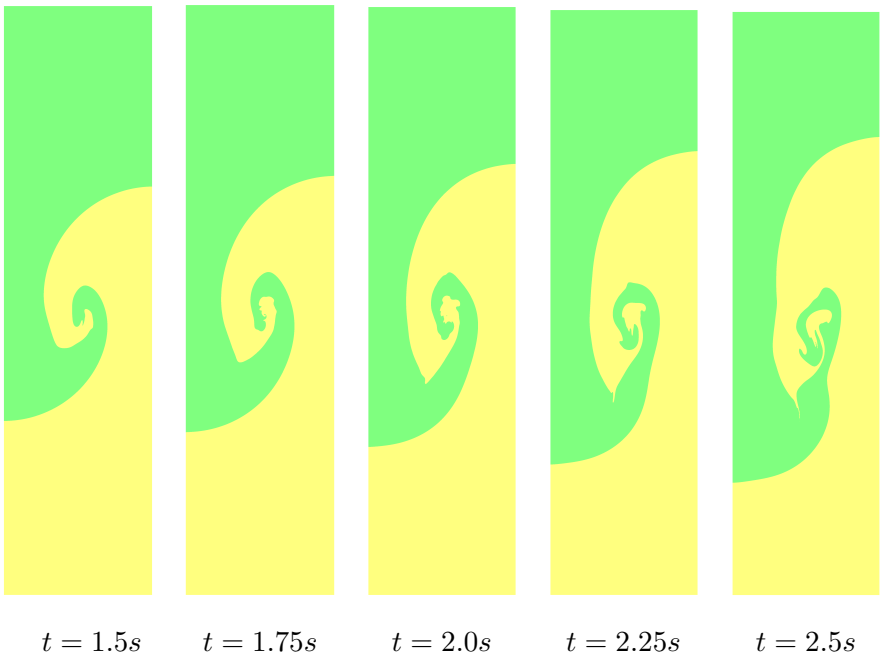


Figure 4.15: *Rayleigh-Taylor instability in 2D: evolution of the interface with At = 0.5, Re = 1000.*

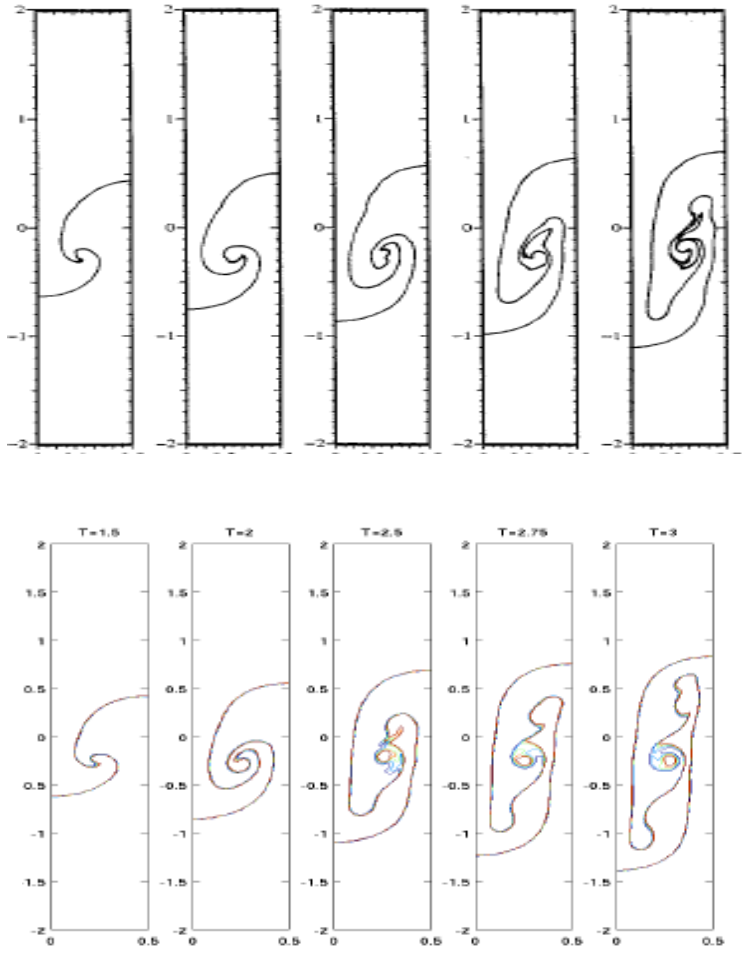


Figure 4.16: *Rayleigh-Taylor instability in 2D: extracted results of evolution of the interface with At = 0.5, Re = 1000 in [FGQ01] (top) and [CCG08] (bottom).*

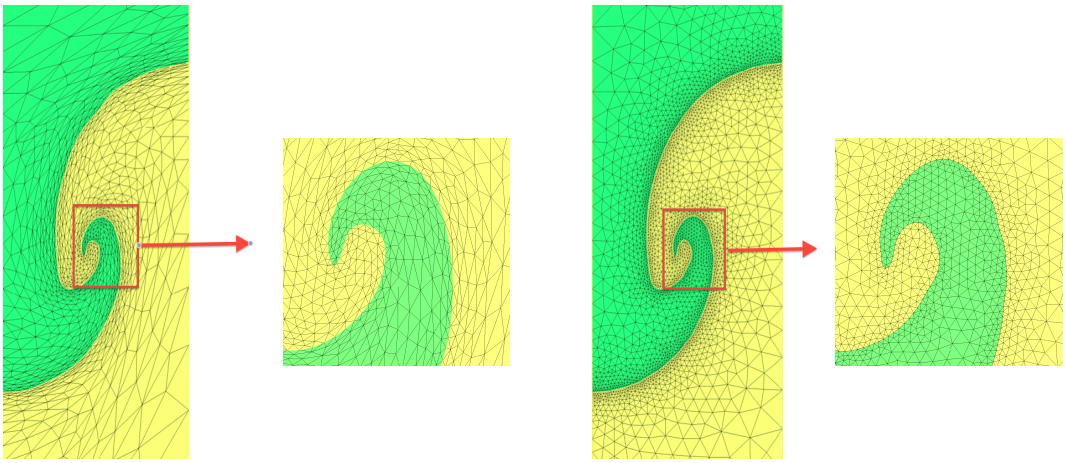


Figure 4.17: *Rayleigh-Taylor instability in 2D: zoom of adapted mesh in the vicinity of the interface, left: anisotropic mesh, right: isotropic mesh.*

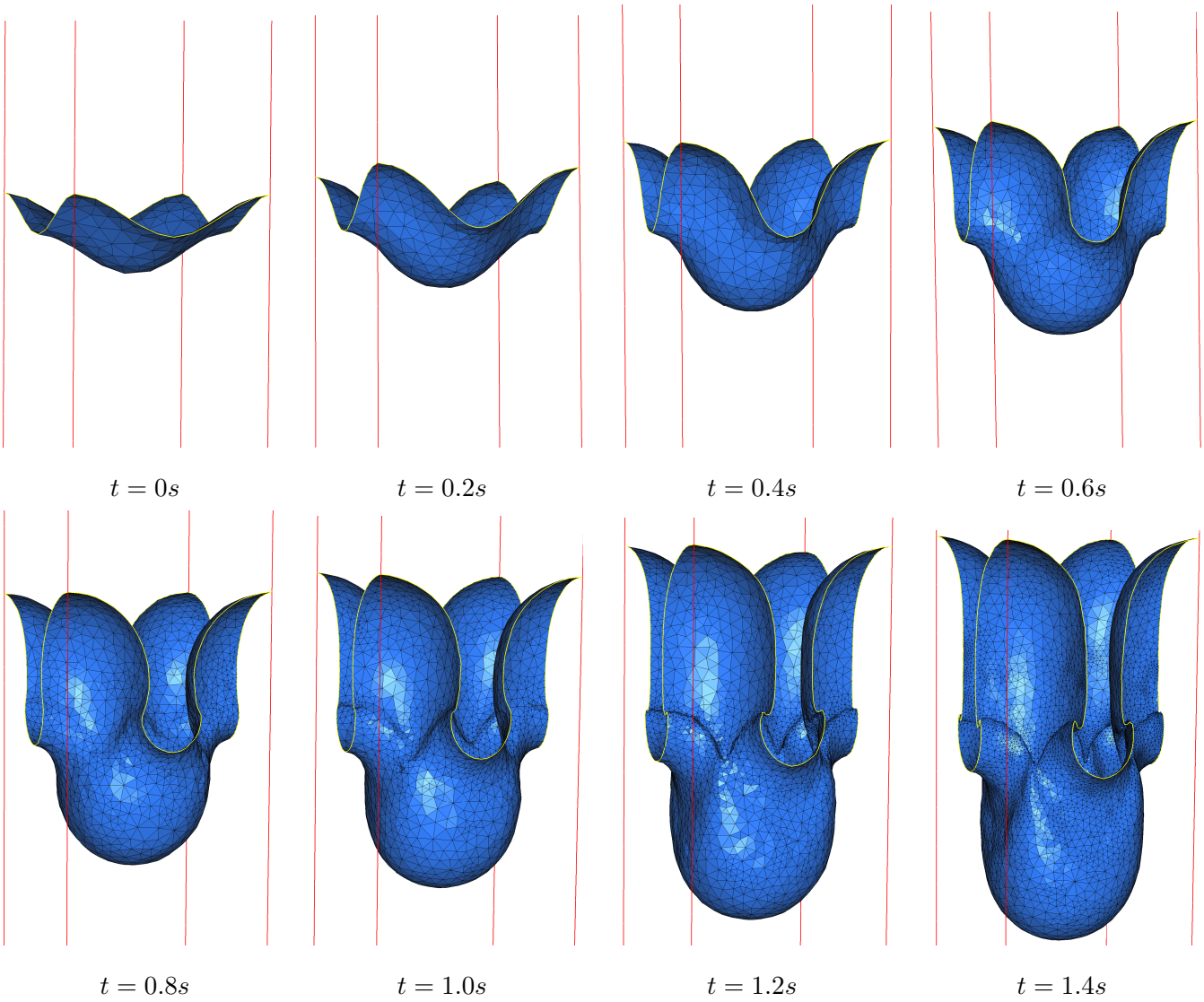


Figure 4.18: *Rayleigh-Taylor instability in 3D, evolution of the interface in time with At = 0.5.*

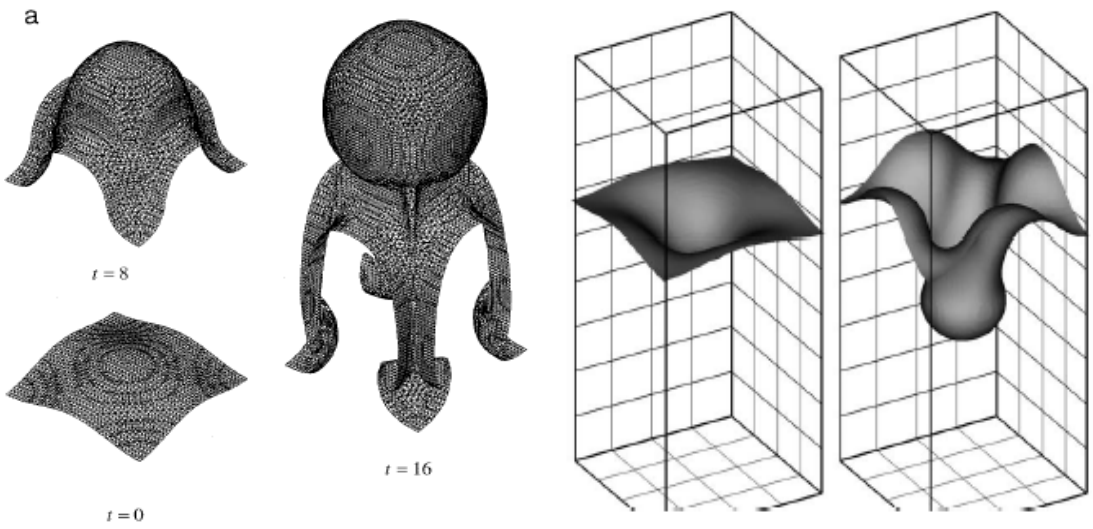


Figure 4.19: *Rayleigh-Taylor instability in 3D: results obtained in [LJG96] (left) and [LFX05] (right).*

mesh consists of 103.687 vertices (566.753 tetrahedral) with $h_{min} = 0.002$, $h_{max} = 0.1$. Due to mesh adaptation h_{min} is decreased at each time step in order to well capture the interface with the minimal computational time.

4.4 The coalescence of two rising bubbles

Finally, we consider the rising of two bubbles in a fluid under the gravity and the coalescence during the rising. The aim of this simulation is to show the capacity of managing topology changes of the present scheme.

4.4.1 The coalescence of two rising bubbles in 2D

The computational domain is $\Omega = [0, 1] \times [0, 4]$. The under circular bubble is initially centered at $(0.5, 1)$, with a radius of 0.25 and the upper one is initially centered at $(0.5, 1.75)$, with a radius of 0.15. The density and the viscosity of the fluid and two bubbles are respectively:

$$\nu_1 = 10^{-1} kg/(m.s), \quad \rho_1 = 100 kg/m^3$$

$$\nu_2 = 10^{-2} kg/(m.s), \quad \rho_2 = 1 kg/m^3$$

The surface tension coefficient is set as $\gamma = 6 \times 10^{-3}$. The gravity is $g = 9.81 m/s^2$. In this simulation, the problem is endowed with the following boundary conditions:

– homogeneous Dirichlet conditions, $\mathbf{u} = 0$, on the horizontal walls;

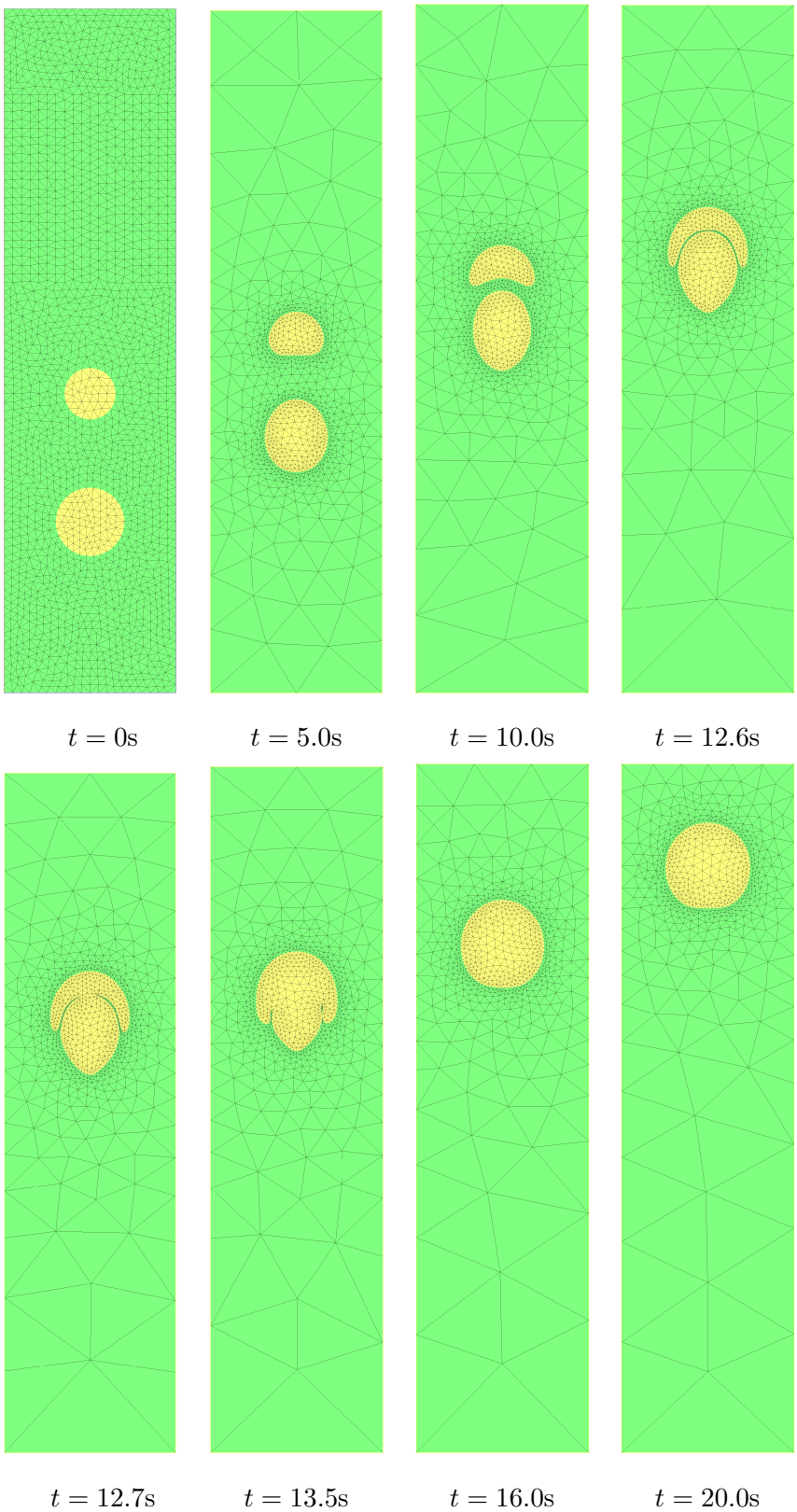


Figure 4.20: *Coalescence of two bubbles: evolution of the interface in time. The coalescence occurs at* $t = 12.7s$.

– free-slip condition on the vertical walls: $\tau.\sigma\mathbf{n} = 0$ and $\mathbf{u}.\mathbf{n}$=0;

The time step has been set to $\Delta t = 0.1$. It is observed that the rising and the coalescence process can be divided into three consecutive stages:

– The rising of two separate bubbles.

– The coalescence of two bubbles to the one.

– The rising of aggregate bubble to the top boundary.

Initially, the shape of two bubbles are slightly distorted. We see that the shape of two bubbles become rapidly distorted and the deformations become even more pronounced until the coalescence occurs (see figure 4.20). This simulation is implemented with 200 iterations of the algorithm in

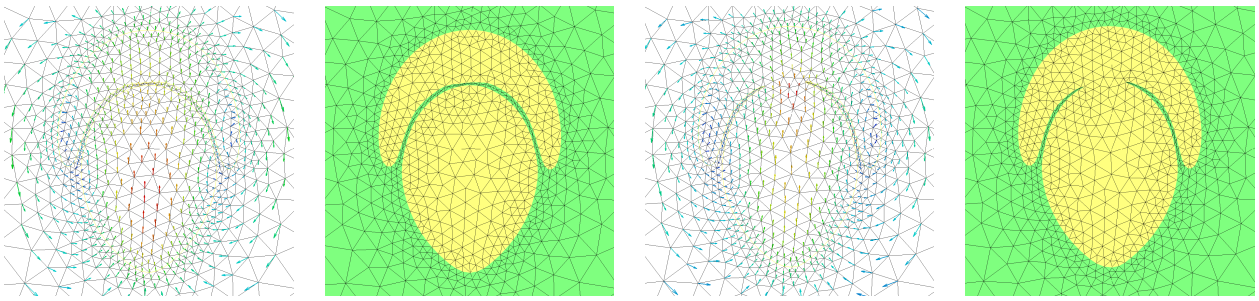


Figure 4.21: *Zoom of mesh and velocity of 2 bubbles at* $t = 12.6s$ *and* $t = 12.7s$.

section 2.5. The adapted mesh is about 1100 vertices at each time with the parameters: $h_{min} = 0.02, h_{max} = 1.0, h_{grad} = 1.5$. See figure 4.21 for the zoom of adapted mesh on the interfaces. The computational time takes about 6 minutes (on MacBook Air 2.13 GHz).

4.4.2 The coalescence of two rising bubbles in 3D

The computational domain is $\Omega = [0,1] \times [0,4]$. The under bubble is initially centered at $(0.5, 0.5, 1.0)$, with a radius of 0.25 and the upper one is initially centered at $(0.5, 0.5, 1.6)$, with a radius of 0.15. The physical parameters of the fluid (ν_1, ρ_1) and two bubbles (ν_2, ρ_2) are similar in the 2D case:

$$\nu_1 = 10^{-1} kg/(m.s), \quad \rho_1 = 100 kg/m^3$$

$$\nu_2 = 10^{-2} kg/(m.s), \quad \rho_2 = 1 kg/m^3$$

We impose also the homogeneous Dirichlet conditions: $\mathbf{u} = 0$ on the horizontal walls and the free-slip condition on the vertical walls: $\tau.\sigma\mathbf{n} = 0$ and $\mathbf{u}.\mathbf{n}$=0.

Figure 4.22 shows the evolution of the interfaces during the rising with the coalescence of two bubbles.

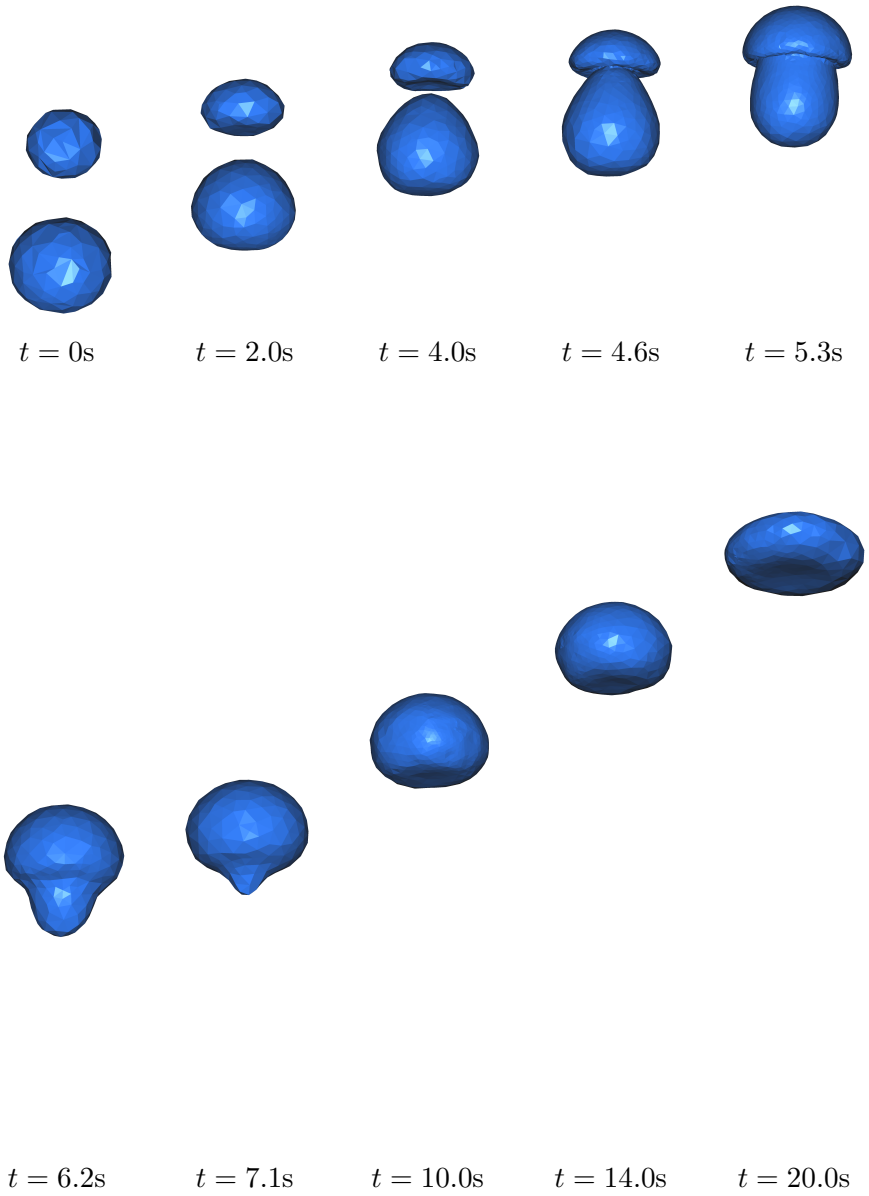


Figure 4.22: *Coalescence of two bubbles in 3D: evolution of the interfaces in time. The coalescence occurs at* $t = 4.6s$.

A comparison was performed between the present results and the experimental results obtained by Narayanan et al. [NGK74], numerical results obtained by Chen and Li [CL98], also those in [dSMN$^+$04]. The similarity of the shape development of the two bubbles can be clearly seen from figures 4.23, 4.24 and 4.25. Because of different parameters and time evolution, we are not interested in more detail comparison in this experiment.

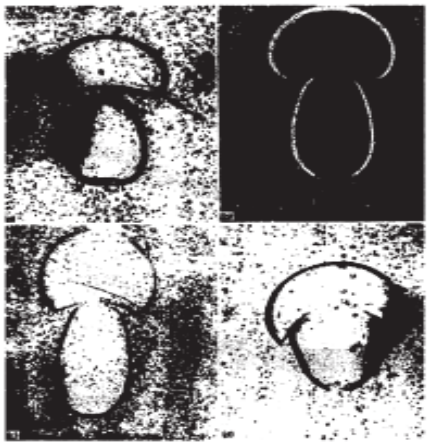


Figure 4.23: *Coalescence of two bubbles in 3D: results obtained in [NGK74].*

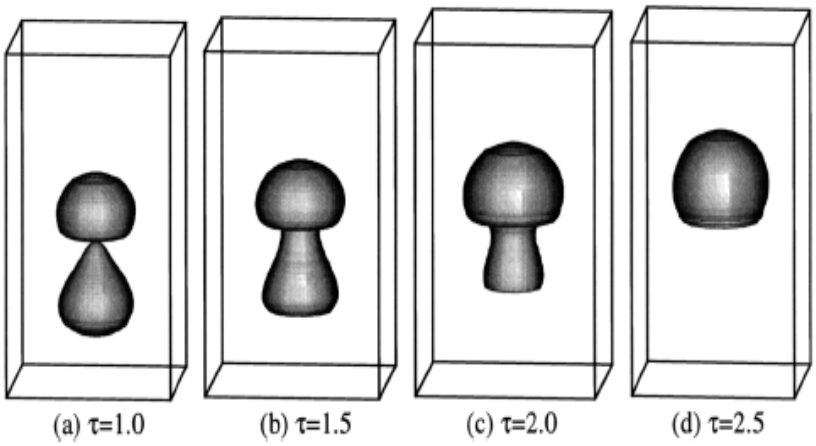


Figure 4.24: *Coalescence of two bubbles in 3D: results obtained in [CL98].*

To be more specific, the details of the coalescence of the two bubbles are shown in figure 4.26 for the velocity fields of the flows during the evolution. It can be seen that the liquid circulation around each bubble produces a jet that pushes the lower surface of both upper and under bubbles, and deformations of the bubbles occur (figure 4.26,(b)). Due to the effect of the velocity field around the upper bubble, the under bubble is stretched and becomes a spherical-oval shape (figure

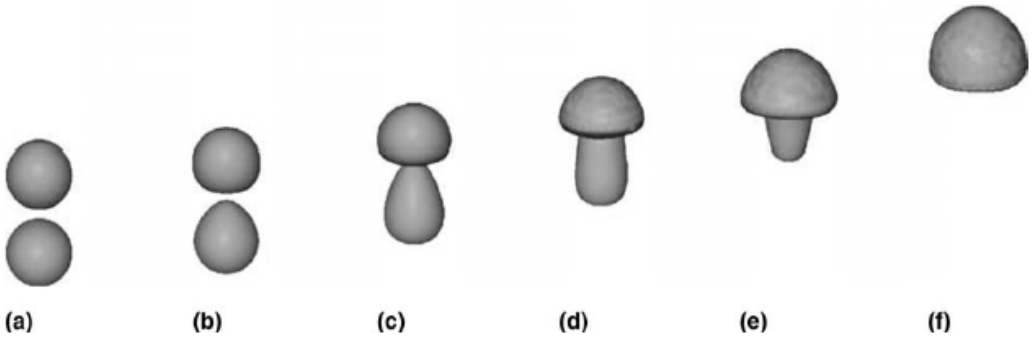


Figure 4.25: *Coalescence of two bubbles in 3D: results obtained in [dSMN$^+$04]: (a) 0.0 s; (b) 0.03 s; (c) 0.06 s; (d) 0.09 s; (e) 0.12 s; (f) 0.15 s.*

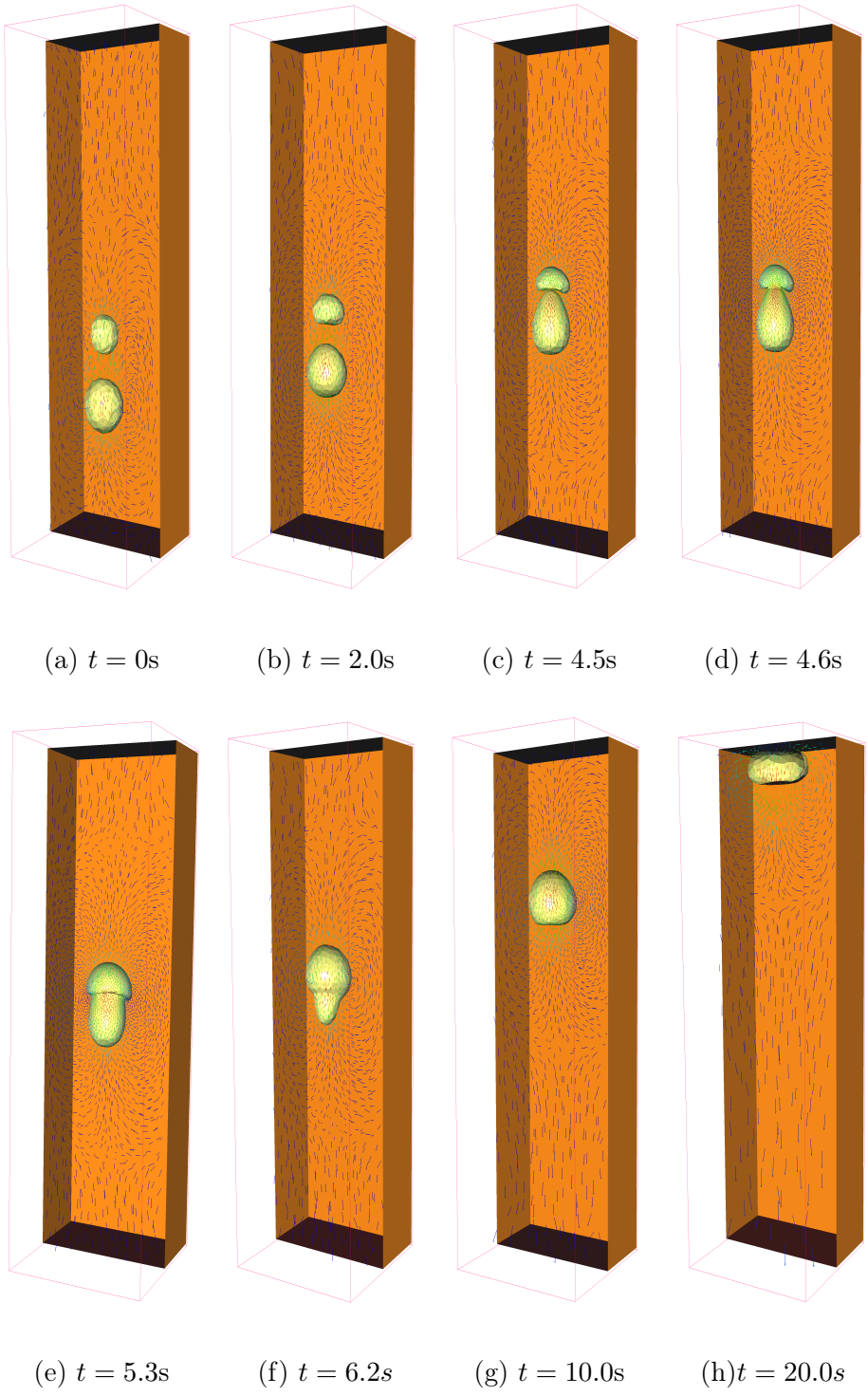
4.26,(c)). After the coalescence occurs (figure 4.26,(d)), the lower surface of the aggregate bubble is accelerated by the liquid jet and a spherical cap is obtained (figure 4.26, (e),(f),(g)). The aggregate bubble becomes dimpled ellipsoidal shape and more dimpled in time process (figure 4.26,(h)) as the deformation of single rising bubble in section 4.4.1. The contour velocity fields in figure 4.26 have shown an accurate representation with those in figure 9 of [CL98].

This simulation is implemented with 200 iterations. The adapted mesh is about 30000 vertices at each time with the parameters: $h_{min} = 0.005, h_{max} = 0.2, h_{grad} = 1.1, h_{hausd} = 0.005$. See figure 4.27 for more detail images of initial mesh and adapted one at the coalescence. The computational time takes about 2.5 minutes for each iteration.

4.5 Conclusion

The numerical results obtained in this chapter are in good accordance with those in many references which have been cited in each test case. It can be seen that the proposed scheme for bi-fluid flows, that is based on mesh adaptation, is capable of managing complex movements of the interface include topology change. Furthermore, the computational adapted triangulations have a modest number of nodes, while maintaining a relatively high accuracy near the interface, and the computational time is acceptable.

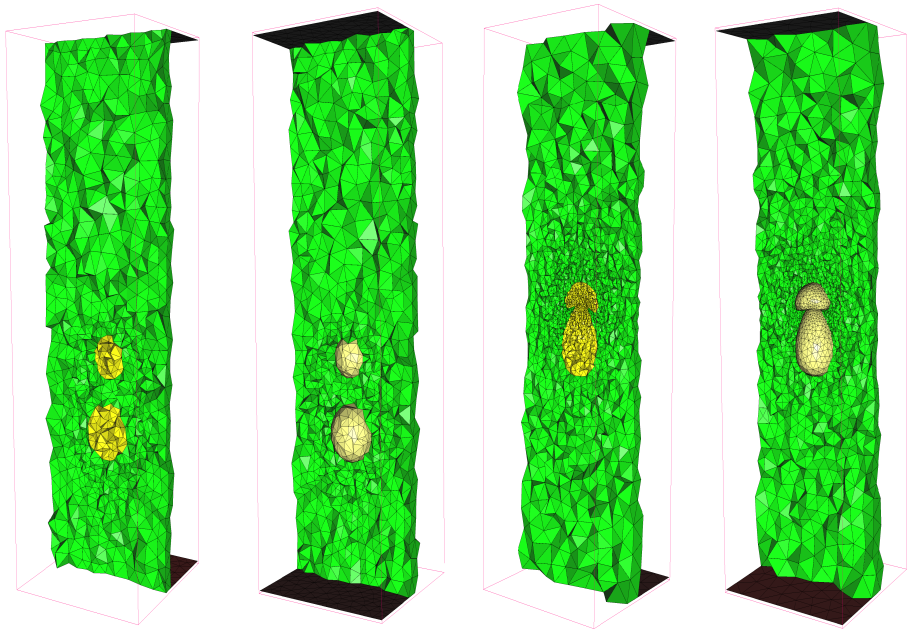
This chapter finishes also the first part of this thesis. We go to the second part where the considered problem is related to the optimization of computational domain, i.e the computational domain is variable during the time evolution.



(a) $t = 0$s (b) $t = 2.0$s (c) $t = 4.5$s (d) $t = 4.6$s

(e) $t = 5.3$s (f) $t = 6.2s$ (g) $t = 10.0$s (h)$t = 20.0s$

Figure 4.26: *Coalescence of two bubbles in 3D: Velocity fields, cutting in the plan* $y = 0.5$.



(a) (b) (c) (d)

Figure 4.27: *Coalescence of two bubbles in 3D: Initial mesh with 11113 vertices (two first ones) and adapted mesh with 30101 vertices at the coalescence (two last ones).*

Part II

Shape optimization in fluid mechanics

In the first part of this thesis, we have mentioned the following problems:

- The numerical resolution of Navier-Stokes equations for two fluids flow with the interface.
- The numerical resolution of the evolution of the interface (free surface) by a formulation of the level set method.
- Mesh adaptation techniques, especially for controlling the geometrical discrepancy between the free surface and its approximation.

All these problems and their numerical counterparts allow us to resolve any stationary or non stationary problems for two phase flows in a fixed domain. However, in many aspects of the application another challenge is related to the optimization of the geometry of the computational domain with respect to a given functional. This part of computational mechanics has soared over the last decades and is devoted to finding the optimal *size, shape* or *topology* of a mechanical part, with respect to a given mechanical criterion - such as, for instance, the work of external loads, or the internal stress. These problems strongly depend on the physics at play (elasticity, thermoelasticity, fluid mechanics ...), and on the constraints that should be fulfilled by the shapes.

In the context of *structural optimization*, size, shape and topology optimization problems address

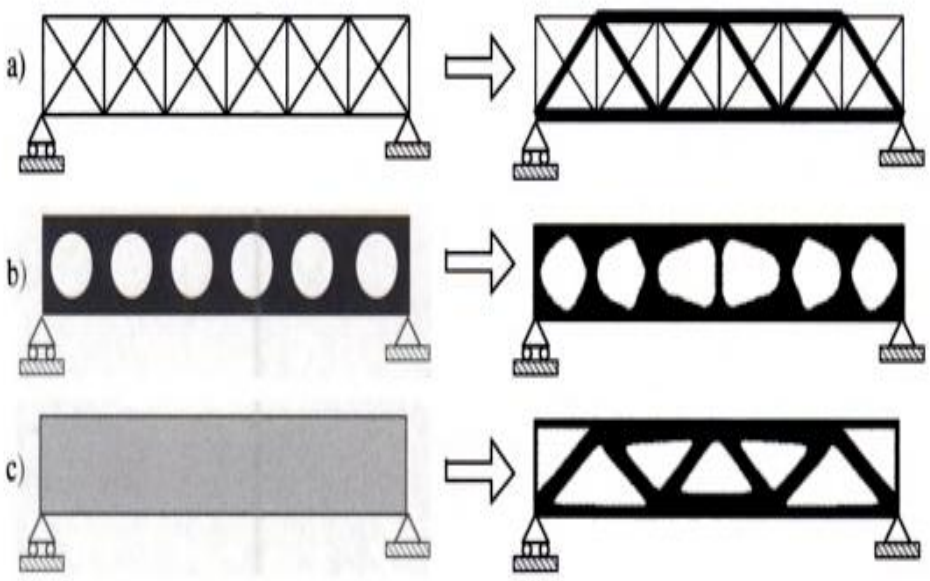


Figure 4.28: *Three categories of structural optimization. (a) Size optimization of a truss structure, (b) shape optimization and (c) topology optimization. The initial problems are shown at the left hand side and the optimal solutions are shown at the right. (from [BS03])*

different aspects of the structural design problem. In a typical *size* problem the goal may be to find

the optimal thickness distribution of a linearly elastic plate or the optimal member areas in a truss structure. On the other hand, in a *shape* optimization problem the goal is to search the optimum shape of this domain, that is, the shape is now the design variable. Unlike the case of classical shape optimization where only the boundary of the domain is optimized, in *topology* optimization the structure of the domain (the number of holes and the connectivity of the domain for instance) may change during the optimization process. See figure 4.28 for the illustration of three categories of structural optimization.

Altogether different problems coming from physics, mechanics, biology, etc... can be cast into the different frameworks of structural optimization. To name a few:

- The broad approaches on the design optimization of heat conduction problems can be found in [SK96, Mer98, YPG01, HMY03]. Shape optimization methods for thermal structure have also been utilized for multiobjective function in [PCL03, NPR06], for practical uses. See also in [LSQX99] an extended algorithm for evolutionary structural optimization (ESO) to shape and topology design problems subjected to steady heat conduction.
- Many structure topology optimization approaches have been introduced to the field of elastic structure design in recent years, see [SW00, WWG03] for examples. The requiring of structural optimization for a model of linear or nonlinear elasticity with different objective functions have been investigated in [AJT02, AJT04].
- Last but not least, shape optimization in fluid mechanics has received a large amount of attention from both engineers and mathematicians since the pioneering works of Pironneau [Pir73, Pir74]. Many applications of shape optimization concepts in this domain, to name a few: the optimal shape design (OSD) of airplane wing which induces minimal drag, i.e. minimal reaction from the surrounding fluid in aeronautic industry, see [MP04, POTTP06]; the optimal control approaches to shape optimization of aorto-coronaric bypass anastomoses are applied in [QR03, AQR06a, AQR06b]; the problem of optimal swimming of microorganisms at low Reynolds number is considered in [ADH11], more specifically, the authors are in search for the shape of such microorganism that allows to reach a prescribed displacement while undergoing minimal stress from the surrounding fluid, the shape of the microorganisms is parametrized by means of a few physical parameters, and the sensitivity of the stress exerted by the fluid on the microorganisms with respect to perturbations on these parameters is computed; an interesting application of the topology optimization method for fluid flow can be found in [BP03] for minimum power dissipation Stokes flow problems in two dimensions. Since then, alternative

parameterizations have been suggested for the Stokes flow problem [GP06] and the Navier Stokes flow problem [Evg06]. Topology optimization for Stokes flow has been extended to large-scale problems [APGHS08] and has been applied to fluid flows with regions of Darcy and Stokes flow [WKB07] and with low-to-moderate Reynolds numbers [OOB06, DMZ08a, ZL08]. For more works and many other applications of shape or topology optimization concepts in fluid mechanics, we refer to [BS03, Gun03, MP04, MP10] and references therein.

In this part our aim is to construct a numerical scheme for the problem of shape optimization in fluid mechanics: Let $\Omega \in \mathbb{R}^d (d = 2 \text{ or } 3)$ be a bounded open set occupied by the fluid governed by Stokes equations:

$$\begin{cases} -\nu\Delta u + \nabla p &= f \text{ in } \Omega \\ \text{div} u &= 0 \text{ in } \Omega \end{cases} \tag{1}$$

with some adequate boundary conditions. The problem we consider is to minimize an objective functional denoted by $J(\Omega)$ which depends on the domain Ω via the solution u_Ω of (1) where the variable shape Ω belongs to a set of admissible shapes $\mathcal{U}_{ad}$. Formally speaking, our model of shape optimization is:

$$\inf_{\Omega \in \mathcal{U}_{ad}} J(\Omega)$$

This problem has also been considered in [BP03, DMZ08b, DMZ08a]. Before talking about our approach, let us mention some techniques of shape optimization that have been introduced in computational mechanics since the 1960s. Regarding numerous methods to handle the structural optimization problems in the literatures, one can see that these methods are mainly distinguished by the way they represent the shapes and the way they compute the sensitivity of the objective criterion with respect to the design.

Describing shapes is a complicated problem since it needs to satisfy two requirements. Firstly, shape optimization techniques must be able to perform mechanical computations on the considered shapes, e.g. by means of finite differences, finite element or finite volume methods, and not all kind of representations lend themselves to such computations. Secondly, the adopted representation must be versatile enough to allow for a robust account of shapes' deformations. Some typical approaches for descriptions of shapes can be named as:

- *explicit methods*: in this case, shapes are explicitly accounted by a set of degrees of freedom, see for instance [BF84] around these issues, or the review article [HG86], and monograph [Pir84].
- *homogenization methods* and their variants (power-law method or solid isotropic microstructure

with penalization (SIMP) method), may be considered quite classical in view of the number of publications see [BK88, Ben95, BS03, ABFJ97, All02] for examples. These methods belong to the class of *density methods*. They have made a notable change in perspectives in structural optimization by using a density function $\theta : D \to [0, 1]$ defined over a computational domain D, where θ is close to 0 if there is almost only void (or a very "soft" material, mimicking void), and where θ is close to 1 if there is almost only the shape. The problem of finding the "best" shape is transformed into that of the optimal distribution of a mixture of material and void in a large computational domain. Therefore, the shape optimization problem ends rephrased as a parametric optimization problem.

– *implicit methods*: have been used broadly for interface-tracking or interface-capturing in various fields and this method became very popular in shape-topology optimization, the most famous of them being the *level set method*. For examples, Wang et al. developed a topology optimization method for linearly elastic structures using an implicit moving boundary [WWG03]. They represented the structural boundary using the level set model embedded in a higher-dimensional scalar function. Allaire et al. proposed a structural optimization method based on a combination of shape derivative and the level set method for front propagation in [AJT04]. In their formulation, the compliance and target displacement objectives are considered. Optimization problems for heat conduction have been solved by level set-based optimization and extended to nonlinear problems by Ha and Cho [HC05, HC08]. Kim et al. developed a level set-based topology optimization method for heat conduction problems utilizing topological derivatives as a nucleation criterion [KHC09]. Recently Duan et al. have applied variational level set methods to shape and topology optimization of fluid flows in [DMZ08b, DMZ08a]. They demonstrated their approach with two-dimensional examples and proposed new evolution equations for the level set function in order to achieve a smooth evolution without reinitialization. Zhou and Li [ZL08] have also used the level set method for the topology optimization of steady-state Navier Stokes flows in both two and three dimensions. They stated that the computational costs incurred by remeshing the fluid domain, extending the velocity and reinitializing the level set function were limitations of their model.

It is obvious that the performing of perturbations of the considered shapes is closely related to the problem of shape description. Our approach for shape description belongs to the last class: we apply a conventional level set approach [WWG03, AJT04] where a precise description of the boundary is kept under the form of an auxiliary function (level set function). In our approach the

deformation of shape Ω is reformulated in terms of an associated level set function $\phi(.,t)$ as the following *level set advection* equation over a larger domain D :

$$\frac{\partial \phi}{\partial t} + V.\nabla\phi = 0$$

where the advection vector V is the steepest descent direction for optimization corresponding to the shape derivative of the given objective function. We also compute a shape derivative by using an adjoint problem, however unlike the approach in [AJT04] using the so-called "ersatz material" which amounts to fill the holes by a weak phase, we only need impose the Stokes and adjoint problems on a sub-mesh T of the shape domain. This can be achieved thanks to the explicit discretization of generated shape during the optimization process.

It is worth noting that although this method is not specifically designed for topology optimization, it can easily handle topology changes. For this reason, if the optimum design of the structure includes information on the topology and shape of the structure, the level set models allow to deal with these two problems simultaneously. We shall describe the details of our proposed approach and its numerical implementation in the chapter 5 hereafter.

Chapter 5

Shape optimization in fluid mechanics

Contents

As mentioned in the introduction, our numerical scheme is based on a combination of classical shape derivative, level set method and mesh adaptation. The motivation of this approach related to the following components:

– In the context of level set method, the variational shape Ω is classically represented as the negative subdomain of a level set function ϕ of "larger" computational domain D. In the first part of this thesis, this method has been approached for the evolution of the interface Γ where

Γ is represented as the zero-level set of ϕ. Hence, if we consider the boundary $\partial\Omega$ of variable shape Ω as the "interface" defined by the zero-level set: $\partial\Omega = \{x \in D : \phi(x) = 0\}$, then the problem of shape deformation will be considered as the problem of domain evolution which involves only to the advection of $\partial\Omega$. In this way, we can apply the numerical resolution of advection equation (is detailed in Chapter 2) for the evolution of boundary of Ω, and hence for the deformation of Ω.

- Consequently, during the shape deformation the mesh adaptation process is constructed by the same approach (Chapter 3) to guarantee the accuracy of geometrical approximation of the continuous shape Ω as well as the quality of the adapted mesh for the resolution of mechanical problems.
- Level set method has been used broadly in structural optimization. In [Dap13], the combination of level set methods on unstructured meshes and mesh evolution has led to successful results in shape optimization of elastic structures.

The main difficulty is related to the problem of introducing perturbations of the considered shapes, and thus on how to compute the sensitivity of the objective function J with respect to the shape. We overcome this problem in section 5.1, and show that the computation of the sensitivity of objective function is related to the Hadamard's boundary variation method and the shape derivatives is computed by Céa's formal method. They are the key ingredients to construct a general scheme for shape optimization in fluid mechanics (section 5.2). Section 5.3 will detail our numerical scheme using level set method and mesh adaptation. Last, in section 5.4 a numerical example with the objective function of energy dissipation is presented to assess the efficiency and the reliability of the proposed scheme.

5.1 Shape sensitivity analysis using Hadamard's boundary variation method

In this section we focus on one particular method for describing variations of a shape, namely Hadamard's boundary variation method, as well as the notions of differentiation with respect to the domain. At first, we introduce the basic ideas of Hadamard's method and set some notations, that we shall use in the rest of this part. Then, we present the derived notions of differentiation with respect to a domain: the notion of shape derivative of a scalar function of the domain, or of a function which is itself defined on the domain is introduced. Some classical results is recalled to

prepare for the computation of shape derivative. Finally, we apply the stress on a specific context: the optimization of mechanics fluid in context of Stokes system.

5.1.1 Hadamard's boundary variation method

The central idea of Hadamard's boundary variation is proposed in its seminal paper [Had07] (see also [SZ92]). It was then exploited in depth in [MS76]. Let θ be a displacement field of "small" amplitude, we consider variations of a given reference shape Ω of the form $(I+\theta)(\Omega)$ (see figure 5.1). Thus, all the considered transformations $(I+\theta)(\Omega)$ are homeomorphisms "close" to the identity; in particular, all the variations of Ω achieved in this way share the same topology.

Note that in the framework of Hadamard's method, the variations Ω_θ of Ω only depend on the

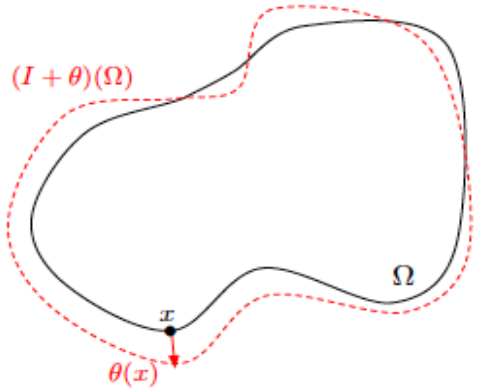


Figure 5.1: *Variation $I+\theta$ of a reference shape Ω.*

values taken by θ on $\partial\Omega$, and we have following consequence of Picard's fixed point theorem (see lemma 6.13 in [All06] for a proof):

Lemma 1. *For every deformation field $\theta \in W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)$ such that $\|\theta\|_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)} < 1$, the application $(I+\theta) : \mathbb{R}^d \to \mathbb{R}^d$ is a Lipschitz homeomorphism with Lispchitz inverse.*

To the end of this part $\Omega \subset \mathbb{R}^d$ stands for a fixed domain, which enjoys at least Lispchitz regularity. For any $\theta \in W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d), \|\theta\|_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)} < 1$, we denote as $\Omega_\theta := (I+\theta)(\Omega)$ the deformed shape with respect to θ. Note that $W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d) \subset L^\infty(\mathbb{R}^d)^d$ is the Banach space of bounded functions $\theta : \mathbb{R}^d \to \mathbb{R}^d$, endowed with the natural norm:

$$\forall \theta \in W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d), \|\theta\|_{W^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d)} := \|\theta\|_{L^\infty(\mathbb{R}^d)^d} + \|\nabla\theta\|_{L^\infty(\mathbb{R}^d)^{d\times d}}$$

Hence, variations of a given shape Ω end up parametrized by means of an open subset of a Banach space. As we shall see in the next section, this allows among other things to introduce a notion a

differentiability with respect to the shape by rewriting any operation performed on shapes close to Ω in terms of the underlying deformation field θ.

5.1.2 Shape differentiability

Following the approach of Murat and Simon [MS76, Sim80], starting from a smooth reference open set Ω, we consider domains of the type $\Omega_\theta = (I+\theta)(\Omega)$ with $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ as been defined in the previous subsection. Let us present some classical definition and results about the differentiation with respect to the domain of functionals of type: $\Omega \mapsto J(\Omega) \in \mathbb{R}$.

Definition 4. *Let $J(\Omega)$ a functional of the domain. J is shape differentiable at Ω if the underlying application:*

$$\begin{aligned} W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) &\longrightarrow \mathbb{R} \\ \theta &\longmapsto J(\Omega_\theta) \end{aligned}$$

is Fréchet differentiable at $\theta = 0$. The associated Fréchet differential; denoted as $J'(\Omega)$ is called the shape derivative of J at Ω. We have following expansion in the vicinity of $0 \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$:

$$J(\Omega_\theta) = J(\Omega) + J'(\Omega)(\theta) + o(\theta), \quad \text{with } \lim_{\theta \to 0} \frac{|o(\theta)|}{\|\theta\|} = 0 \tag{5.1}$$

Remark 6.

1. $J'(\Omega)$ is a continuous linear form on $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$.

2. In the case of a functional which also depends on other variables than Ω, the partial Fréchet differential with respect to the domain is denoted as $\frac{\partial}{\partial \Omega}$.

We recall a classical result which states that the directional derivative $J'(\Omega)(\theta)$ depends only the normal trace $\theta.n$ on the boundary $\partial\Omega$, see [AJT04].

Lemma 2. *Let Ω be a smooth bounded open set and $J(\Omega)$ a differentiable function at Ω, if $\theta_1, \theta_2 \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ are such that: $\theta_2 - \theta_1 \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$ and $\theta_1.n = \theta_2.n$, then we have:*

$$J'(\Omega)(\theta_1) = J'(\Omega)(\theta_2)$$

We also recall here two results on shape derivative that will be useful in the computation of shape derivatives in the section 5.1.3 (see [HP05] or [AJT04] for the proofs).

Lemma 3. *Let Ω be a bounded Lispchitz domain and let $f(x) \in W^{1,1}(\mathbb{R}^d)$. Define: $J(\Omega) = \int_\Omega f(x)dx$ then $J(\Omega)$ a differentiable function at Ω, and we have:*

$$J'(\Omega)(\theta) = \int_\Omega div(\theta f)dx = \int_{\partial\Omega} f\theta.nds$$

for any $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$

Lemma 4. *Let Ω be a bounded Lispchitz domain in class $\mathcal{C}^2$ and $g(x) \in W^{2,1}(\mathbb{R}^d)$. Define: $J(\Omega) = \int_{\partial\Omega} g(x)ds$ then $J(\Omega)$ a differentiable function at Ω, and:*

$$J'(\Omega)(\theta) = \int_{\partial\Omega} (\frac{\partial g}{\partial n} + \kappa g)\theta.nds$$

for any $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$, where κ is the mean curvature of $\partial\Omega$ defined by $\kappa = divn$. Furthermore, this result still holds true if one replaces $\partial\Omega$ by Γ, a smooth open subset of $\partial\Omega$, and assumes that $h = 0$ on the surface boundary of Γ.

In particular, Lemma 3 is useful in order to compute the shape derivative of a volume constraint $V(\Omega) = C$. Indeed, we have:

$$V(\Omega) = \int_\Omega dx \quad \text{and} \quad V'(\Omega)(\theta) = \int_{\partial\Omega} \theta(x).n(x)ds. \tag{5.2}$$

Similarly, Lemma 4 is useful in order to compute the shape derivative of a perimeter constraint $P(\Omega) = C$ since we have:

$$P(\Omega) = \int_{\partial_\Omega} ds \quad \text{and} \quad P'(\Omega)(\theta) = \int_{\partial\Omega} \kappa\theta(x).n(x)ds.$$

5.1.3 Lagrange's approach for the computation of shape derivatives in the context of Stokes system

In this section, we present the computation of the shape derivative for objective functionals $J(\Omega)$ which depend on the domain Ω via the solutions u_Ω to the Stokes system:

$$\begin{cases} -\Delta u + \nabla p &= f \text{ in } \Omega \\ \text{div} u &= 0 \text{ in } \Omega \end{cases} \tag{5.3}$$

endowed with some adequate boundary conditions.

The rigorous presentation of this topic is not an easy task. In practice, we rely on a formal method to obtain the expressions of these shape derivatives, namely Céa's fast derivation method, introduced in [Céa86]. It is formal in the sense that all the data at hand: objective functions, domains, deformations are assumed smooth enough and the state solution u_Ω, p_Ω is differentiable with respect to the shape. Consider the following cases of boundary conditions:

1. *Newmann boundary condition*

We consider the objective function of type:

$$J(\Omega) = \int_\Omega j(u_\Omega)$$

and the state equations (5.3) are endowed by Newmann condition as follows:

$$\begin{cases} -\Delta u + \nabla p &= f \text{ in } \Omega \\ \text{div} u &= 0 \text{ in } \Omega \\ \sigma(u,p)n &= \varphi_N \text{ on } \partial\Omega \end{cases} \tag{5.4}$$

In this case, we introduce the Lagrange functional defined for (u, p, v, q) as follows:

$$\mathcal{L}(\Omega, u, p, v, q) = \int_\Omega j(u)dx + \int_\Omega (-\Delta u + \nabla p - f).v dx - \int_\Omega q\text{div}u$$

Taking $v \in (H^1(\mathbb{R}^d)^d)$ and $q \in L^2(\mathbb{R}^d)$ and using Green formula:

$$\mathcal{L}(\Omega, u, p, v, q) = \int_\Omega j(u)dx + \int_\Omega (\nabla u.\nabla v - p\text{div}v - f.v - q\text{div}u)dx - \int_{\partial\Omega} \varphi_N.v$$

We have the partial differential of $\mathcal{L}$ with respect to the each variable as follows:

$$\langle \frac{\partial \mathcal{L}}{\partial v}(\Omega, u, p, v, q), \bar{u} \rangle = \int_\Omega (\nabla u.\nabla \bar{u} - p\text{div}\bar{u} - f.\bar{u})dx - \int_{\partial\Omega} \varphi_N.\bar{u} \tag{5.5}$$

$$\langle \frac{\partial \mathcal{L}}{\partial q}(\Omega, u, p, v, q), \bar{p} \rangle = \int_\Omega -\bar{p}\text{div}u \tag{5.6}$$

$$\langle \frac{\partial \mathcal{L}}{\partial u}(\Omega, u, p, v, q), \bar{v} \rangle = \int_\Omega j'(u).\bar{v}dx + \int_\Omega (\nabla \bar{v}.\nabla v - q\text{div}\bar{v})dx \tag{5.7}$$

$$\langle \frac{\partial \mathcal{L}}{\partial p}(\Omega, u, p, v, q), \bar{q} \rangle = \int_\Omega -\bar{q}\text{div}v \tag{5.8}$$

We can see that (5.5), (5.6) lead the variational formulation of the state system (5.4). It has been known that in the weak form this problem has unique solution (u_Ω, p_Ω), and we can easily verify that:

$$J(\Omega) = \mathcal{L}(\Omega, u_\Omega, p_\Omega, v, q), \quad \forall\, (v, q) \in (H^1(\mathbb{R}^d)^d, L^2(\mathbb{R}^d))$$

Taking the partial differential with respect to domain Ω we have:

$$\begin{aligned} J'(\Omega)(\theta) = &\frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_\Omega, p_\Omega, v, q)(\theta) \\ &+ \langle \frac{\partial \mathcal{L}}{\partial u}(\Omega, u_\Omega, p_\Omega, v, q), u'(\Omega)(\theta) \rangle + \langle \frac{\partial \mathcal{L}}{\partial p}(\Omega, u_\Omega, p_\Omega, v, q), p'(\Omega)(\theta) \rangle \end{aligned} \tag{5.9}$$

Hence, suppose that (v_Ω, q_Ω) is solution of the system:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial u}(\Omega, u_\Omega, p_\Omega, v, q) = 0 \\ \frac{\partial \mathcal{L}}{\partial p}(\Omega, u_\Omega, p_\Omega, v, q) = 0 \end{cases} \tag{5.10}$$

then $J'(\Omega)(\theta)$ can be written with the reduced formula:

$$J'(\Omega)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_\Omega, p_\Omega, v_\Omega, q_\Omega)(\theta) \tag{5.11}$$

The transformations (5.7), (5.8) lead us to formulate the system (5.10), also called the adjoint equations as:

$$\begin{cases} -\Delta v + \nabla q &= -j'(u_\Omega) \text{ in } \Omega \\ \text{div} v &= 0 \text{ in } \Omega \\ \sigma(v,q)n &= 0 \text{ on } \partial\Omega \end{cases} \tag{5.12}$$

Applying Lemma 3, Lemma 4 and taking into account the expression (5.11), the shape derivative of objective function can be computed as follows:

$$J'(\Omega)(\theta) = \int_{\partial\Omega} \theta.n \left(j(u) + \nabla u.\nabla v - p\text{div}v - f.v - q\text{div}u - \frac{\partial(\varphi_N.v)}{\partial n} - \kappa\varphi_N.v \right) ds \tag{5.13}$$

2. *Dirichlet boundary condition*

Let us now consider the case of Dirichlet condition, i.e the state equations are:

$$\begin{cases} -\Delta u + \nabla p &= f \text{ in } \Omega \\ \text{div} u &= 0 \text{ in } \Omega \\ u &= u_D \text{ on } \partial\Omega \end{cases} \tag{5.14}$$

We construct the Lagrange function as:

$$\mathcal{L}(\Omega, u, p, v, q, \lambda) = \int_\Omega j(u)dx + \int_\Omega (-\Delta u + \nabla p - f).v dx - \int_\Omega q\text{div}u dx + \int_{\partial\Omega} \lambda.(u - u_D) ds \tag{5.15}$$

The partial differential of $\mathcal{L}$ respect to the each variable:

$$\langle \frac{\partial \mathcal{L}}{\partial v}(\Omega, u, p, v, q, \lambda), \bar{u} \rangle = \int_\Omega (-\Delta u + \nabla p - f).\bar{u} dx \tag{5.16}$$

$$\langle \frac{\partial \mathcal{L}}{\partial q}(\Omega, u, p, v, q, \lambda), \bar{p} \rangle = \int_\Omega -\bar{p}\mathrm{div} u \tag{5.17}$$

$$\langle \frac{\partial \mathcal{L}}{\partial \lambda}(\Omega, u, p, v, q, \lambda), \bar{\lambda} \rangle = \int_{\partial\Omega} \bar{\lambda}.(u - u_D) \tag{5.18}$$

$$\begin{aligned}
\langle \frac{\partial \mathcal{L}}{\partial u}(\Omega, u, p, v, q, \lambda), \bar{v} \rangle &= \int_\Omega j'(u).\bar{v} dx + \int_\Omega -\Delta \bar{v}.v dx - \int_\Omega q\mathrm{div}\bar{v} dx + \int_{\partial\Omega} \lambda.\bar{v} ds \\
&= \int_\Omega j'(u).\bar{v} dx + \int_\Omega -\Delta v.\bar{v} dx + \int_\Omega \nabla q.\bar{v} dx \\
&+ \int_{\partial\Omega} (\partial_n v - qn + \lambda).\bar{v} ds - \int_{\partial\Omega} v.\partial_n \bar{v} ds \\
&= \int_\Omega (j'(u) - \Delta v + \nabla q).\bar{v} dx + \int_{\partial\Omega} (\sigma(v, q)n + \lambda).\bar{v} ds - \int_{\partial\Omega} v.\partial_n \bar{v} ds
\end{aligned} \tag{5.19}$$

$$\langle \frac{\partial \mathcal{L}}{\partial p}(\Omega, u, p, v, q, \lambda), \bar{q} \rangle = \int_\Omega \nabla \bar{q}.v = -\int_\Omega \bar{q}\mathrm{div} v + \int_{\partial\Omega} \bar{q} n.v \tag{5.20}$$

Introducing the condition $\sigma(v, q)n + \lambda = 0 \,\mathrm{on}\, \partial\Omega$, the transformation (5.19) and(5.20) lead us to the adjoint system:

$$\begin{cases} -\Delta v + \nabla q &= -j'(u) \,\mathrm{in}\, \Omega \\ \mathrm{div} v &= 0 \,\mathrm{in}\, \Omega \\ v &= 0 \,\mathrm{on}\, \partial\Omega \end{cases} \tag{5.21}$$

Hence if we consider the Lagrange function at the equilibrium points of the state and the adjoint problems with the condition $\sigma n + \lambda = 0 \,\mathrm{on}\, \partial\Omega$, we obtain the formula:

$$J'(\Omega)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_\Omega, p_\Omega, v_\Omega, q_\Omega, \lambda)(\theta) \tag{5.22}$$

Applying Lemma 3 and Lemma 4 to this expression leads us to write the shape derivative of the objective function as:

$$J'(\Omega)(\theta) = \int_{\partial\Omega} \theta.n \left(j(u) + (-\nabla u + \Delta p - f).v - q\mathrm{div} u + \frac{\partial \lambda (u - u_D)}{\partial n} + \kappa \lambda (u - u_D) \right) ds \tag{5.23}$$

3. *Mixed boundary conditions*

Let us come to the more general case in which the state system is imposed both Dirichlet and

Newmann conditions ($\partial\Omega = \Gamma_D \cup \Gamma_N$). We consider here the state system:

$$\begin{cases} -\Delta u + \nabla p &= f \,\text{in}\, \Omega \\ \text{div} u &= 0 \,\text{in}\, \Omega \\ \sigma(u,p)n &= \varphi_N \,\text{on}\, \Gamma_N \\ u &= 0 \,\text{on}\, \Gamma_D \end{cases} \tag{5.24}$$

and the objective function we consider is the following:

$$J(\Omega) = \int_\Omega j(u_\Omega(x))dx + \int_{\Gamma_N} l(u_\Omega(x))ds$$

The Lagrange function can be constructed as:

$$\begin{aligned} \mathcal{L}(\Omega,u,p,v,q) = &\int_\Omega j(u)dx + \int_{\Gamma_N} l(u)ds + \int_\Omega (\nabla u.\nabla v - p.\text{div}v - q.\text{div}u - f.v)dx \\ &- \int_{\Gamma_D} (\sigma(v,q)n.u + \sigma(u,p)n.v) - \int_{\Gamma_N} \varphi_N.v ds \end{aligned}$$

The partial differential of $\mathcal{L}$ with respect to the each variable read:

$$\begin{aligned} \langle \frac{\partial \mathcal{L}}{\partial v}(\Omega,u,p,v,q), \bar{u}\rangle &= \int_\Omega (\nabla u.\nabla \bar{u} - p.\text{div}\bar{u} - f.\bar{u})dx - \int_{\Gamma_D} (\partial_n \bar{u}.u + \sigma(u,p)n.\bar{u}) - \int_{\Gamma_N} \varphi_N.\bar{u} ds \\ &= \int_\Omega (-\Delta u + \nabla p - f).\bar{u} dx + \int_{\partial\Omega} \sigma(u,p)n.\bar{u} \\ &- \int_{\Gamma_D} \sigma(u,p)n.\bar{u} - \int_{\Gamma_N} \varphi_N.\bar{u} ds - \int_{\Gamma_D} \partial_n \bar{u}.u \\ &= \int_\Omega (-\Delta u + \nabla p - f).\bar{u} dx + \int_{\Gamma_N} (\sigma(u,p)n - \varphi_N).\bar{u} ds - \int_{\Gamma_D} \partial_n \bar{u}.u \end{aligned} \tag{5.25}$$

$$\langle \frac{\partial \mathcal{L}}{\partial q}(\Omega,u,p,v,q), \bar{p}\rangle = \int_\Omega -\bar{p}\text{div}u + \int_{\Gamma_D} \bar{p}n.u \tag{5.26}$$

$$\begin{aligned} \langle \frac{\partial \mathcal{L}}{\partial u}(\Omega,u,p,v,q), \bar{v}\rangle &= \int_\Omega j'(u)\bar{v}dx + \int_{\Gamma_N} l'(u).\bar{v}ds + \int_\Omega (\nabla \bar{v}.\nabla v - q\text{div}\bar{v})dx - \int_{\Gamma_D} (\sigma(v,q)n.\bar{v} + \partial_n \bar{v}.v) \\ &= \int_\Omega (-\Delta v + \nabla q + j'(u)).\bar{v}dx + \int_{\Gamma_N} (\sigma(v,q)n + l'(u)).\bar{v}ds - \int_{\Gamma_D} \partial_n \bar{v}.v \end{aligned} \tag{5.27}$$

$$\langle \frac{\partial \mathcal{L}}{\partial p}(\Omega,u,p,v,q), \bar{q}\rangle = \int_\Omega -\bar{q}\text{div}v + \int_{\Gamma_D} \bar{q}n.v \tag{5.28}$$

It has been observed that the equilibrium of $\frac{\partial \mathcal{L}}{\partial v}(\Omega,u,p,v,q)$ and $\frac{\partial \mathcal{L}}{\partial q}(\Omega,u,p,v,q)$ yields the variational formulation associated to the state system (5.24). There exists unique solution (u_Ω, p_Ω) of the weak form of the state system and we can easily verify that:

$$J(\Omega) = \mathcal{L}(\Omega, u_\Omega, p_\Omega, v, q) \,\forall\, (v,q) \in (H^1_{\Gamma_D}(\mathbb{R}^d)^d, L^2(\mathbb{R}^d))$$

By taking the partial derivative of two parts with respect to domain Ω, we have:

$$J'(\Omega)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_\Omega, p_\Omega, v, q)(\theta) \tag{5.29}$$

$$+ \frac{\partial \mathcal{L}}{\partial u}(\Omega, u_\Omega, p_\Omega, v, q)u'_\Omega(\theta) + \frac{\partial \mathcal{L}}{\partial p}(\Omega, u_\Omega, p_\Omega, v, q)p'_\Omega(\theta) \tag{5.30}$$

Hence, if (v_Ω, q_Ω) are solutions of the system of $\frac{\partial \mathcal{L}}{\partial u}(\Omega, u_\Omega, p_\Omega, v, q) = 0$ and $\frac{\partial \mathcal{L}}{\partial p}(\Omega, u_\Omega, p_\Omega, v, q)$ which also called adjoint system written as follows:

$$\begin{cases} -\Delta v + \nabla q &= -j'(u_\Omega) \quad \text{in}\,\Omega \\ \text{div} v &= 0 \qquad\quad \text{in}\,\Omega \\ \sigma(v,q)n &= -l'(u_\Omega) \quad \text{on}\,\Gamma_N \\ v &= 0 \qquad\quad \text{on}\,\Gamma_D \end{cases} \tag{5.31}$$

then the partial differential of objective function with respect to domain Ω can be formulated in a rather simple way:

$$J'(\Omega)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u_\Omega, p_\Omega, v_\Omega, q_\Omega)(\theta) \tag{5.32}$$

Applying Lemma 3, Lemma 4 and noting that $u = v = 0/\Gamma_D$, we have the shape derivative of the objective function in this case:

$$\begin{aligned} J'(\Omega)(\theta) = &\int_{\partial\Omega} \theta.n(j(u) + \nabla u.\nabla v - p\text{div}u - q\text{div}v - f.v)ds \\ &+ \int_{\Gamma_N} \theta.n\left(l(u) - \varphi_N.v + \kappa\frac{\partial(l(u) - \varphi_N.v)}{\partial n}\right)ds \\ &- \int_{\Gamma_D} \theta.n\left(\sigma(v,q)n.u + \sigma(u,p)n.v\right)ds \end{aligned} \tag{5.33}$$

We have computed the shape derivative for objective functionals. Next, we turn to construct an algorithm for shape optimization problem in the context of Stokes system.

5.2 The generic shape optimization algorithm

5.2.1 The gradient method

As suggested by the name, this method relies on the knowledge of the first-order shape derivative $J'(\Omega)$ to produce a descent direction θ for J. We have stated that only the normal component of the deformations is considered and play a role in the shape derivative of objective functions. Actually, the shape derivatives of all the functionals we shall get interested in enjoy a structure of the form:

$$J'(\Omega)(\theta) = \int_{\partial\Omega} w_\Omega \theta.n ds \tag{5.34}$$

where w_Ω is a certain scalar defined over $\partial\Omega$ (depending on the Stokes and the adjoint systems), which makes the computation of the possible deformation field for $J(\Omega)$:

$$\theta_t = -tw_\Omega n \tag{5.35}$$

Indeed, from the expansion (5.1), for $t > 0$ small enough, we have:

$$J(\Omega_{t\theta}) = J(\Omega) - t\int_{\partial\Omega} w_\Omega^2 ds + o(t) < J(\Omega),$$

hence, the new shape $\Omega_{t\theta}$ is obtained from initial form Ω as: $\Omega_{t\theta} = (I + \theta_t)(\Omega)$.

5.2.2 The global algorithm

We are now in position to introduce a *shape gradient algorithm* for the considered problem (5.3).

Algorithm 5: Shape gradient algorithm

n=0, start with an initial guess Ω^0;

for $n = 1, \ldots,$ *untill convergence* **do**

1. Compute the solution $(u_{\Omega^n}, p_{\Omega^n})$ to the Stokes system on Ω^n, and the solution $(v_{\Omega^n}, q_{\Omega^n})$ to the adjoint system (section 5.1.3).
2. Infer a descent direction θ^n for the objective functional following the lines in section 5.2.1.
3. Choose a descent step $\tau^n > 0$ small enough so that $J(\Omega^n_{\tau^n\theta^n}) < J(\Omega^n)$.
4. The new shape is obtained as $\Omega^{n+1} = \Omega^n_{\tau^n\theta^n}$

end

5.3 Shape optimization basing on level set method and mesh adaptation

5.3.1 Description of the level set method for shape optimization

The idea of this method is to combine an implicit domain evolution method with an explicit type of shape representation which have been used in many references: for examples, in the two-dimensional work [YNKN11], the evolution of shapes is tracked on a triangular mesh $\mathcal{D}$ of a working domain D owing to the level set method, and at each iteration of the process, an exact mesh of the current shape Ω is obtained by relocating vertices of $\mathcal{D}$ onto $\partial\Omega$. In [XSLW12], a similar strategy is applied for dealing with the motion of shapes; a computational mesh for any shape arising during the process is moreover constructed by first identifying the intersection points of the implicitly-defined boundary $\partial\Omega$ with the edges of the mesh $\mathcal{D}$ of D, then using them as an input for a Delaunay-based

mesh generation algorithm. In our approach, a computational domain D is defined, and is equipped with an unstructured mesh which is modified at each iteration of the algorithm, in such a way that all the shapes $\Omega^0, \ldots, \Omega^n$ produced during the shape optimization process are explicitly discretized in this mesh, i.e all elements (vertices, edges, faces) of mesh of shape are also as elements of the mesh of D (see figure 5.2, left). The direction of the deformation of each shape Ω^n is inferred from computing the shape gradient $J'(\Omega^n)$ (which depends on solutions of the state and the adjoint systems on Ω^n, see section 5.1.3). The determination and the deformation of the shape Ω^n is related to the level set function ϕ^n defined on D: at each step, ϕ^n is advected by descent direction of Ω^n to obtain the new level set function ϕ^{n+1}. This function leads to define the new shape Ω^{n+1} as the negative subdomain of D (see figure 5.2, right). This strategy presents several advantages:

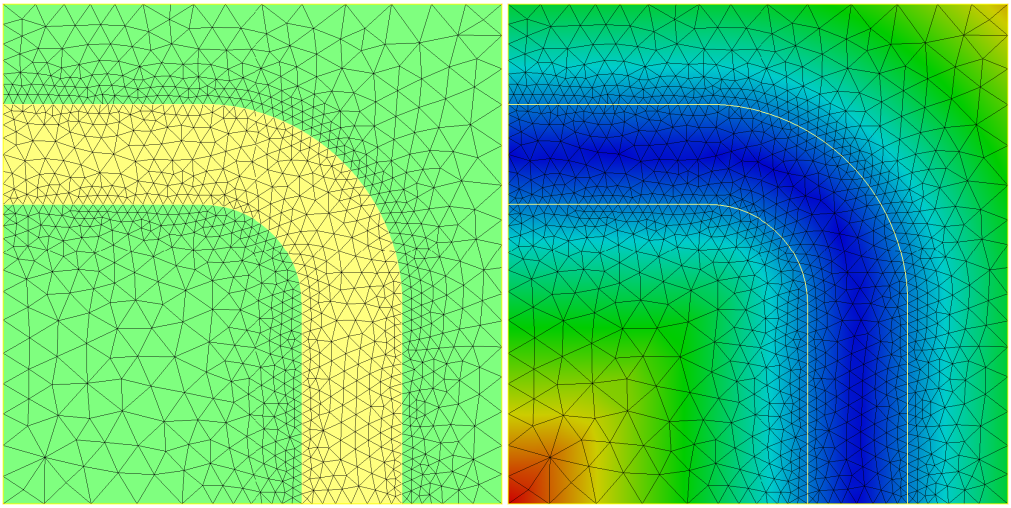


Figure 5.2: *Left: explicit discretisation of shape Ω (yellow zone) in mesh of computational domain D. Right: representation of shape Ω as negative subdomain of a level set function defined on D*

- The explicit discretization of shape Ω in D allows us define and solve the concerned mechanical problem only on a submesh (the negative part). hence the computational time can be reduced, since we know that normally we need to resolve both state and adjoint problem which always are challenges in numerical resolution.
- As a shape is known as a submesh of the computational mesh of D, no projection between different meshes is involved in the computation of a descent direction for the considered objective function of the domain.
- The proposed method does not pose any theoretical difficulty to be extended to the three-dimensional case.

Let us now recast the algorithm 5 in the context of level set method:

Algorithm 6: Shape gradient algorithm (in combination with level set method)

n=0, start with an initial guess Ω^0;

for $n = 1, \ldots,$ *untill convergence* **do**

1. Compute the solution $(u_{\Omega^n}, p_{\Omega^n})$ to the Stokes system on Ω^n, and the solution $(v_{\Omega^n}, q_{\Omega^n})$ to the adjoint state which is constructed in section (5.1.3)
2. Infer a descent direction θ^n for the objective functional following the lines of section (5.2.1)
3. Choose a descent step $\tau^n > 0$,and solve the Hamilton-Jacobi equation:

$$\begin{cases} \phi_t^{n+1}(x,t) + \theta^n.\nabla\phi^{n+1}(x,t) & = 0 \quad \text{on } D \times [0,\tau^n] \\ \phi^{n+1}(x,0) & = \phi^n(x) \quad \text{on } D \end{cases} \tag{5.36}$$

to obtain the new level set function ϕ^n.

4. The new shape is obtained as $\Omega^{n+1} := \{x \in D, \phi^{n+1}(x) < 0\}$.

end

5.3.2 Model of fluid mechanics in the context of Stokes system

Let us now come back to the problem (5.3). Suppose that the boundary of Ω is decomposed by three disjoint parts:

$$\partial\Omega = \Gamma \cup \Gamma_N \cup \Gamma_D$$

where $\Gamma \neq \emptyset$ is the variable part during the optimization process, Γ_D is a fixed part on where we impose an inlet (a constant velocity or a profile) and Γ_N is also a subset of fixed part where we apply one condition for the outlet (one type of Newmann condition, see chapter 1). As consequence, the velocity u of the fluid is the solution of the following systems:

$$\begin{cases} -\nu\Delta u + \nabla p & = f \text{ in } \Omega \\ \text{div} u & = 0 \text{ in } \Omega \\ \sigma(u,p)n & = \varphi_N \text{ on } \Gamma_N \\ u & = u_D \text{ on } \Gamma_D \\ u & = 0 \text{ on } \Gamma \end{cases} \tag{5.37}$$

According to the previous requirements, the considered set $\mathcal{U}_{ad}$ of admissible shapes is defined as:

$$\mathcal{U}_{ad} = \{\Omega \text{ bounded and Lipschitz, } \Gamma_D \cup \Gamma_N \subset \partial\Omega\}, \tag{5.38}$$

so that the set Θ_{ad} of admissible variations is:

$$\Theta_{ad} = \{\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) : \theta = 0 \,\text{on}\, \Gamma_D \cup \Gamma_N\}.$$

Given a objective function $J(\Omega)$, in fact the minimization problem:

$$\inf_{\Omega \in \mathcal{U}_{ad}} J(\Omega) \tag{5.39}$$

usually requires some smoothness or geometrical or topological constraints, for example, one could require the volume $V(\Omega) := \int_\Omega dx$ or the perimeter $P(\Omega) := \int_{\partial\Omega} ds$ of an admissible shape Ω, to be equal (or lower or equal) to a prescribed upper bound. Here, we limit ourselves to consider the problem with the constraint on the volume of shapes: a variant of (5.39) turns out to be a minimization problem:

$$\inf_{\Omega \in \mathcal{U}_{ad}} (J(\Omega) + lV(\Omega)) \tag{5.40}$$

where $\mathcal{U}_{ad}$ is defined by (5.38) and $l > 0$ is a positive Lagrange multiplier.

Other regularized variants of (5.39) have been proposed in [Che75, Cha03] for such existence theories or more efficient algorithms for handling constraints such as the Method of Moving Asymptotes described in [Sva87], or the Method of Feasible Directions in [VM73], but they are not dealt in the scope of this thesis.

Note that: Result (5.2) of Lemma (3) and formulation (5.34) lead to write:

$$\left(J'(\Omega) + lV'(\Omega)\right)(\theta) = \int_{\partial\Omega} (w_\Omega + l)\theta.n$$

This implies that for the optimization problem under a volume constraint, the computation of the scalar function w_Ω in (5.35) requires adding the Lagrange multiplier l. In order to simplify the notation, the notation $J(\Omega)$ of the objective function will hereafter have the meaning of $(J(\Omega) + lV(\Omega))$ while w_Ω stands for $(w_\Omega + l)$.

5.3.3 Computation of a descent direction

From a given shape $\Omega \in \mathcal{U}_{ad}$, a descent direction $V \in \Theta_{ad}$ for the considered objective function $J(\Omega)$ is computed on a whole mesh $\mathcal{D}_\Omega$ of D which encloses an explicit discretization of Ω.

The generic expression (5.35) for the shape derivative of J suggests the immediate choice:

$$\forall x \in \partial\Omega,\, V(x) = -w_\Omega(x)n(x). \tag{5.41}$$

As we have seen, w_Ω depends on the solution of Stokes and adjoint systems posed on Ω, which can be accurately solved on the submesh T_Ω of $\mathcal{D}_\Omega$, using the finite element method. Unfortunately, the choice (5.35) for a descent direction turns out to be hazardous for two independent reasons:

- Formula (5.35) is only defined on $\partial\Omega$, whereas we are in search for a velocity field V which is defined at least in a vicinity of $\partial\Omega$, as well to comply with the requirement $V \in \Theta_{ad}$ as to use V in the context of the level set method.
- The scalar field w_Ω usually depends on the derivatives of the Stokes solution and the adjoint solution, which may be quite irregular - in the theoretical framework as well as when it comes to their numerical approximation. This may jeopardize the numerical stability of the process.

As advocated by [Bur03, dG06] (see also in chapter 2) an efficient way to address both problems at the same time consists in using the gradient of J as a descent direction associated with a different scalar product instead of canonical one of $L^2(\Gamma)$.

More accurately, $\alpha > 0$ being a small "extension - regularization" parameter, let us consider the functional space:

$$H^1_{\Gamma_D \cup \Gamma_N}(D) = \{w \in H^1(D), w = 0 \,\text{on}\, \Gamma_D \cup \Gamma_N\}$$

and let $\widetilde{w} \in H^1_{\Gamma_D \cup \Gamma_N}$ be the unique solution to the variational problem (see [Bur03] for alternative choices):

$$\forall w \in H^1_{\Gamma_D \cup \Gamma_N(D)}, \int_D (\widetilde{w}w + \alpha \nabla \widetilde{w}.\nabla w dx) = \int_\Gamma w_\Omega w ds = J'(\Omega)(wn) \tag{5.42}$$

Consider now the choice:

$$\forall x \in D, \widetilde{V}(x) = \widetilde{w} n(x)$$

where n is an extension to D of the normal vector field to $\partial\Omega$. Using the asymptotic expansion (5.1) we can show that $\widetilde{V}$ is still a descent direction for J (for t small enough: $J(\Omega_{t\widetilde{V}}) = J(\Omega) - t\int_\Gamma \widetilde{w}^2 w_\Omega n ds + o(t)$). However, $\widetilde{V}$ intrinsically enjoys more regularity than V owing to the classical regularity theory for elliptic equations, and is inherently defined on the whole domain D.

In the numerical setting, $\widetilde{w}$ is easily computed by solving (5.42) with the classical finite element method carried out on mesh $\mathcal{D}_\Omega$, after computing w_Ω (the discretization of the right-hand side being straightforward since the computational mesh $\mathcal{D}_\Omega$ encloses an explicit discretization of $\partial\Omega$).

5.3.4 The proposed algorithm

We are now in position to outline the proposed general strategy for handling mesh evolution in the context of shape optimization.

Starting with an initial shape Ω^0, and a simplicial mesh $\mathcal{D}_{\Omega^0}$ of D in which Ω^0 is explicitly discretized. For $n = 0, ...$ *untill convergence*, the current shape Ω^n is known via a submesh T_{Ω^n} of $\mathcal{D}_{\Omega^n}$ where T_{Ω^n} is a mesh of Ω^n.

1. Generate the level set function ϕ^n (as the signed distance function) to Ω^n on the whole mesh $\mathcal{D}_{\Omega^n}$ of D.
2. Compute the value of the scalar field w_{Ω^n} appearing in the shape derivative of the considered functional in (5.35). This may involve one, or several finite element analyses for solving the Stokes and adjoint systems, to be held on the part T_{Ω^n} of the mesh $\mathcal{D}_{\Omega^n}$ corresponding to Ω^n. The quantity w_{Ω^n} is defined only on $\partial\Omega^n$, i.e. in the numerically setting, on the discretization of $\partial\Omega^n$ which explicitly appears in both T_{Ω^n} and D_{Ω^n}.
3. Extend w_{Ω^n} to a vector field V^n defined on the whole mesh $\mathcal{D}_{\Omega^n}$ of D, following the lines in section 5.3.3
4. Choose a descent step $\tau^n > 0$, and solve the following level set advection equation on $\mathcal{D}_{\Omega^n}$:

$$\begin{cases} \partial_t\phi(x,t) + V^n(x).\nabla\phi(x,t) \text{ for } (x,t) \in D \times (0,\tau^n) \\ \phi(x,0) = \phi^n(x) \text{ for } x \in D \end{cases} \tag{5.43}$$

   This produces a new level set function $\phi^{n+1} := \phi(x,\tau^n)$ associated to the new shape Ω^{n+1}.
5. Generate the meshed representation of Ω^{n+1} from the set of data $\mathcal{D}_{\Omega^n}, \phi^{n+1}$. A new mesh $\mathcal{D}_{\Omega^{n+1}}$ of D is produced through an intermedia "ill-shaped" mesh $\widetilde{\mathcal{D}}_{\Omega^{n+1}}$ in which Ω^{n+1} is explicitly discretized. $\widetilde{\mathcal{D}}_{\Omega^{n+1}}$ is remeshed by local modification operators to obtain the resulting "well-shaped" mesh $\mathcal{D}_{\Omega^{n+1}}$ for better quality and geometric approximation (see Chapter 3).
6. Evaluate $J(\Omega^{n+1})$. If $J(\Omega^{n+1}) < J(\Omega^n)$, Ω^{n+1} is retained as the new shape; else $\Omega^{n+1} = \Omega^n$. Then, go back to step 4, decreasing the chosen value for the time step.

5.4 Numerical examples

We consider the situation depicted in figure 5.3: a flow with the constant viscosity ν travels throughout a domain of elbow Ω with three parts of the boundary:

$$\partial\Omega = \Gamma \cup \Gamma_D \cup \Gamma_N$$

We endow a parabolic velocity profile u_D on the inlet boundary Γ_D and assume that the flow comes out at the boundary Γ_N with the free-out condition. This domain Ω is the optimization variable of the problem where Γ_D, Γ_N are the subset of fixed part of the boundary and only the part Γ is variable during the optimization process. We are interested in the minimization the energy dissipation in fluid, then the object functional $J(\Omega)$ defined as:

$$J(\Omega) = \int_\Omega \nu|\nabla u|^2 dx$$

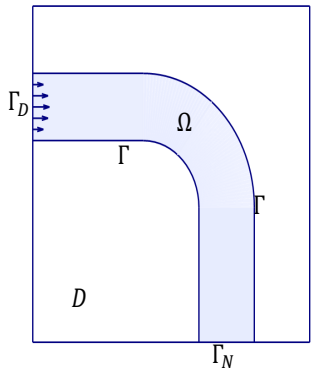


Figure 5.3: *The model of problem in the elbow case test.*

where u is the solution of the following the Stokes systems:

$$\begin{cases} -\nu\Delta u + \nabla p &= f \text{ in } \Omega \\ \text{div} u &= 0 \text{ in } \Omega \\ u &= 0 \text{ on } \Gamma \\ u &= u_D \text{ on } \Gamma_D \\ \sigma(u,p)n &= 0 \text{ on } \Gamma_N \end{cases} \tag{5.44}$$

The domain Ω is enclosed in a fixed working domain D. Applying the results of shape derivative in the section 5.1.3 for $j(u) = \nu\nabla u.\nabla u, f = 0$ and note that only the part Γ of $\partial\Omega$ is variable during the optimization process, we have the computation the shape derivative of $J(\Omega)$:

$$J'(\Omega) = \int_\Gamma (\nu\nabla u.\nabla u + \nu\nabla u.\nabla v - p\text{div}v - q\text{div}u)\theta.nds - \int_\Gamma (\sigma(u,p)n.\partial_n v + \sigma(v,q)n.\partial_n u)\theta.nds \tag{5.45}$$

The second integral can be reduced as follows:

$$\begin{aligned} \int_\Gamma (\sigma(u,p)n.\partial_n v + \sigma(v,q)n.\partial_n u)\theta.nds &= \int_\Gamma ((\nu\nabla u - pI_d)n.\partial_n v + (\nu\nabla v - qI_d)n.\partial_n u) \\ &= \int_\Gamma (2\nu\partial_n u.\partial_n v - pn.\partial_n v - qn.\partial_n u) \end{aligned} \tag{5.46}$$

Otherwise, $\partial_\tau u = \partial_\tau v = 0$ (as $u = v = 0/\Gamma$). Hence, we have:

$$\nabla u.\nabla v = \partial_n u.\partial_n v + \partial_\tau u.\partial_\tau v = \partial_n u.\partial_n v,$$

$$\text{div}u = n.\partial_n u + \tau.\partial_\tau u = n.\partial_n u,$$

$$\text{div}v = n.\partial_n v + \tau.\partial_\tau v = n.\partial_n v.$$

Finally, we obtain the following result:

$$J'(\Omega) = \int_{\Gamma} (\nu \nabla u.\nabla u - \nu \nabla u.\nabla v)\theta.n ds \tag{5.47}$$

So, the scalar function w_Ω (on Γ) in step 1 is:

$$w_\Omega = \nu(\nabla u.\nabla u - \nabla u.\nabla v)$$

where u is the solution of Stokes systems (5.37) and v is solution of adjoint systems constructed by (5.31)

w_Ω is extended to obtain the descent direction V defined on the whole mesh $\mathcal{D}$. As explained in the section 5.3.2, we will consider this minimization problem with a volume constraint incorporated under the form of a penalization by a fixed Lagrange multiplier l. In this case test, we set $l = 2.5$ and 300 iterations of the algorithm of section 5.3.4 are performed. Each mesh $\mathcal{D}_\Omega$ of D arising in the course of the process has approximately 1200 vertices, and the whole computation takes about 10 minutes. The detail implements in each iteration are displayed on figure 5.4 and the convergence history for the aggregated objective functional $(J(\Omega) + lV(\Omega))$ is reported on figure 5.5.

We have reduced considerably the energy dissipation:

$$\left| \frac{J(\Omega_{final}) - J(\Omega_0)}{J(\Omega_0)} \times 100 \right| \simeq 40\%$$

It can be observed that in this case test, the final shape is the domain connected by the "almost straight" lines between input and output parts. This result is in good agreement with the one obtained in [BP03, DMZ08b, Pri08, CG09].

5.5 Conclusion

This chapter has given a numerical scheme for shape-topology optimization in fluid mechanics. One numerical example in 2D for Stokes problems with the minimizing energy dissipation has been presented. Regarding the perspectives of this work, we would like to mention a few options:

- Investigate the different test cases with Stokes problem and dissipated object function in 2D and 3D, for examples the test cases such as in [BP03, DMZ08b, CG09].
- Extend the proposed scheme in the context of Navier-Stokes problem like in [DMZ08a] and may be combine with the different objective functions as the suggest of maximum permeability in [ZL08].

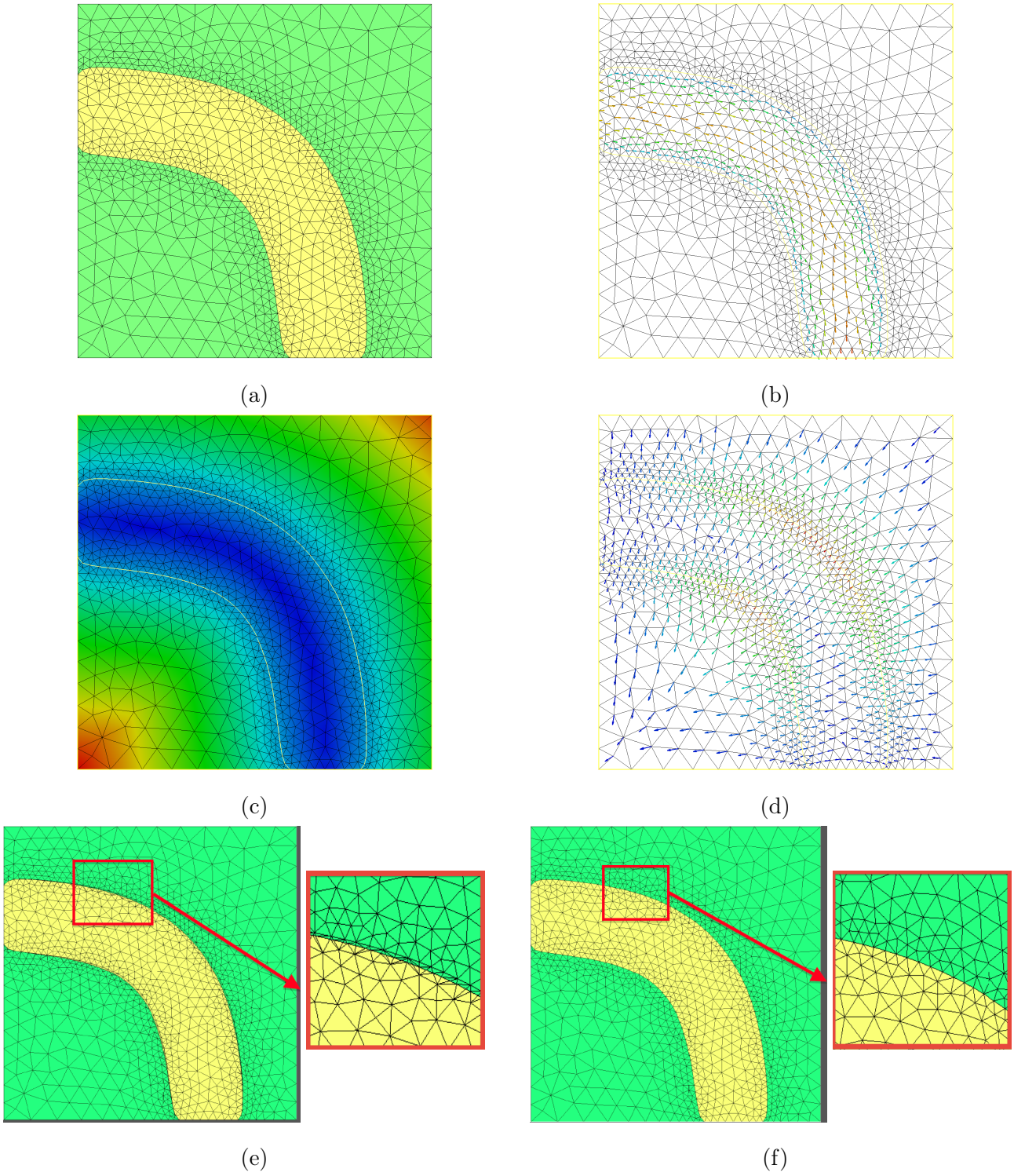


Figure 5.4: *Implements in n^{th}-iteration. (a): Explicit discretization of shape Ω^n in mesh $\mathcal{D}_{\Omega^n}$, (b): Velocity field of Stokes problem on sub-mesh T_{Ω^n} of shape Ω^n, (c): Shape-domain Ω^n in mesh $\mathcal{D}_{\Omega^n}$ corresponding level set fuction ϕ^n, (d): Velocity field V^n on $\mathcal{D}_{\Omega^n}$ for advection of ϕ^n, (e): Explicit discretization of zero-level set of ϕ^{n+1}, the obtained mesh $\widetilde{\mathcal{D}}_{\Omega^{n+1}}$ is very ill-shaped, (f): High-quality mesh $\mathcal{D}_{\Omega^{n+1}}$ in which the new shape Ω^{n+1} is explicitly discretized.*

.

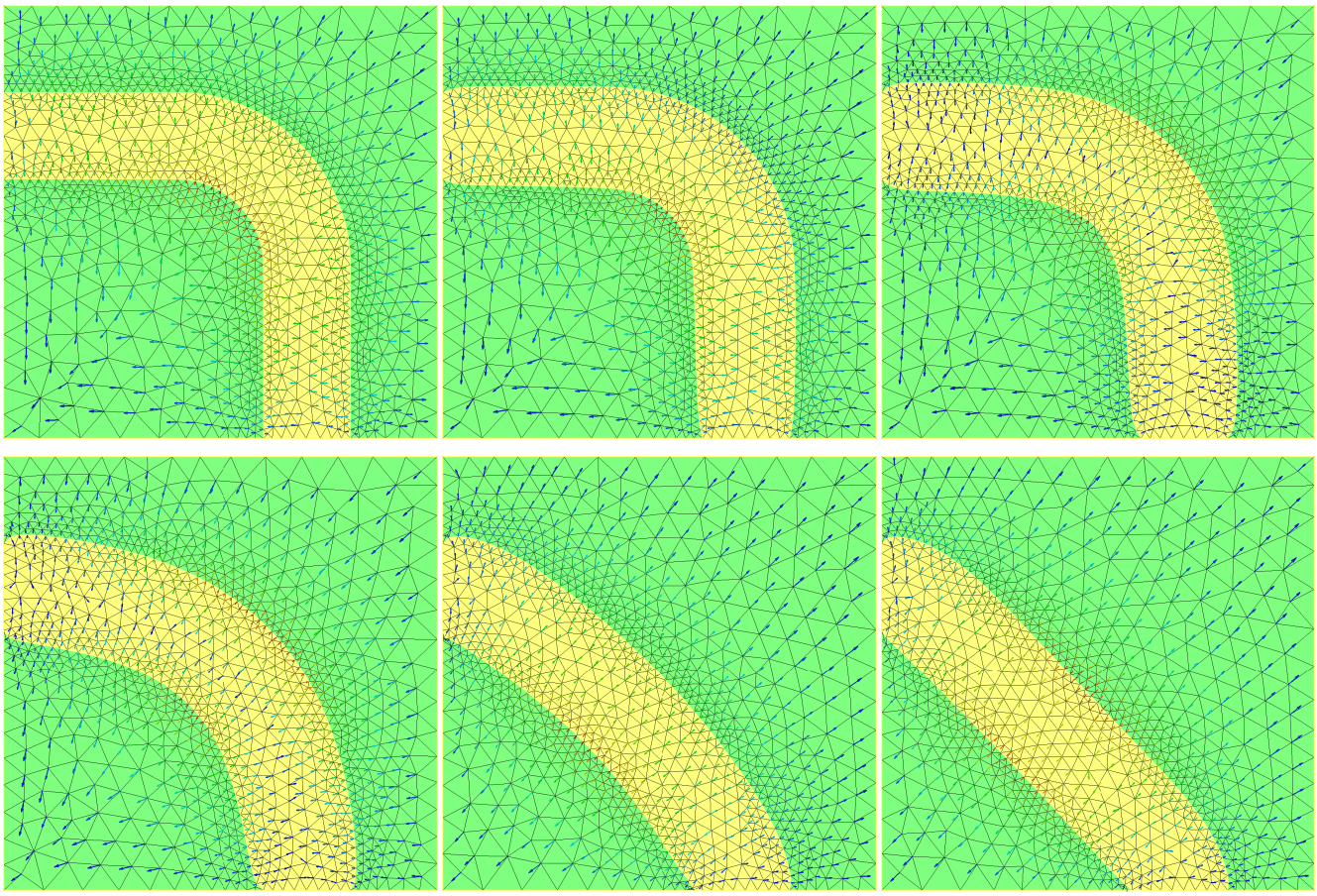


Figure 5.5: *Deformation process of domain in the case test of elbow. From top to bottom, left to right:* $1^{th}, 50^{th}, 70^{th}, 85^{th}, 100^{th}$ *and* 150^{th} *iterations of the case test.*

.

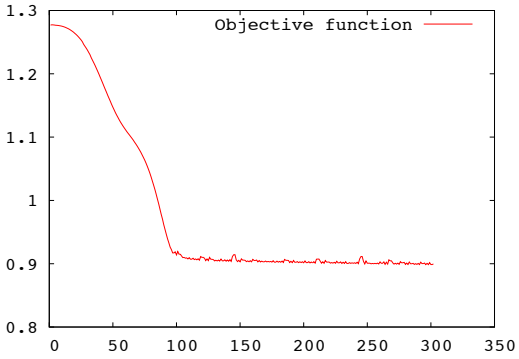


Figure 5.6: *Convergence history for the case test to* 300^{th} *iteration.*

– Our proposed scheme which is based on a combination of classical shape derivative, level set method and mesh adaptation process has been utilized successfully in the elastic structure [Dap13]. This study has demonstrated its efficiency and its reliability in fluid mechanics problem. We could expect a promising application of the present scheme for shape-topology optimization in other physical problems (thermal structure,...).

Bibliography

[ABFJ97] G. Allaire, E. Bonnetier, G. Francfort, and F. Jouve. Shape optimization by the homogenization method. *Numer. Math.*, 76:27–68, 1997.

[ABPV12] R. Araya, G. R. Barrenechea, A. H. Poza, and F. Valentin. Convergence analysis of a residual local projection finite element method for the Navier-Stokes equations. *SIAM J. Numer. Anal.*, 50(2):669–699, 2012.

[ADH11] F. Alouges, A. DeSimone, and L. Heltai. Numerical strategies for stroke optimization of axisymmetric microswimmers. *Math. Models Methods Appl. Sci.*, 21(2):361–387, 2011.

[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.

[AJT02] G. Allaire, F. Jouve, and A.M. Toader. A level-set method for shape optimization. *C.R. Acad. Sci. Paris, Serie I*, 334:1125–1130, 2002.

[AJT04] G. Allaire, F. Jouve, and A.M. Toader. Structural optimization using shape sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194:363–393, 2004.

[All02] G. Allaire. *Shape optimization by the homogenization method.* Springer Verlag, New York, 2002.

[All06] G. Allaire. *Conception optimale de structures, Mathematiques et Applications*, volume 58. Springer, Heidelberg, 2006.

[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.

[AO97] M. Ainsworth and J.T. Oden. A posteriori error estimation in finite element analysis. *Comput. Meths. Appl. Mech. Engrg.*, 142(1-88), 1997.

[Ape99] T. Apel. Anisotropic finite elements: local estimates and applications. *Advances in Numerical Mathematics. B. G. Teubner, Stuttgart*, 8, 1999.

[APGHS08] N. Aage, T.H. Poulsen, A. Gersborg-Hansen, and O. Sigmund. Topology optimization of large scale Stokes flow problems. *Struct Multidisc Optim*, 35:175–180, 2008.

[APV14] R. Araya, A. H. Poza, and Frédéric Valentin. An adaptive residual local projection finite element method for the Navier–Stokes equations. *Adv Comput Math*, 2014.

[AQR06a] V. Agoshkov, A. Quarteroni, and G. Rozza. A mathematical approach in the design of arterial bypass using unsteady Stokes equations. *Journal of Scientific Computing*, 28(2-3):139–161, 2006.

[AQR06b] V. Agoshkov, A. Quarteroni, and G. Rozza. Shape design in aorto-coronaric bypass anastomoses using perturbation theory. *SIAM Journal on Numerical Analysis*, 44(1):367–384, 2006.

[BA76] I. Babuška and A. K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13(2):214–226, April 1976.

[Bae01] E. Baensch. Finite element discretization of the Navier-Stokes equations with a free capillary surface. *Numer. Math.*, 88:203–235, 2001.

[BBD+06] D. Boffi, F. Brezzi, L. F. Demkowicz, R. G. Durán, Richard S. Falk, and M Fortin. *Mixed Finite Elements, Compatibility Conditions, and Applications*. Springer, 2006.

[BCdV14] A. Bernard-Champmartina and F. de Vuyst. A low diffusive Lagrange-remap scheme for the simulation of violent air–water free-surface flows. *Journal of Computational Physics*, 274:19–49, 2014.

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

[Ben95] M.P. Bendsøe. Methods for optimization of structural topology. In *Shape and Material*. Springer- Verlag, New York, 1995.

[BF84] V. Braibant and C. FIeury. Shape optimal design using B-splines. *Computer Methods in Applied Mechanics and Engineering*, 44(3):247–267, 1984.

[BFH98] H. Borouchaki, P. Frey, and F. Hecht. Mesh gradation control. *International Journal for Numerical Methods in Engineering*, 43:1143–1165, 1998.

[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, January 2010.

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

[BK88] M.P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71:197–224, 1988.

[BKZ92] J. U. BrackBill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100:335–354, 1992.

[BNV06] A. Bermudez, M.R. Nogueiras, and C. Vasquez. Numerical analysis of convection-diffusion-reaction problems with higher order characteristics/finite elements. part ii : fully discretized scheme and quadrature formulas. *SIAM Journal on Numerical Analysis*, 44(5):1829–1853, 2006.

[BP03] T. Borrvall and J. Petersson. Topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Fluids*, 41:77–107, 2003.

[BS03] M.P. Bendsøe and O. Sigmund. *Topology Optimization, Theory, Methods and Applications*. Springer Verlag, Berlin Heidelberg, 2nd edition, 2003.

[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.

[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.

[CDF08] A. Claisse, V. Ducrot, and P. Frey. Levelsets and anisotropic mesh adaptation. *Discrete and Continuous Dynamical Systems - Series A (DCDS-A)*, 23(1-2):165–183, 2008.

[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.

[Céa86] J. Céa. Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût. *Math. Model. Num.*, 3(20):371–420, 1986.

[CG09] V.J. Challis and J.K. Guest. Level set topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Engineering*, 79:1284–1308, 2009.

[Cha03] A. Chambolle. A density result in two-dimensional linearized elasticity and applications. *Arch. Rational Mech. Anal.*, 167:211–233, 2003.

[Che75] D. Chenais. On the existence of a solution in a domain identification problem. *J. Math. Anal. Appl.*, 52:189–289, 1975.

[CHMO96] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, 124(449-464), 1996.

[Cia78] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North Holland Publishing Company, 1978.

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

[CL98] L. Chen and Y. Li. A numerical method for two-phase flows with an interface. *Environmental Modelling Software*, 13(247-255), 1998.

[Dap13] C. Dapogny. *Shape optimization, level set methods on unstructured meshes and mesh evolution.* PhD thesis, Universite Pierre et Marie Curie, 2013.

[DDF14] C. Dapogny, C. Dobrzynski, and P. Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of Computational Physics*, 262:358–378, 2014.

[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.

[DF08] C. Dobrzynski and P. Frey. Anisotropic Delaunay mesh adaptation for unsteady simulations. pages 177–194, Sandia National Labs, Pittsburgh, PA, 2008. In Proc. 17th Int. Meshing Roundtable.

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

[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.

[DK91] A. Doi and A. Koide. An efficient method of triangulating equivalued surfaces by using tetrahedral cells. *IEICE Transactions on Communications and Electronics Information Systems*, E74-D(1):214–224, 1991.

[DMZ08a] X. Duan, Y. Ma, and R. Zhang. Shape-topology optimization for Navier-Stokes problem using variational level set method. *Journal of Computational and Applied Mathematics*, 222:487–499, 2008.

[DMZ08b] X.B Duan, Y.C Ma, and R. Zhang. Shape-topology optimization of Stokes flow via variational level set method. *Applied Mathematics and Computation*, 202:200–209, 2008.

[dSMN+04] F.S. de Sousa, N. Mangiavacchi, L.G. Nonato, A. Castelo, M.F. Tome, 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. *Journal of Computational Physics*, 198:469–499, 2004.

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

[DVB+02] J. Dompierre, M.G. Vallet, Y. Bourgault, M. Fortin, and W.G. Habashi. Anisotropic mesh adaptation: towards user independent, mesh-independent and solver independent cfd. iii. unstructured meshes. *International Journal for Numerical Methods in Fluids*, 39(8):675–702, 2002.

[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.

[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.

[Evg06] A. Evgrafov. Topology optimization of slightly compressible fluids. *ZAMM - Z. Angew. Math. Mech*, 86(1):46–62, 2006.

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

[FB96] P. Frey and H. Borouchaki. Texel : triangulation de surfaces implicites. partie i : aspects théoriques. Technical Report 3066, INRIA, 1996.

[FG08] P. Frey and P. L. George. *Mesh generation. Application to finite elements.* Wiley, 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.

[Fou02] G. Fourestey. Stabilité des méthodes de Lagrange-Galerkin du premier et du second ordre. Technical Report 4505, INRIA Rocquencourt, 2002.

[FP01] L. Formaggia and S. Perotto. New anisotropic a priori error estimates. *Numercical Mathematic*, 89(4):641–667, 2001.

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

[Fre08] P. J. Frey. *A differential approach to mesh generation*, volume 9 of *Differential Geometry: Theory and Applications in "Series in Contemporary Applied Mathematics"*, pages 222–292. World Scientific, 2008.

[FWE70] C.O. Frederick, Y.C. Wong, and F.W. Edge. Two-dimensional automatic mesh generation for structural analysis. *International Journal for Numerical Methods in Engineering*, 2:133–144, 1970.

[Geo71] J.A. George. *Computer Implementation of the Finite Element Method.* PhD thesis, Stanford University, 1971.

[GLM06] V. Girault, H. López, and B. Maury. One time-step finite element discretization of the equation of motion of two-fluid flows. *Numercical Methods Partial Differential Equations*, 22(3):680–707, 2006.

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

[GP06] J.K. Guest and J.H. Prévost. Topology optimization of creeping fluid flows using a Darcy-Stokes finite element. *International Journal for Numerical Methods in Engineering*, 66:461–484, 2006.

[GR86] V. Girault and P.A Raviart. *Finite Element Methods for Navier-Stokes Equations*. Springer Verlag, 1986.

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

[GRR06] S. Grob, V. Reichelt, and A. Reusken. A finite element based level set method for two-phase incompressible flows. *Journal Computing and Visualization in Science*, 9(4):239–257, November 2006.

[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.

[Gun03] M.D. Gunzburger. *Perspectives in Flow Control and Optimization*. Advances in Design and Control. SIAM, 2003.

[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.

[Had07] J. Hadamard. Mémoire sur le problème d'analyse relatif à l'équilibre des plaque élastiques encastrées. Technical report, Bull. Soc. Math. France, 1907.

[HC05] S.H. Ha and S. Cho. Topological shape optimization of heat conduction problems using level set approach. *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, 48:67–88, 2005.

[HC08] S.H. Ha and S. Cho. Level set-based topological shape optimization of nonlinear heat conduction problems. *Numerical Heat Transfer, part B*, 54:454–475, 2008.

[HG86] R.T. Haftka and R.V. Grandhi. Structural shape optimization - a survey. *Communications in numerical methods in engineering*, 57(91-106), 1986.

[HMY03] S.Y. Han, J.S. Maeng, and D.H. Yoo. Shape optimization of cutoff in a multiblade fan/scroll system using response surface methodology. *Numerical Heat Transfer, part B*, 43:87–98, 2003.

[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.

[HP05] A. Henrot and M. Pierre. *Variation et optimisation de formes, une analyse géométrique*. Springer, 2005.

[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.

[Hua05] W. Huang. Metric tensors for anisotropic mesh generation. *Journal of Computational Physics*, 204:633–665, 2005.

[Hys06] S. Hysing. A new implicit surface tension implementation for interfacial flows. *International Journal for Numerical Methods in Fluids*, 51:659–672, 2006.

[KHC09] M.G. Kim, S.H. Ha, and S. Cho. Level set–based topological shape optimization of nonlinear heat conduction problems using topological derivatives. *Mech. Based Design Struct. Machines*, 37:550–582, 2009.

[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 and L. Amaya-Bower. Single bubble rising dynamics for moderate Reynolds number using Lattice Boltzmann method. *Journal of Computational Physics*, 39:1191–1207, 2010.

[LC87] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Comput. Graph*, 21(4):163–169, 1987.

[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.

[LSQX99] Q. Li, G.P. Steven, O.M. Querin, and Y.M. Xie. Shape and topology design for heat conduction by evolutionary structural optimization. *International Journal of Heat and Mass Transfer*, 42:3361–3371, 1999.

[Mer98] R.A Meric. Shape design sensitivity analysis and optimization for nonolinear heat and electric conduction problems. *Numerical Heat Transfer part A*, 34:185–203, 1998.

[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.

[MOS92] W. Mulder, S. Osher, and J. A. Sethian. Computing interface motion in compressible gas dynamics. *Journal of Computational Physics*, 100:209–228, 1992.

[MP04] B. Mohammadi and O. Pironneau. Shape optimization in fluid mechanics. *Annual Review of Fluid Mechanics*, 36:255–279, 2004.

[MP10] B. Mohammadi and O. Pironneau. Applied shape optimization for fluids. In *Numerical Mathematics and Scientific Computation*. Oxford University Press, 2nd edition, 2010.

[MPS88] K.W. Morton, A. Priestley, and E. Suli. Stability of the Lagrange-Galerkin method with non-exact integration. *RAIRO-Modélisation mathématique et analyse numérique*, 22(4):625–653, 1988.

[MS76] F. Murat and S. Simon. *Etudes de problèmes d'optimal design*, pages 54–62. Springer Verlag, Berlin, 1976. Lecture Notes in Computer Science 41.

[NGK74] S. Narayanan, L.H.J. Goossens, and N.W.F. Kossen. Coalescence of two bubbles rising in line at low Reynolds numbers. *Chemical Engineering Science*, 29:2071–2082, 1974.

[NPR06] E. Nobile, F. Pinto, and G. Rizzetto. Geometric parameterization and multiobjective shape optimization of convective periodic channels. *Numerical Heat Transfer, part B*, 50:425–453, 2006.

[OOB06] L.H. Olesen, F. Okkels, and H. Bruus. A high-level programming-language implementation of topology optimization applied to steady-state Navier-Stokes flow. *International Journal for Numerical Methods in Engineering*, 65(7):975–1001, 2006.

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

[PCL03] K. Park, D.H. Choi, and K.S. Lee. Numerical shape optimization for high performance of a heat sink with pin-fins. *Numerical Heat Transfer, part A*, 46:909–927, 2003.

[Pir73] O. Pironneau. On optimum profiles in Stokes flow. *Journal of Fluid Mechanics*, 59(1):117–128, 1973.

[Pir74] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.

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

[Pir84] O. Pironneau. *Optimal Shape Design for Elliptic Systems*. Springer, 1984.

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

[PLT92] O. Pironneau, J. Liou, and T. Tezduyar. Characteristic-Galerkin and Galerkin/least-squares space-time formulations for the advectiondiffusion equation with time-denpendent domains. *Computer Methods in Applied Mechanics and Engineering*, 100(117-141), 1992.

[POTTP06] S. Painchaud-Oullet, C. Tribes, J.Y. Trepanier, and D. Pelletier. Airfoil shape optimization using a nonuniform rational b-spline parametrization under thickness constraint. *IAAA*, 44(10):2170–2178, 2006.

[Pri08] Y. Privat. *Quelques problèmes d'optimisation de formes en sciences du vivant*. PhD thesis, Université Henri Poincaré, Nancy-I, 2008.

[QR03] A. Quarteroni and G. Rozza. Optimal control and shape optimization of aorto-coronaric bypass anastomoses. *Mathematical Models and Methods in Applied Sciences*, 13(12):1801–1823, 2003.

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

[Saa03] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[Set96] J.A. Sethian. *Level Set Methods*. Cambridge University Press, Cambridge, 1996.

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

[She94] J.R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. School of Computer Science, Carnegie Mellon University, Pittsburgh, 1994.

[She12] J.R. Shewchuk. Lecture notes on Delaunay mesh generation, available at http://www.cs.berkeley. edu/~jrs/meshpapers/delnotes.pdf, 2012.

[Sim80] J. Simon. Differentiation with respect to the domain in boundary value problems. *Numer. Funct. Anal. Optim.*, 2:649–687, 1980.

[SK96] A. Sluzalec and M. Kleiber. Shape sensitivity analysis for nonlinear steady-state heat conduction problems. *International Journal of Heat anh Mass Transfer*, 39:2609–2613, 1996.

[SSO94] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to in compressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

[Sul98] E. Suli. Convergence and nonlinear stability of the Lagrange-Galerkin method for the Navier-Stokes equations. *Numercical Mathematic*, 53:459–483, 1998.

[Sva87] K. Svanberg. The method of moving asymptotes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.

[SW00] J.A. Sethian and A. Wiegmann. Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics*, 163(2):489–528, 2000.

[SZ92] J. Sokolowski and J.-P. Zolesio. *Introduction to Shape Optimization; Shape Sensitivity Analysis*, volume 16 of *Series in Computational Mathematics*. Springer, Heidelberg, 1992.

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

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

[TBE+01] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y. J. Jan. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169:708–759, 2001.

[TCC+00] M.F. Tome, A. Castelo, J.A. Cuminato, N. Mangiavacchi, and S. McKee. GENSMAC3D: a numerical method for solving unsteady three-dimensional free surface flows. *Journal of Computational Physics*, 157:441–472, 2000.

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

[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.

[VG04] E. Ramm V. Gravemeier, W.A. Wall. 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.

[VHM91] M.G. Vallet, F. Hecht, and B. Mantel. Anisotropic control of mesh generation based upon a Voronoï type method. *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, 1991.

[VM73] G.N. Vanderplaats and F. Moses. Structural optimization by methods of feasible directions. *Computers Structures*, 3:739–755, 1973.

[WKB07] N. Wiker, A. Klarbring, and T. Borrvall. Topology optimization of regions of Darcy and Stokes flow. *International Journal for Numerical Methods in Engineering*, 69:1374–1404, 2007.

[WWG03] M.Y. Wang, X.M. Wang, and D.M. Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192:227–246, 2003.

[XSLW12] Q. Xia, T. Shi, S. Liu, and M. Y. Wang. A level set solution to the stress-based structural shape and topology optimization. *Computers and Structures*, 90-91(55-64), 2012.

[YJ07] G. Yang and P. Jiang. SSOR and ASSOR preconditioners for Block-Broyden method. *Applied Mathematics and Computation*, 188:194–205, 2007.

[YNKN11] S. Yamasaki, T. Nomura, A. Kawamoto, and S. Nishiwaki. A level set-based topology optimization method targeting metallic waveguide design problems. *International Journal for Numerical Methods in Engineering*, 87:844–868, 2011.

[YPG01] M. Younes, A. Potiron, and A. Genetic. Algorithm for the shape optimization of parts subjected to thermal loading. *Numerical Heat Transfer part A*, 39:449–470, 2001.

[Zhi12] S. Zhi. Second order modified method of characteristics mixed defect-correction finite element method for time dependent Navier-Stokes problems. *Numerical Algorithms*, 59(2):271–300, 2012.

[ZL08] S. Zhou and Q. Li. A variational level set method for the topology optimization of steady-state Navier-Stokes flow. *Journal of Computational Physics*, 227:10178–10195, 2008.

List of Figures

List of Tables

List of Algorithms