An efficient interface capturing method to simulate dense suspensions of bodies immersed in a fluid

Meriem Jedouaa

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel : 7 août 2006

Présentée par

Meriem JEDOUAA

Thèse dirigée par **Emmanuel MAITRE**
et codirigée par **Charles-Henri BRUNEAU**

préparée au sein **du Laboratoire Jean KUNTZMANN**
et de **L'École Doctorale *Mathématiques, Sciences et Technologies de l'Information, Informatique***

An efficient interface capturing method to simulate dense suspensions of bodies immersed in a fluid.

Thèse soutenue publiquement le **05 juillet 2017**,
devant le jury composé de :

M. Bertrand MAURY
Professeur, Université Paris-Sud, Rapporteur
M. Kai SCHNEIDER
Professeur, Université Aix-Marseille, Rapporteur
M. François BOUCHON
Maitre de conférences, Université Clermont-Auvergne, Examinateur
M. Georges-Henri COTTET
Professeur, Université Grenoble-Alpes, Examinateur
M. Charles-Henri BRUNEAU
Professeur, Université Bordeaux I, Co-Directeur de thèse
M. Emmanuel MAITRE
Professeur, Université Grenoble-Alpes, Directeur de thèse

Abstract

In this work, we propose a method to efficiently capture an arbitrary number of fluid/solid or fluid/fluid interfaces, in a level-set framework. This technique, borrowed from image analysis, is introduced in the context of the interaction of several bodies immersed in a fluid.
A configuration of the bodies in the fluid/structure domain is described by three label maps providing the first and second neighbours, and their associated distance functions. Only one level set function captures the union of all interfaces and is transported with the fluid velocity or with a global velocity field which takes into account the velocity of each structure. A multi-label fast marching method is then performed in a narrow-band around the interfaces allowing to update the label and distance functions.

Within this framework, the numerical treatment of contacts between the structures is achieved by a short-range repulsive force depending on the distance between the closest bodies.

The method is validated through the simulation of a dense suspension of rigid bodies immersed in an incompressible fluid. A global penalization model uses the label maps to follow the solid bodies altogether without a separate computation of each body velocity. Consequently, the method shows its efficiency when dealing with a large number of rigid bodies.
We also investigate the numerical simulation of vesicle suspensions for which a computation of elastic and bending forces on membranes is required. In the present model, only one elastic and bending force is computed for the whole set of membranes according to the level set function and the label maps.

Keywords: *fluid/structure interaction, level set method, multiple bodies, collision model*

Résumé

Dans ce travail, nous nous sommes intéressés à la simulation numérique de suspensions denses d'objets immergés dans un fluide. En s'inspirant d'une méthode de segmentation d'image, nous avons développé une méthode efficace de capture d'interface permettant d'une part de localiser les structures immergées et d'autre part de gérer les contacts numériques entre les structures.
Le domaine fluide/structure est représenté à l'aide de trois fonctions labels et deux fonctions distances qui permettent de localiser chaque structure et son plus proche voisin. Les interfaces sont capturées par une seule fonction level set, celle-ci est ensuite transportée par la vitesse du fluide ou par la vitesse de chaque structure. Un algorithme de multi-label fast marching permet de réinitialiser à chaque pas de temps les fonctions labels et distances dans un périmètre proche des interfaces. La gestion des contacts numériques est effectuée grâce à une force répulsive à courte portée prenant en compte l'interaction entre les objets les plus proches.
Dans un premier temps, la méthode est appliquée à l'évolution de solides rigides immergés. Un modèle de pénalisation global couplé aux fonctions labels permet de calculer en une seule fois l'ensemble des vitesses des structures rigides. Les résultats obtenus montrent l'efficacité de la méthode à gérer un grand nombre de solides.
Nous avons ensuite appliqué la méthode à des suspensions de vésicules immergées. Ce type de simulation requiert le calcul des forces élastiques et de courbures exercées sur les membranes. Grâce au modèle proposé, seulement une force élastique et une force de courbure sont calculées pour l'ensemble des membranes à l'aide de la fonction level set et des fonctions labels.

Mots clés: *interaction fluide/structure, méthode level set, suspensions d'objets, modèle de collision*

Notations

Physical and geometrical quantities

- d,$2 \leq d \leq 3$: the space dimension
- $\mathbb{I}_d$: the identity matrix
- Ω: the fluid/structure domain
- $\partial\Omega$: the boundary of Ω
- Ω_i: the region occupied by the ith structure
- Γ_i: the boundary of Ω_i
- U: Eulerian velocity field
- P: Pressure field
- ρ: density
- μ: dynamical viscosity
- $D(U)$: the deformation tensor
- σ, the Cauchy Stress tensor ($\sigma = \mu(\frac{\nabla U + \nabla U^t}{2}) - P\mathbb{I}_d$)
- g: gravity
- Re: the Reynolds number
- H_ε: smooth Heaviside function of thickness 2ε
- ζ_ε: smooth delta function of thickness 2ε

Numerical parameters

- Δx, Δy: the horizontal and vertical space step
- Δt: the time step
- ε: the half interface thickness
- λ: the penalization parameter

Contents

Introduction

Suspensions of particles in fluids can be found in many problems encountered in real life. Typical examples of such problems are the motion of living cells in blood vessels and the motion of fish swimming or dust particles. This type of flow exhibits very complex behavior, in the case of blood vessels living cells can deform and aggregate to form rouleaus, for fish swimming collective effects are observed (school of fish) even in the case of many rigid particles the behavior and the motion of particles is not fully understood. In fact, from chemistry, biology, physics to mathematics a large amount of research has been developed to understand these complex flows.

From a mathematical and numerical point of a view, the first challenge for the simulation of these complex flow stems from the coupling of the fluid and the structures. Indeed, the fluid is classically described in an Eulerian manner that is by the velocity of the fluid at all points of the domain and at all times whereas to describe the evolution of the structure the Lagrangian formulation is most suitable as it uses the evolution of each point of the structure with respect to a reference configuration.
Monolithic methods have been developed and consists in treating the fluid and the structure in the same mathematical framework. In Lagrangian methods, a set of Lagrangian markers or mesh cells are moving with the continuous medium. On the contrary, in Eulerian methods, a time independent mesh is used and represents the whole fluid/structure domain. Hybrid methods have also been employed and consists in combining both the Lagrangian description for the structure and the Eulerian one for the fluid.
Furthermore, the choice of the flow modeling has to be combined to the fluid/structure interface modeling which could be classified in two major class: the front tracking and front capturing methods. The front tracking approach consists in representing the interface by a set of Lagrangian particles which carry the physical properties of the structures, these markers are transported in a Lagrangian manner . In the front capturing methods, an Eulerian scalar function allows to reconstruct the interface. The interface modeling and flow modeling are then coupled to either form several coupled equations or a whole system.

Once these three criterias selected, a collision strategy is needed in order to avoid numerical contacts between the structures. Indeed, in the case of two perfectly smooth bodies evolving in a fluid, lubrication theory and experiments demonstrated that no contacts can occur in finite time between those particles [2; 77; 193]. However, during numerical calculations, the flow between two converging particles is not accurately resolved which leads to contacts or inter-penetration of particles. The numerical handling of these contacts and overlaps hence is crucial for the simulation of particles suspensions.
A natural approach proposed in [81] is to refine the mesh in the inter particle gap in order to resolve accurately the flow fields. However, this strategy is highly expensive as several refinements are necessary and in the case of multiple particles the cost of the simulation

is highly increased.
Other techniques consists in adding the theoretical lubrication forces when the gap between the bodies is very small [39; 121]. Due to the singular behavior of the forces, and the time discretization errors, this approach appears to be unsufficient and might still lead to contacts and overlaps at low spatial resolution.
Other numerical strategies less respectful of the underlying physics have been employed. These techniques consists in imposing a constraint on the particle motion by means for instance of artificial short range repulsive forces [31; 65] or by directly enforce a minimal distance between the particles maury1999direct. Contrary to the refinement strategy, these last collision methods allow, in addition of handling overlaps and contacts between particles, to use as coarse discretization as possible, reducing substantially the computational cost compared to the method proposed in [81].
An other major issue to be taken into consideration is the computational complexity due to an increasing number of bodies. For instance, both short-range repulsive forces proposed in [31; 65] are quadratically dependent on the number of particles if no special trick is used. Moreover, depending on the numerical method used for the Fluid/Structure interaction coupling the computational can be reduced. A typical example is the Boundary integral method ([178; 179]), which by only discretizing the membrane boundaries allows to obtain an efficient method to deal with several bodies.

In the present work, a fully eulerian formulation of the fluid-structure interaction is adopted and combined to a level set method which is part of the front capturing methods. Within this framework, we investigate the two major difficulties we stated above: the high computational complexity due to a large number of particles and contacts between particles during numerical calculation.
We adapt in the context of Fluid-Structure interaction a model introduced by J.Bogovic ([11]) for image segmentation. In the context of fluid-structure interaction, this model provides access to the location of each structure and its closest neighbors at each point of the Fluid/Structure domain, it is thus a very desirable model for dealing with dense suspensions of particles. Indeed, by evolving only five field functions, the proposed approach can (1) locate and evolve each structure in the domain (2) specify a speed or a force independently for each structure (3) and handle numerical contacts between the structures.
In order to validate the ability of this method to handle numerical contacts and its efficiency to deal with a large number of structures, two applications are explored: rigid and deformable bodies. The deformable bodies considered are vesicles which are simplified model to study the behavior of red blood cells.

This work is organized as follows:

In Chapter 1, we provide an overview of popular numerical methods for fluid/structure interaction as well as interface modeling techniques, in particular we focus on the level set method. Then, the physical models considered are presented as well as the numerical resolution of the fluid model combined to a level set method.

In Chapter 2, the capturing method proposed in [11] is presented in the context of multiple bodies immersed in a fluid. The three label maps and two distance functions which allows to locate the structures and their first neighbors are presented.

Chapter 3 is devoted to the collision strategies, we present a brief overview of existing methods to deal with numerical contacts. Then, a short-range repulsive force inspired by [31] is proposed and depends on the relative velocities between the closest structures. The influence of these two collision models on the dynamics of disks transported by steady stokes fluid flows is compared.

In Chapter 4, we present the numerical algorithm used to evolve the structures in the domain and to redefine the neighbors of the particles. This algorithm consists in evolving a level set function which captures the union of all interfaces, in a neighborhood of the interfaces, then a multi-label fast marching procedure is performed in these thin narrow-bands to update the five field functions.

Chapter 5 proposes an application of the method to the simulation of rigid bodies evolving in an incompressible fluid, we present a penalization model which is made independent on the number of rigid bodies and a benchmark of numerical simulations of the sedimentation of rigid bodies is proposed.

Finally, in Chapter 6 the numerical simulation of immersed vesicles is carried out. The fluid/elastic membrane coupling is achieved by using the model introduced in [32], only one elastic and curvature force is computed for the whole set of vesicles. In order to compute correctly the elastic and curvature forces a finer grid for the level set function is used and thanks to the label maps the forces exerted on the membrane are accurately computed even if the elastic membranes are close.

Chapter 1

Fluid/Structure Interaction and Level set methods

Contents

The focus of this chapter is to present the context of this work. The numerical simulation of fluid/structure interaction (FSI) involves to deal with three different problems: the governing equations of each media (fluid or structure) and the coupling between the structure and the fluid which itself depends on the properties of the structure. The numerical techniques devoted to the simulation of FSI can hence be classified according to different criterias: the Eulerian or Lagrangian formulation for the fluid and/or the structure, the interface modeling which can be achieved by tracking or following the interface and the coupling between the structure and the fluid.

This chaper is organized as follows:
In a first section, an overview of numerical methods for fluid/structure interaction is presented. Then, the physical models we are interested in are described. The third section is dedicated to the existing numerical methods to capture and evolve interfaces between these medias, in particular we focus on the level set method.
Finally, the discretization of the fluid/structure domain Ω and the numerical resolution of the fluid equations is presented as well as the evolution of the interfaces using the level set method.

1.1 Numerical methods for Fluid/Structure Interaction

The interaction between viscous fluid flow and structure is of great significance in many natural processes such as in biomechanics (cell deformation, blood-heart valves interaction, sperm motility), aerodynamics, sedimentation of particles. In fact, FSI problems play a key role in a large panel of fields, and there is a great deal of science works devoted to these problems. Accordingly, increasingly sophisticated numerical techniques have been developed over the past few years, making of numerical simulation an essential

tool to understand FSI problems.

The main difficulty in simulating this problem comes from the divergence in the modelling approach for each medium : the Eulerian and Lagrangian description.
In the Eulerian approach, the physical variables are defined at all points of the domain and at all times. From a numerical point a view, the continuum medium is discretized with a mesh grid fixed in time. A Cartesian mesh grid can be used to represent the medium and allows for instance to use finite difference method.
With the Lagrangian description, the physical variables are defined at points of the media which moves over time. Numerically, the continuum media is discretized by a time-varying mesh grid. Particles located at the mesh grid points are then displaced with their associated velocity. Denoting by $X(a,t)$ the position of a particle a at time t, this particle will be transported as:

$$\frac{\partial X}{\partial t}(a,t) = u(X(a,t),t)$$

This type of description allows to better track the motion and deformation of a structure. A major inconvenient of this approach is that after deformation the Lagrangian particles accumulate or disperse in the domain. Consequently, additional procedures are required in order to follow large distortions of the computational domain.

Generally, the fluid flow is described in an Eulerian manner whereas the evolution of structures are more naturally handled in the Lagrangian formulation. To overcome this difficulty, the existing methods have adopted three different points of view: purely Lagrangian, Eulerian/Lagrangian and purely Eulerian.

As its name suggests, the first one consists in describing both the structure and the fluid with a Lagrangian approach.

Two of the most widely used particle methods are the Smooth Particle Hydrodynamics method and the Vortex Method.
The Smooth Particle Hydrodynamics method has been originally designed to study gas dynamics for astrophysics [64; 113]. This particle method does not require a mesh grid to discretize the domain, only the Lagrangian particles are handled. The variables of the fluid and structure are updated simultaneously allowing to take into account easily large displacement of the fluid/structure interfaces. This technique is accurate and stable for the simulation of compressible flows and to capture interfaces. An extension for free-surface incompressible fluid flow has been proposed by Monaghan in [131]. To impose the incompressibility constraint, the pressure gradient is evaluated at each Lagrangian markers. A force which takes into account the local variation of the density is carried by each marker, the incompressibility constraint consists then to impose a constant value to this density. With this technique, only an approximation of the incompressibility constraint is imposed. Improved SPH methods [48; 82] have hence been developed in order to overcome the main weakness of this method which lies in its inaccuracy to achieve volume conservation. This technique has been widely used like fluid/fluid interactions ([28]), fluid/structure interaction ([5]), to simulate interaction between different solids ([182]) and many others.

In the case of incompressible fluid flow, it is most convenient to use the Vortex Method. Indeed, contrary to the SPH methods, the incompressibility constraint is explicitly en-

forced as thanks to the vortex formulation the velocity field computed is implicitly divergence free. The Vortex Methods uses a velocity-vorticity formulation of the Navier-Stokes equations, compared to SPH methods the set of Lagrangian particles are only located at the regions where the vorticity is non zero providing efficiency in the case where these regions represent small portions of the whole computational domain. The main drawback of these methods comes from the treatment of the boundary conditions as a reformulation of the velocity boundary conditions are required which can in some cases represent a source of difficulties. As all the Lagrangian techniques, these two particle methods suffers from particle distortion problem. To circumvent this problem problem, a widely used technique is the remeshing procedure originally designed in [84; 104]. These methods consist in redistributing the Lagrangian markers in the domain at each iteration by means of high order interpolation kernel allowing to control efficiently the distance between the particles. This technique have been widely and successfully used in vortex methods [24; 34; 103].

An other approach is the boundary integral method which has been proposed in [146] for the simulation of an immersed vesicle and then improved and extensively used in several works, for instance in [178] where the numerical simulation of a suspension of vesicles has been investigated. Thanks to the integral equation formulation, this method is efficient to deal with interfacial flow and vesicles as only the interface is discretized. However, this approach is limited to Stokes flow and simple structure geometries.

A popular and wide spread fluid/structure method is the Arbitrary Lagrangian Eulerian approach (ALE) introduced by Donea in 1982 (see [44]) and extensively studied by several teams [54; 73; 80; 149; 187]. The ALE strategy is an hybrid method that combines the Lagrangian and Eulerian descriptions using a mobile non structured grid that follows the normal displacement of the fluid/structure interface. The fluid and solid equations are solved individually and continuity conditions for the velocity and stress tensor are explicitly discretized at the interface.
The main weakness of the ALE method is its difficulty of implementation, especially when dealing with large displacements in dimension three. In addition the added mass effect [29] has been a long standing difficulty which has been worked around only recently [51–53; 55; 56]. Moreover, the computational grid has to be remeshed when the elements get too distorted, which could be a very costly procedure in three-dimension.
An other popular Eulerian/Lagrangian method is the immersed boundary approach proposed in [141; 142] by Peskin for the simulation of an immersed elastic membrane. In this approach, an Eulerian grid represents the fluid and the membrane is discretized by a set of Lagrangian particles. The fluid/structure coupling is then achieved by projecting on the Eulerian grid points the forces carried by the Lagrangian particles. Since its inception, this technique has found a large number of applications, it can deal with elastic membranes as well as moving rigid bodies.

The fully Eulerian method is the point of view that we adopt in this work. The idea is to use a purely Eulerian description of the fluid/structure problem. A popular set of purely Eulerian methods is the fictitious boundary methods. Initially, in this set of techniques, the fluid is defined in the all computational domain Ω including the regions occupied by the particles. To take into account the immersed structures, a constraint is applied to the fluid inside and at the interface of the structures. A global model is defined in the fluid and structures regions allowing to use a fix grid mesh in the entire

fluid/structure domain (see Figure 1.1). Those techniques are generally easy to implement, allow to use fast solvers and the extension to several immersed structures is often straightforward.
Numerous methods based on this conceptual framework have been developed, for instance in [41], a mixing of the "Fictitious concept" and the ALE technique has been achieved, adding to these methods the high accuracy of the ALE approach.

We present in what follows some popular existing methods for FSI that can be applied to our problems. The first one is the well-known Immersed Boundary methods, then some Fictitious domain methods are described.

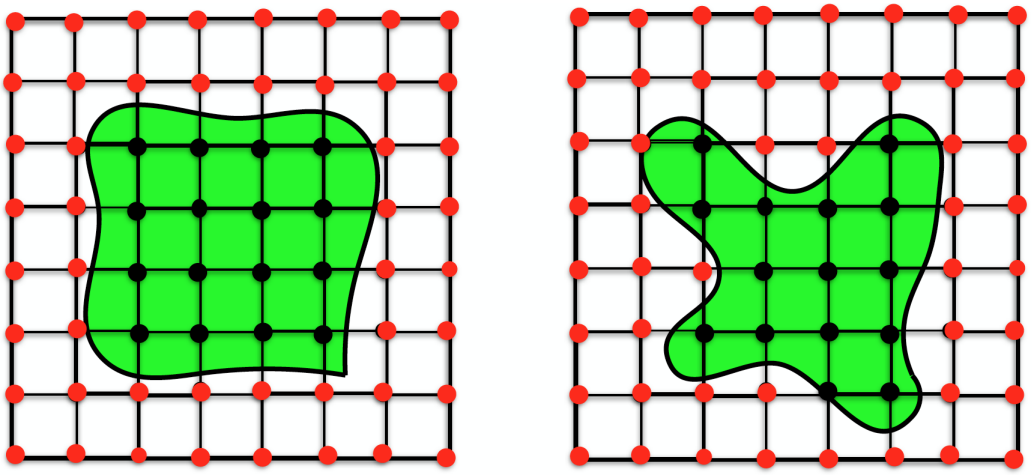


Figure 1.1: Sketch of FSI problem discretized with an uniform Eulerian mesh grid at different time. The red points represents the fluid.

1.1.1 The immersed boundary method

The immersed boundary technique has been firstly introduced by Peskin in [141–143] to simulate immersed elastic membrane in the vicinity of heart valves. A fix uniform Cartesian grid is used to represent the fluid region and the fluid/structure interface is described by a set of Lagrangian particles. The elastic forces are computed on the Lagrangian markers and corresponds to a sum of discrete delta functions localized on each marker. Then, a numerical technique is performed in order to spread these delta functions on the Eulerian grid points. Peskin proposed to construct a discrete delta function regularized at the interface making the forces spread on few Eulerian discretization points around the interface. This delta function has to fulfil different properties: continuity at the interface to avoid jump of the velocity and forces, allows to conserve the moment of the forces across the interface, and sufficiently narrow support. Finally, the forces computed on the Eulerian grid are then inserted in the right hand side of the governing equations of the fluid flow. Once the fluid velocity field has been computed, the Lagrangian particles are advected in order to evolve the interface.

Let $\Omega \in \mathbb{R}^2$, and Γ_1 be a closed curve parametrized overtime by $X(s,t)$ where $s \in [0, L]$. We consider that the fluid is governed by the Navier-Stokes equations.
The forces projected on the Eulerian mesh grid are given by:

$$f(x,t) = \int_0^L F(s,t)\delta(x - X(s,t))ds$$

Then, the interface is evolved as:

$$\frac{\partial X(s,t)}{\partial t} = u(X(s,t),t) = \int_\Omega u(x,t)\delta(x - X(s,t))ds$$

Finally, these forces are added to the Navier-Stokes equations giving the following fluid/structure model.

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U) - \mu\Delta U + \nabla p = \rho g + f & \text{in } \Omega \\ f(x,t) = \int_0^L F(s,t)\delta(x - X(s,t))ds & \text{in } \Omega \\ \dfrac{\partial X(s,t)}{\partial t} = u(X(s,t),t) = \int_\Omega u(x,t)\delta(x - X(s,t))ds & \text{in } \Omega \end{cases} \tag{1.1}$$

The delta function locates the forces and provides the interface velocity in function of the velocity of the fluid. The force F depends on the type of the considered structure. In the case of an elastic membrane, this force will determine the elastic behaviour of the structure. An Eulerian formulation has been proposed in [32; 33], in the level set framework, with application to immersed elastic membranes.

A first adaptation of this technique for moving particles has been performed in [57] and then for rigid boundaries in [70; 71; 152]. Then, several improvements of this method have been introduced to deal with moving rigid bodies for instance in [18; 175; 176]. In fact, this technique gave born to a large number of methods, we can cite for instance the immersed interface method (see [109; 111]), the extended immersed boundary method.

1.1.2 Fictitious domain methods

The original concept of the fictitious domain methods, have been firstly proposed by Hyman in [85] and then developed by Saul'ev in [154] who gave the name "Fictitious domain". In the context of FSI, the main idea of this set of methods is to extend the fluid domain in a larger and geometrically simpler domain called the "Fictitious" domain which contains the immersed structure. The advantages of these methods is that a time-independent mesh can be used for the entire fluid/structure domain allowing to use fast solution methods. We present in the following some popular approaches stemming from this conceptual framework.

The Penalization method

The Penalization technique has firstly been introduced by [20] and further developed in [4] for the study of a fixed rigid body immersed in a incompressible fluid.
The principle of this technique is to define the fluid in the whole computational domain including the rigid bodies and then to penalize the fluid velocity inside the rigid body. This is achieved by adding a penalization term in the Navier-Stokes equations, allowing to impose a zero motion inside the solid and to solve the boundary value problem inside the whole domain Ω including the bodies. So, there is no need for an adapted mesh to

the geometry of the bodies and a Cartesian mesh on a box domain can be used.

Let Ω_1 denotes a fixed rigid body immersed in a fluid Ω_{N_f}, the L^2 penalization consists in adding a penalization term in the Navier-Stokes equations as follows:

$$\rho(\partial_t U + (U \cdot \nabla)U) - \mu \Delta U + \nabla p = \rho g + f \tag{1.2}$$

where the penalization term is given by:

$$f = \frac{U}{K}$$

and the density function is provided by:

$$\rho = \begin{cases} \rho_1 & \text{in } \Omega_1 \\ \rho_f & \text{in } \Omega_{N_f} \end{cases}$$

The parameter K represents the permeability of the medium. In the fluid region, the permeability coefficient K tends to infinity giving the Navier Stokes equations. In the solid body, the coefficient K tends to zero and we solve a Darcy Law with a velocity proportional to K. The solid body can hence be viewed as a porous media with a very small permeability.
Numerically, the permeability coefficient can either be defined on each grid point of the computational domain with a very high value in the fluid and a very small one in the rigid body or by using a characteristic function χ of the solid body, we can set $\frac{1}{K} = \lambda\chi$ where $\lambda >> 1$ is a fixed penalization coefficient. Inside the fluid the penalization term is exactly zero and in the solid body $f = \lambda U$; numerically a smooth version of the characteristic function is required in order to regularize the penalization term at the interface.

The numerical analysis of the penalization method is presented in [126], it is shown that the L^2 penalization method is of order one and $\frac{1}{2}$ with the H^1 penalization this is due to the discontinuity of the normal derivatives of the velocity field at the interface. An extension to a prescribed velocity has been proposed in [140], for a solid body Ω_1 moving with a velocity U_1, the L^2 penalization problem to solve is:

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U) - \nabla \cdot (\mu \nabla U) + \nabla p = \rho g + \lambda\chi(U_1 - U) & \text{in } \Omega_{N_f} \times (0,T) \\ \nabla \cdot U = 0 & \text{in } \Omega_{N_f} \times (0,T) \\ U_1 = U_1^t + w_1 \times (x - x_1^g) & \text{on } \Gamma_1, \\ m_1 \dfrac{du_1^t}{dt} = F_1 \\ J_1 \dfrac{dw_1}{dt} = T_1 \end{cases} \tag{1.3}$$

where J_1 is the inertial matrix, x_1^g the gravity center and m_1 the mass, u_1^t and w_1 are the translational and angular velocities of Ω_1. The total force and torque F_1 and T_1 takes into account the hydrodynamical forces and the external forces F_{ext} exerted on the particle Ω_1:

$$\begin{cases} F_1 = F_{ext} + \int_{\Gamma_1} \sigma n_1 & \text{on } \Gamma_1 \\ T_1 = T_{ext} + \int_{\Gamma_1} (x - x_1^g) \times \sigma n_1 & \text{on } \Gamma_1 \end{cases} \tag{1.4}$$

where σ denotes the Cauchy Stress tensor, n_1 is the normal to Γ_1 pointing outside the rigid body Ω_1.
The rigid velocity has to be computed before the resolution of the penalized Navier-Stokes equation. This can be achieved by averaging the global velocity field inside the rigid body providing:

$$U_1 = \frac{1}{|\Omega_1|}\int_{\Omega_1} \rho\chi U dx + \left(J_1^{-1}\int_{\Omega_1} \rho\chi U \times (z - x_1^g)dz\right) \times (x - x_1^g) \tag{1.5}$$

Fat boundary method

The fat boundary method has first been introduced in [123] to solve a Poisson problem on a perforated domain and further developed in [86] for moving holes.
The main idea of this technique is to replace the initial problem defined in the entire perforated domain by two sub-problems: a global model and a local one. The two sub problems are resolved on two different meshes: a fixed in time global mesh grid defined on the all perforated domain and a local mesh defined around the holes. The local mesh is time-varying and finer than the global mesh grid. A Cartesian mesh grid can be used to discretize the global model allowing to use fast solvers and efficient preconditioners.
Then, the two problems written on the different meshes are coupled, by using an iterative domain decomposition method and a jump of the normal derivative at the interface is prescribed. The final solution is computed so that the solution on both meshes match and a continuity on the normal derivative at the interface is imposed. A mathematical analysis of this method is presented in [6]. Better accuracy is obtained compared to the penalization method H^1 as the continuity of the normal derivatives is imposed. This model has been applied to moving structures immersed in fluid.

Let us take the example of a perforated domain Ω like shown in Figure 1.2, where the green zone represented the safety zone (S) in which the local model is computed.

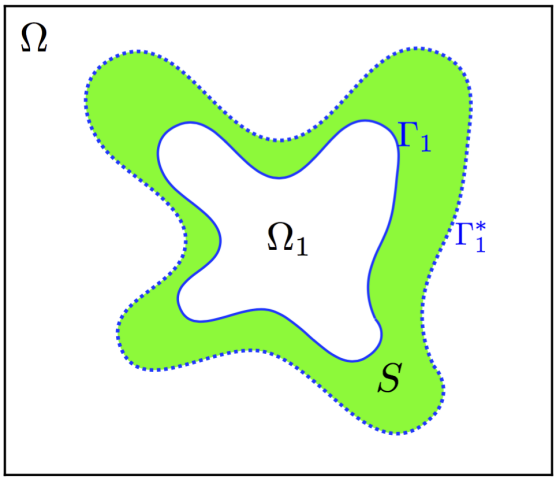


Figure 1.2: 2D example of a perforated domain Ω. The hole is represented by Ω_1, its boundary Γ_1 and the safety zone S.

Using the Fat boundary method, the Poisson problem :

$$\begin{cases} -\Delta U = F & \text{in } \Omega \setminus \Omega_1 \\ U = 0 & \text{on } \Gamma = \partial\Omega \\ U = 0 & \text{on } \Gamma_1 \end{cases} \tag{1.6}$$

is decoupled into the two following problems:

$$\begin{cases} \text{Local Problem:} & \begin{cases} -\Delta V = F & \text{in } S \\ V = U* & \text{on } \Gamma_1^* \end{cases} \\ \\ \text{Global Problem:} & \begin{cases} -\Delta U^* = F^* + \dfrac{\partial V}{\partial_n}\delta_{\Gamma_1} & \text{in } \Omega \\ U^* = 0 & \text{on } \Gamma \end{cases} \end{cases} \tag{1.7}$$

where Γ_1 and Γ_1^* are the boundaries of the safety zone, F^* is the extension of F by 0 in Ω_1. The boundary condition on the structure boundary Γ_1 is extended inside the structure by the term $\dfrac{\partial V}{\partial_n}\delta_{\Gamma_1}$. An illustration of the global and local meshes for this problem is shown on Figure 1.3 in very coarse configuration.

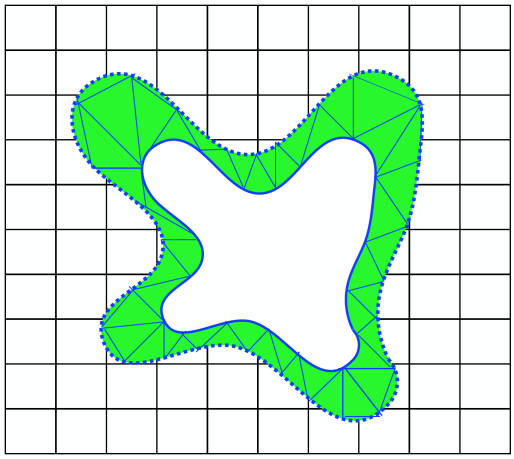


Figure 1.3: Illustration of the global and local meshes for FBM.

Lagrange multiplier method

This approach has been firstly proposed in [65; 67] in order to model the motion of a rigid body immersed in a fluid. The principle of this method is to introduce lagrange multipliers to impose the rigidity constraint inside the rigid body and the Dirichlet boundary condition at the interface of the rigid body.
In [65], the constraint used is the following:

$$U = U_1^t + w_1 \times (x - x_1^g) \quad \text{in } \Omega_1 \tag{1.8}$$

An improvement of this approach in term of implementation and computational saving is proposed in [158] and uses the rigidity constraint:

$$\begin{cases} \nabla \cdot (D(u)) = 0 & \text{in } \Omega_s \\ D(U) \cdot n_s = 0 & \text{on } \Gamma_s \end{cases} \tag{1.9}$$

which is equivalent to:

$$D(U) = 0 \text{ in } \Omega_s \tag{1.10}$$

Then, the Lagrangian multipliers are solutions of the following equations:

$$\begin{cases} \nabla \cdot (D(u)) = \nabla \cdot (D(\lambda)) = 0 & \text{in } \Omega_s(t) \\ D(u) \cdot n = D(\lambda) \cdot n = 0 & \text{on } \Gamma_s(t) \end{cases} \tag{1.11}$$

The Navier Stokes equations:

$$\rho(\partial_t U + (U \cdot \nabla)U) - \nabla \cdot (\mu \nabla U) + \nabla p = \rho g + f$$

are then supplemented with the forcing term $f = \nabla \cdot (D(\lambda))$, the incompressibility constraint and the boundary condition on the structure boundary.
In order to tackle this equation, a time splitting method is proposed in [158] like the projection method of Chorin type.
The first step is to compute the Navier Stokes equations without the forcing term f giving an intermediate velocity u^* which is divergence free.
The second step is to impose the rigidity constraint, by computing the Lagrangian multipliers. The obtained velocity is divergence free in the all domain and the rigidity constraint is ensured inside and at the interface of the rigid body.
Two variants of the Lagrange multiplier method exists: Boundary and Distributed approachs.
With the Boundary approach, the Lagrangian multipliers are only computed on the surface of the structure, thus a mesh defined over the surface is sufficient (see for instance [68]).
With the second approach, the Lagrange multipliers are distributed all over the immersed structure, the interior and the interface has to be meshed. The choice over these two variants lies on the type of applications one wants to simulate, we represent on Figure 1.4 an illustration of these techniques.

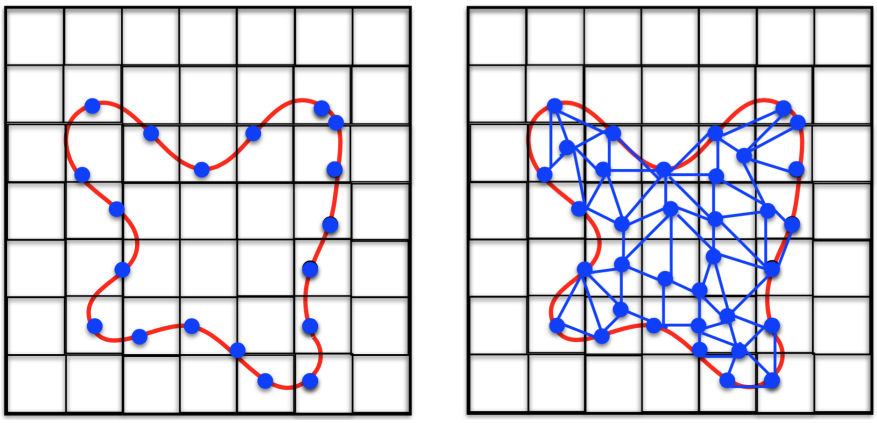


Figure 1.4: Illustration of the Boundary (left) and Distributed (right) Lagrange multiplier methods. The Lagrange multipliers are applied at the blue nodes.

1.1.3 Cut-cell methods

The methods presented above do not depend on the choice of the discretization schemes used to solve the equations. On the contrary, in the "discrete forcing methods", the forcing depends on the discretization and thus on the numerical schemes used. In particular, this approach allows to control directly the numerical precision, stability and conservation of the solvers. The most popular one is the Cut-Cell method.

This method also called Cartesian grid method has been firstly proposed in [27] for inviscid flow computations and then applied to the simulation of viscous flow in [173; 174; 190]. This technique has been designed to satisfy the conservation laws in the grid cells located in a vicinity of the immersed boundary. In order to conserve locally and globally the mass and the momentum, the cut-cell method is achieved in a Finite Volume framework.
Then, the main idea is to reshape the grid cells crossed by the immersed boundary as follows: if the center cell is located in the fluid part, the cell is reshaped by discarding the portion that lies in the solid region. On the contrary, portions of grid cells whose centers are part of the solid region, are absorbed by the surrounding cells. The obtained cut-cells define the new control volumes in the vicinity of the structure boundary for the finite-volumes discretization scheme.
In the case of staggered grid like MAC grid, the reconstruction of the cut-cells and the position of the unknowns (the velocity field and the pressure field) has to be achieved in accordance with the five point structure of a cell. If a structure is immersed in the Cartesian grid, the five point structure of the cells is usually not preserved, even more if one uses the merging technique.

In [25], Cheny and Bottela proposed a new cut cell method based on MAC schemes which keeps the five point structure of the cells and satisfy the conservation laws. The structure interface is accurately represented by a signed algebraic distance to the immersed boundary. The staggered arrangement of both the velocity field and pressure is adapted to the cut-cells geometry. From a computational point of view, this method is highly efficient as it is based on MAC solvers on Cartesian grids. This method has been used by Bouchon et al in [13] who proposed a global second order MAC scheme, the first order is obtained in the vicinity of the immersed boundary.

1.2 Physical models

In this work we study the problem of N structures evolving in an incompressible fluid. We consider a rectangular domain $\Omega \in \mathbb{R}^d, d = 2, 3$ filled with structures and a fluid as illustrated in Figure 1.5. We denote by $(\Omega_i)_{i=1,..,N}$ the N structures and the surrounding fluid is denoted by Ω_{N_f} where $N_f = N+1$. With these notations, the entire fluid/structures domain Ω is partitioned into $N+1$ objects as:

$$\left\{ \begin{array}{rcl} \forall i \neq j, \Omega_i \bigcap \Omega_j & = & \emptyset \\ \Omega_{N_f} & = & \Omega \backslash \overline{\{\bigcup_{i=1}^{N} \Omega_i\}} \\ \Gamma_{N_f} & = & \bigcup_{i=1}^{N} \Gamma_i. \end{array} \right. \tag{1.12}$$

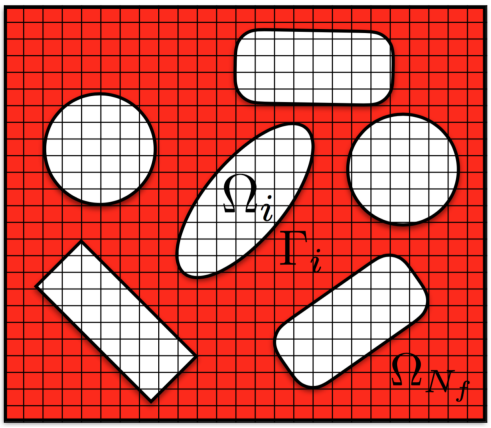


Figure 1.5: 2D example of a fluid/structure domain Ω represented by an Eulerian grid.

1.2.1 Fluid model

The fluid Ω_{N_f} is a Newtonian viscous incompressible fluid. It is governed by the incompressible Navier Stokes equations. Let $T > 0$, $t \in [0, T]$, and $x \in \Omega_{N_f}(t)$, we denote by:

- $U(x,t)$: fluid velocity field
- $P(x,t)$: pressure field
- $\mu_f(x,t)$: dynamic viscosity of the fluid
- $\rho_f(x,t)$: fluid density

The incompressibility constraint is given by:

$$\nabla \cdot U = 0 \text{ in } \Omega_{N_f}(t)$$

This constraint gives the following mass conservation equation:

$$\partial_t \rho_f + U \cdot \nabla \rho_f = 0 \text{ in } \Omega_{N_f}(t)$$

Then, the fundamental principles of dynamics gives the following equation :

$$\rho_f(\partial_t U + (U \cdot \nabla)U) = \nabla \cdot \sigma \text{ in } \Omega_{N_f}(t)$$

Where σ is the Cauchy stress tensor and $D(U)$ is the deformation tensor. Finally, the incompressible Navier Stokes equations are given by:

$$\begin{cases} \rho_f(\partial_t U + (U \cdot \nabla)U) - \mu_f \Delta U + \nabla P = \rho_f f & \text{in } \Omega_{N_f} \times (0,T) \\ \nabla \cdot U = 0 & \text{in } \Omega_{N_f} \times (0,T) \end{cases} \tag{1.13}$$

where f represents the external forces exerted on the fluid. These equations have to be supplemented with initial conditions and boundary conditions on Γ the boundary of Ω. After non-dimensionnalization, one can obtain the dimensionless Navier-Stokes equations :

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U)) - \dfrac{1}{Re}\Delta U + \nabla P = 0 & \text{in } \Omega \times (0,T) \\ \nabla \cdot U = 0 & \text{in } \Omega \times (0,T) \end{cases} \tag{1.14}$$

where the Reynolds number is given by:

$$Re = \frac{L_r U_r}{\mu_r}$$

L_r and U_r are the characteristic lenght and velocity of the model, μ_r denotes the characteristic viscosity of the considered fluid. The Reynolds number is a physical parameter which provides information on the type of flow. Depending on its values, the flow can be classified in different categories. Mainly, if $Re < 2000$ the fluid is laminar meaning that the viscous phenomenons are more important than the inertial effects. When the Reynolds number is higher than 4000, the inertial effects are the most important, this is called the turbulent regime.
In the case of small Reynolds number, one can neglect the convective term $(U \cdot \nabla U)$, the obtained system is the Stokes equations. In particular, the Stokes model has been widely used for modeling blood flow at places where the Reynolds number is of order 10^{-4} (small vessels or capillaries).

1.2.2 Immersed rigid bodies

The motion of a rigid body is obtained by means of the principle of conservation of linear and angular momentum. For a body Ω_i immersed in a fluid, the equation which governs the translation is:

$$m_i \frac{dV_i}{dt} = F_i$$

In the above equation, m_i denotes the mass of Ω_i, V_i the translational velocity, t is the time and F_i is the sum of all the forces exerted on the particle Ω_i. The total force F_i takes into account the hydrodynamical forces and the external forces F_{ext} (for instance the gravity force) :

$$F_i = F_{ext} + \int_{\Gamma_i} \sigma n_i$$

where n_i is the normal to Γ_i pointing outside the particle Ω_i and σ is the Cauchy Stress tensor for a newtonian fluid. The conservation principle of angular momentum can be stated as:

$$J_i \frac{dw_i}{dt} = T_i$$

where J_i is the inertial matrix, x_i^g its gravity center, w_i denotes the rotational velocity of the rigid body Ω_i.The total torque T_i due to hydrodynamical forces and external forces is:

$$T_i = T_{ext} + \int_{\Gamma_i} (x - x_i^g) \times \sigma$$

Then, the interaction between the structures and the fluid is modelled by the Navier Stokes equations for the fluid, the rigid motions in the fluid for the solid bodies and the no slip boundary conditions on each rigid body interface. The no-slip boundary condition means that the fluid adheres to the solid boundaries, this condition reflects the fact the considered bodies are rigid and thus do not deform.
Finally, the complete model can be expressed as follows:

$$\rho_f(\partial_t U + (U \cdot \nabla)U) - \nabla \cdot (\mu \nabla U) + \nabla p = 0 \ \text{ in } \Omega_{N_f}(t) \tag{1.15}$$

$$\nabla \cdot U = 0 \ \text{ in } \Omega_{N_f}(t) \tag{1.16}$$

$$U = V_i + w_i(x - x_i^g)^{\perp} \text{ on } \Gamma_i(t), \tag{1.17}$$

$$m_i \frac{dV_i}{dt} = F_i \tag{1.18}$$

$$J_i \frac{dw_i}{dt} = T_i \tag{1.19}$$

These equations have to be supplemented by boundary conditions for the fluid velocity and initial conditions for the studied variables.

1.2.3 Immersed elastic membranes

We give here a general description of N immersed elastic membranes without precise specification of the elastic behaviour of the membrane. Each structure Ω_i is a closed curve in dimension two or surface in dimension three which encloses an incompressible fluid of dynamic viscosity μ_i and density ρ_i. The elastic forces are located on the membrane and we consider impermeable membrane so that the jump of the velocity field at the interface is zero:

$$\forall i \in \{1, .., N\}, \ \ [U] = 0 \ \text{ on } \ \Gamma_i,$$

We first present the incompressible Navier-Stokes for multi-fluid flows and an approximation of these equations for small variation of the density before introducing the immersed boundary model of immersed elastic membrane.

Navier Stokes equations with variable densities and viscosities

We describe the incompressible Navier-Stokes for variables densities and viscosities, we consider N fluids immersed in a fluid contained in the region objects Ω_{N_f}.
Each interface Γ_i is a closed curve in dimension two or surface in dimension three which encloses an incompressible fluid of dynamic viscosity μ_i and density ρ_i. Defining the density and viscosity function as:

$$\rho = \begin{cases} \rho_i & \text{in } \ \Omega_i, \ \ \forall i \\ \rho_f & \text{in } \ \Omega_{N_f} \end{cases} \qquad \mu = \begin{cases} \mu_i & \text{in } \ \Omega_i, \ \ \forall i \\ \mu_f & \text{in } \ \Omega_{N_f} \end{cases} \tag{1.20}$$

The Navier Stokes equations to model multi-fluid flows are given by:

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U) - \nabla \cdot (\mu D(U)) + \nabla P = \rho g & \text{in } \Omega \times (0, T) \\ \nabla \cdot U = 0 & \text{in } \Omega \times (0, T) \end{cases} \tag{1.21}$$

where g is the gravity force, the surface tension between the two fluids has to be added to these equations.

Boussinesq Approximations for the Navier-Stokes equations

In [14], Boussinesq proposed an approximation of the multi-fluid flows Navier Stokes equations. In the case of two fluids with corresponding densities ρ_1 and ρ_f (where ρ_f is

the density of the lighter fluid), this approximation can be made if the variation of the density is small that is if the Atwood number $A_t = (\rho_1 - \rho_f)/(\rho_1 + \rho_f)$ is negligible, and if the densities ρ_1 and ρ_f are uniform on each side of the interface. By neglecting the density variation in the inertial terms this approximation consider that the leading force (compared to other forces between two fluids) is the one due to gravity.

Considering that :

$$\forall i \in \{1, .., N\}, \forall j \in \{1, .., N\}, \rho_i = \rho_j$$

and denoting by $\delta\rho = \rho_f - \rho_i$ the difference density between the surrounding fluid and each fluid contains in the subdomain Ω_i, if we suppose that:

$$\frac{\delta\rho}{(\rho_i + \rho_f)} << 1$$

the Boussinesq approximation provides the following approximation of the system (1.21):

$$\begin{cases} \rho_m(\partial_t U + (U \cdot \nabla)U) - \nabla \cdot (\mu D(U)) + \nabla P = \rho g & \text{in } \Omega \times (0, T) \\ \nabla \cdot U = 0 & \text{in } \Omega \times (0, T) \end{cases} \tag{1.22}$$

where $\rho_m = \frac{\rho_i + \rho_f}{2}$ is the mean density and ρ is given by 1.20.

The advantage of this approximation is that the obtained system is easy to implement and as the coefficient in the Poisson equation is constant, fast elliptic solvers can be used. This approach has been used in many previous works (see for instance [74; 87; 171]).

Immersed boundary model

In addition to being submitted to hydrodynamical forces the elastic membrane has its own mechanical properties modelled by elasticity laws which can depend on the stretching, the curvature of the surface etc.

We present here the immersed boundary model for an immersed elastic membrane in the two dimensional case.

Consider N elastic closed curves Γ_i each enclosing an incompressible fluid domain Ω_i with corresponding viscosity μ_i and density ρ_i surrounded by the fluid Ω_{N_f} of density ρ_f and dynamic viscosity μ_f. The linear densities of the curves are assumed to be zero.

For each Γ_i, we denote by Γ_i^0 the initial position of the curve and $s \to \gamma_0^i(s)$ a parametrization of Γ_i^0. Let Γ_i^t denotes the position of Γ_i at time t, and $\gamma_i : [0, M] \times [0, T] \to \gamma(s, t)$ denotes the regular parametrization of the curve Γ_t, denoting by X the flow associated to the vector field U, we have:

$$\gamma_i(s, t) = X(t, \gamma_0^i(s), 0)$$

Thus, the parametrization γ is solution of the following system::

$$\begin{cases} \partial_t \gamma_i(s, t) = U(\gamma_i(s, t), t), & s \in [0, M],\ t \in]0, T] \\ \gamma_i(s, 0) = \gamma_0^i(s), & s \in [0, M] \end{cases} \tag{1.23}$$

where U is the velocity field of the continuous medium.

We denote by Γ_i^r, the configuration of Γ_i when it is at rest and by $\theta_i : [0, M] \to R^2, M > 0$

a regular parametrization of Γ_i^r.
For instance, we take for each curve the following elastic behaviour law :

$$G_i(s,t) = E'\left(\frac{|\gamma_i(s,t)|}{|\theta_i(s)|}\right)\frac{\gamma_i(s,t)}{|\gamma_i(s,t)|} \tag{1.24}$$

where E' is a function which satisfy $E'(r) = 0$ for $r \leq 1$. Defining the density and viscosity functions as in (1.20) the immersed boundary model for the N immersed elastic membranes is:

$$\begin{cases} \rho(\partial_t U + U\cdot\nabla U) - \nabla\cdot(\mu D(U)) + \nabla P = \sum_{i=1}^{i=N} \frac{1}{|\gamma_i|}\frac{\partial}{\partial_s}(G_i)\,\delta_{\Gamma_i} + \rho f & \Omega\times(0,T) \\ \nabla\cdot U = 0 & \Omega\times(0,T) \\ \partial_t\gamma_i(s,t) = U(\gamma_i(s,t),t), s\in[0,M], t\in]0,T], \forall i & \\ \gamma_i(s,0) = \gamma_0^i(s), s\in[0,M] & \end{cases} \tag{1.25}$$

This model of immersed elastic membrane consists in a model for multi-fluid flows supplemented by the elastic force term. The model used in this work is described in Chapter 7, it has been introduced in [32] it is inspired by the immersed boundary model, a level set method is used in order to locate and evolve the interface and the elastic forces are expressed in terms of the level set function. The obtained model is described in a fully eulerian formulation allowing to use a Cartesian grid mesh to represent the fluid/structure domain.

1.3 Front tracking and Front capturing methods

Two different approaches can be adopted to represent the interface: the Front tracking and Front capturing method.

The Front tracking methods provide an explicit representation of the interface by means of Lagrangian markers. The Lagrangian points are initially distributed on the interface ("Surface" tracking) or inside the structure (or the fluid) ("Volume tracking methods"). To evolve the structure, the velocity is interpolated from the Eulerian grid points to the Lagrangian markers allowing to advect the Lagrangian points.

Unlike the Front-tracking methods, the Front capturing methods give an implicit representation of the interface. In this class of methods, a scalar field captures the interface and is transported in order to evolve the structure, those methods differ by the choice of the scalar quantity and the technique to reconstruct the interface. For instance, in the Volume of Fluid method [79], this function describes the volume fraction of a particular phase in each grid cell of the all computational domain. At each time step, the function is reconstructed according to the volume fraction of the phase, then it is advected and the volume fraction is reinitialized. In the phase field method [3], the interface is regularized, each phase is characterized by a constant physical parameter allowing to evolve and locate each phase. Another popular approach is the level set method, this is the one we based our work on, we will detail this approach in the next section.

1.3.1 The Front-tracking methods

We consider that the fluid/structure domain Ω is discretized by a time-independent Eulerian mesh grid. The interface is followed by a set of Lagrangian points. Denoting by X_i the position of a particle i, this particle is advected by the local velocity U_i as:

$$\frac{dX_i}{dt} = U_i \tag{1.26}$$

where the velocity U_i is interpolated from the Eulerian grid points to the Lagrangian markers.

The Marker and Cells method

The Marker and Cells has been firstly introduced in [75] to simulate free surface incompressible flows. The fluid/structure domain is discretized by a fix Cartesian grid. Initially, the Lagrangian markers are uniformly distributed in the structure subdomain Ω_1 and then advected by (1.26). The reconstruction of the interface is based on the presence of marker fields in the Eulerian grid cells. Later, this method has been successfully and widely applied to multi-fluid flows. This technique is efficient to conserve the volume and to distinguish the different medias. However, compared to the resolution of the Eulerian mesh, a large number of Lagrangian particles have to be introduced. Moreover, in the presence of large deformations, additional procedures are required in order to distribute regularly the markers on the interface and inside the structure. As a consequence, this method can induce a high computational cost in particular if several immersed structures are considered and/or in the three dimensional case.

The Surface tracking method

The Front tracking method, inspired by the MAC method has been developed by Dali in [37]. Instead of distributing the particles in the all structure domain, the Lagrangian markers are only located on the interface. Then, the interface is reconstructed by high order interpolations of curves passing through these points. Thus, the location of the interface is more precise. However, topological changes are not directly taken into account and again procedures to distribute regularly the markers on the interface are required.

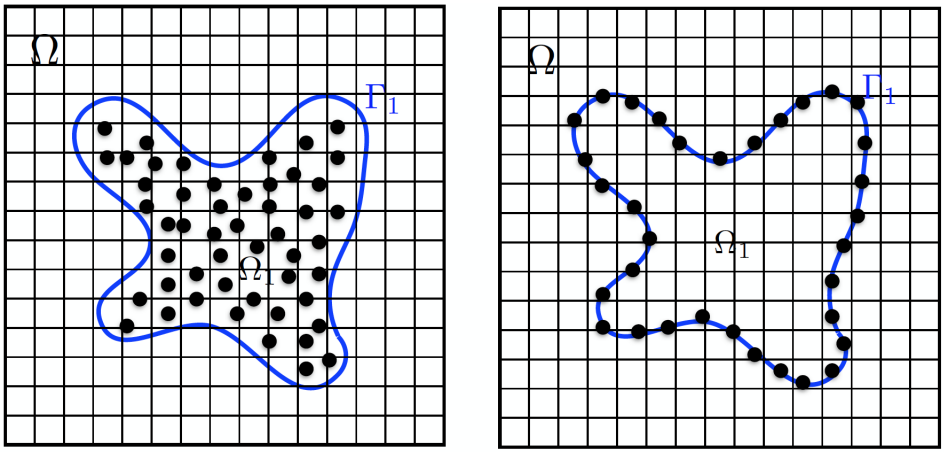


Figure 1.6: Volume tracking vs Surface tracking methods. The black points represent the lagrangian markers.

1.3.2 Front capturing methods

The basic idea of the Front capturing methods is to use a scalar function to locate the interface, this scalar field takes different values in each media. A fix Eulerian mesh grid represents the entire domain Ω and the scalar function is transported by the velocity of the fluid or the structure. The topology changes are taken into account by the scalar function. Moreover, the choice of this scalar quantity may allow to compute explicitly normal or curvature of the curve, to conserve the physical properties of the structure like mass conservation. We present in what follows three popular capturing methods: the phase field method, the Volume of Fluid method and the Level set method which will be present in the next section with more details.

VOF methods

The Volume of Fluid method ([79]) describes the volume fraction of the fluid in each grid cell of the all computational domain. A volume fraction function ϕ is initially equal to 0 in the structure and 1 in the fluid. This function is advected by the fluid velocity field U as:

$$\partial_t \phi + U \cdot \nabla \phi = 0 \ \text{ in } \Omega$$

The transport equation changes the values of the scalar field at the interface, its values will be comprised between 0 and 1 at the grid cells close to the interface. As the volume fraction field ϕ is discontinuous, standard upwind finite-difference schemes can not be used as it would diffuse the interface. Instead, according to the volume fraction of each grid cell, the interface is reconstructed.
There exist various methods for the geometric approximation of the interface. The most famous approach consists either in reconstructing the segments parallel to the mesh grid cells (Simple Line Interface Calculation), [150] or the segments obtained by means of the normal to the interface (Piecewise Linear Interface Calculation). Several numerical schemes have been designed to advect the reconstructed interface [150], those schemes are mostly based on eulerian advection on structured grids. Although mass conservation is accurately achieved with the VOF method, the computation of the geometrical properties of the curve like normal or curvature is not accurate [150]. Moreover, this method is difficult to implement for unstructured grids and in the three dimensional case.

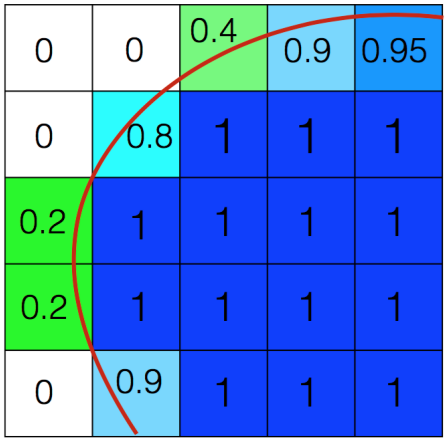


Figure 1.7: Illustration of the VOF methods. The red curve is the fluid/structure interface.

Phase field methods

The phase field approach belongs to the class of diffuse interface methods like for instance the volume tracking method. A non physical scalar function is introduced in order to distinguish between the interior of the structure and the fluid region. This function assumes two distinct constant values in each region (for instance here -1 in the fluid and 1 in the structure) and varies continuously from these two values across a thin boundary layer, which is used to described the structure interface. Most of the phase field models are based on a free energy functional which depends on the phase field ϕ and a diffusive field, in its simplest form this energy is provided by:

$$E = \frac{\alpha}{2}|\nabla\phi|^2 + \frac{\beta}{4}(\phi^2 - 1)^2. \tag{1.27}$$

The advantages of this approach is that the topology changes are automatically handled, the geometric quantities are computed explicitly using the phase field and the extension to dimension three is easily achieved. However, even if the phase field varies continuously at the interface, a fine resolution is required close to the interface to capture high gradients.

1.4 Level set method

The level set method, pioneered by Osher and Sethian in [138], is very popular to treat problems involving interfaces. It is widely used for numerical analysis of surfaces and shapes and in the context of fluid/structure interaction.
The books [137] and [157] gives a detailed description of the method and an overview of its possible applications.

The general idea of the level set method is to define a scalar function in the all computational domain that assumes a 0 value on the location of the interface to capture. Then, this function is transported in order to evolve the interface by taking into account the governing equations of the model. At each moment, the 0 value of the level set function gives the location of the interface. Usually, this function is defined or reconstructed as a signed distance function to the interface, which is regular in each subdomains.

This implicit representation of the interfaces offers several advantages. An interesting one is that it automatically takes into account topology changes and the global representation of the curve allows to consider a curve with several pieces. It permits in particular to handle easily large deformations.
The dimensions two and and higher are treated very easily, in the three-dimensional case the level line 0 is a surface.
Moreover, numerically, the level set function is discretized on a fixed grid, allowing to use simple finite differences schemes to discretize quantities which depend on ϕ. However, the accuracy of the information we have on the surface is not the same everywhere and several procedures have to be considered in order to keep an accurate and robust method.

1.4.1 The level set function

To introduce the level set method we consider a bounded domain $\Omega \in \mathbb{R}^d$ ($d = 2$ or $d = 3$) partitionned into two sub domains Ω_1 and Ω_2. We denote by Γ_1 the boundary of Ω_1 and Γ the boundary of Ω, giving $\Omega = \Omega_1 \cup \Omega_2 \cup \Gamma$. We denotes by ϕ the level set function which captures the interface Γ_1. This function is usually chosen to be negative (respectively

positive) inside Ω_1 and positive (respectively negative) outside, and to assume a 0 value on the interface:

$$\begin{cases} \phi(x) < 0 & x \in \Omega_1 \\ \phi(x) = 0 & x \in \Gamma_1 \\ \phi(x) > 0 & x \in \Omega_2 \end{cases}$$

Moreover, in order to have a differentiable function in a neighborhood of the interface, it is usually initialized as a signed distance function:

$$\phi(x) = \begin{cases} -d(x, \Gamma_1) & x \in \Omega_1 \\ d(x, \Gamma_1) & x \in \Omega_2 \end{cases} \tag{1.28}$$

where

$$d(x, \Gamma_1) = \min_{y \in \Gamma_1} \|x - y\|.$$

By using a signed distance function, the obtained results avoid under-resolution problem and are more accurate., as the computation of geometrical properties of the curve such as curvature or normal.

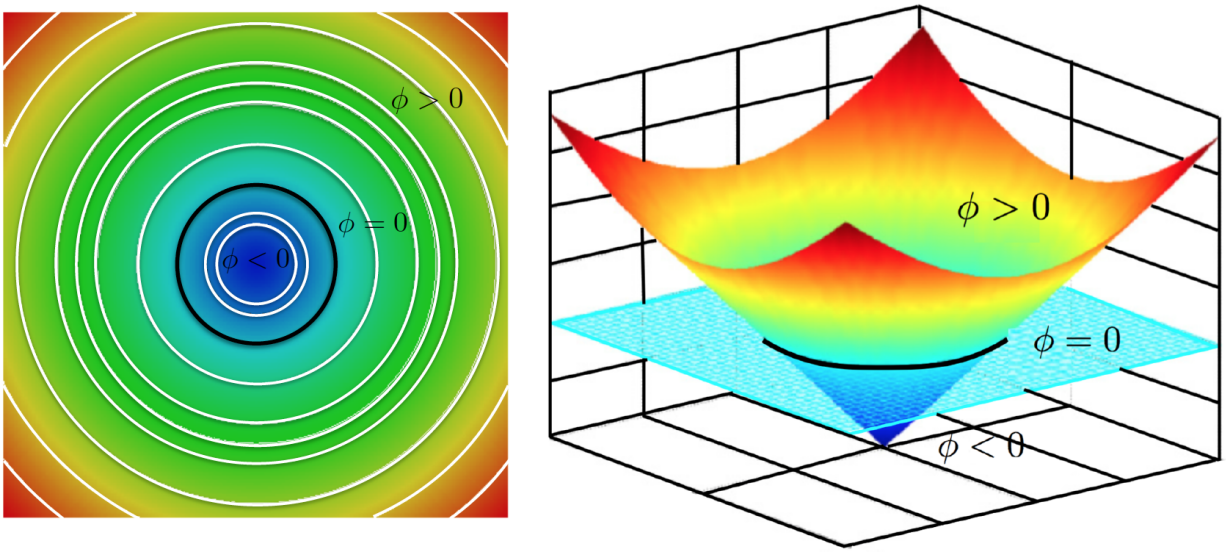


Figure 1.8: Representation of the level set function. The left picture shows the level set function on a plane where the black contour is the zero value of the level set function.

1.4.2 Numerical tools

Using an implicit function to capture the interface tells us directly to which region belongs a point x with the help of a Heaviside function H and its corresponding Dirac function ζ applied to ϕ. The Heaviside function is defined as:

$$H(\phi) = \begin{cases} 0 & \text{if } \phi \leq 0 \\ \frac{1}{2} & \text{if } \phi = 0 \\ 1 & \text{if } \phi > 0 \end{cases} \tag{1.29}$$

In practice, a regularized version of these functions H_ε and ζ_ε is used on the interface in order to reduce grid effects:

$$H_\varepsilon(\phi) = \begin{cases} 0 & \phi \leq -\varepsilon \\ \frac{1}{2}(1 + \frac{\phi}{\varepsilon} + \frac{\sin(\frac{\pi\phi}{\varepsilon})}{\pi}) & |\phi| \leq \varepsilon \\ 1 & \phi \geq \varepsilon \end{cases} \tag{1.30}$$

$$\zeta_\varepsilon(\phi)) = \begin{cases} 0 & \phi \leq -\varepsilon \\ \frac{1}{2\varepsilon}(1 + \cos(\frac{\pi\phi}{\varepsilon})) & |\phi| \leq \varepsilon \\ 0 & \phi \geq \varepsilon \end{cases} \tag{1.31}$$

where ε represents the interface half thickness. The justification of this regularization is described in [177] following the ideas introduced by Peskin in [143]. In the literature, this value is usually taken equal to $1.5h$ where h denotes the space discretization step.

The geometrical characteristics of the curve such as normal vectors n and curvature κ are obtained explicitly using the level set function:

$$n = \frac{\nabla\phi}{|\nabla\phi|}, \quad \kappa = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right).$$

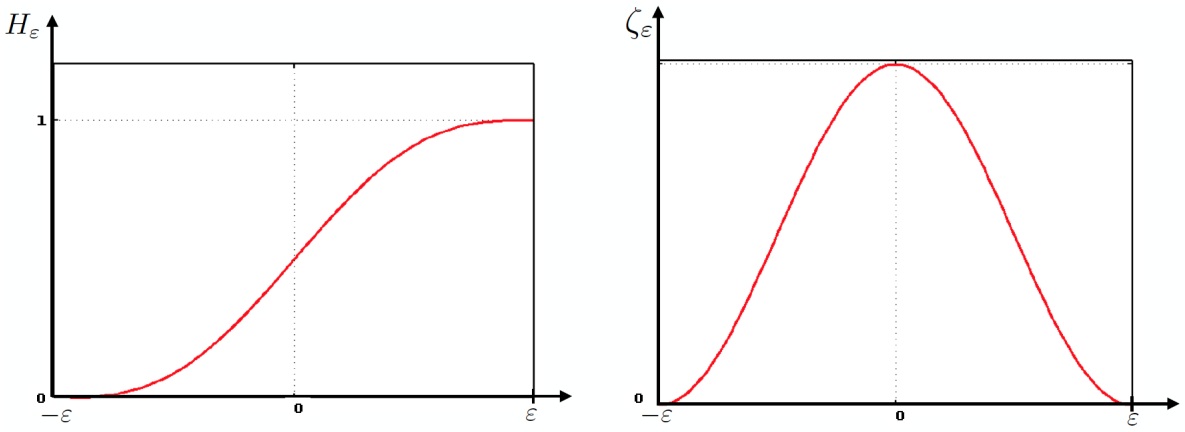


Figure 1.9: Representation of the regularized heaviside and delta functions H_ε and ζ_ε.

1.4.3 Transport equation

The evolution of the interface Γ_1 is based on the transport of the level set function ϕ by the velocity of the continuous medium. The velocity field can depend on the space, time, geometric properties of the curve and/or the physics of the problem. For instance, in a problem of fluid/structure interaction this velocity field corresponds to the fluid and/or the structure velocity field. Denoting by U an incompressible velocity field, the level set function is evolved by:

$$\begin{cases} \partial_t\phi + U \cdot \nabla\phi = 0 & \text{in } \Omega_t \\ \phi(\cdot, 0) = \phi_0 & \text{in } \Omega \end{cases} \tag{1.32}$$

For general velocity field U, the solution of this equation may have a loss of regularity with respect to the initial smoothness of ϕ_0. It does not conserve the level set function as a signed distance function (i.e. $|\nabla\phi| = 1$). In order to solve this problem, several methods have been introduced.

1.4.4 Reinitialization of the level set function

When ϕ is a signed distance function, its value gives a useful information on distance to interface. However as stated above, this property could be lost while transporting ϕ. We give here some existing methods to recover this information.

Renormalization of the level set function

In [33; 139], it has been shown that the function $\frac{\phi}{|\nabla\phi|}$ approaches the signed distance function close to the interface ($\phi = 0$).
This method is very useful as it gives the first order signed distance value close to the interface, allowing to compute correctly the Heaviside and cut off function H_ε, ζ_ε.
It has been used in several applications, including numerical simulations of immersed vesicles [32; 115]. In [115], one immersed vesicle is considered, the velocity is quite regular and the information on the signed distance function is only required in a neighborhood of the interface. Consequently, using $\frac{\phi}{|\nabla\phi|}$ is sufficient for this application.
If the values of the signed distance function is required far from the interface and/or if the velocity is too irregular, this approach is no longer sufficient. Indeed, when the velocity is too irregular, the gradient magnitude of the level set function can take large or very small values leading to significant errors. For instance, if $|\nabla\phi| << 1$, computation of $\frac{\phi}{|\nabla\phi|}$ could lead to increasing numerical error, while for $|\nabla\phi| >> 1$ so that the term $\frac{\phi}{|\nabla\phi|}$ becomes close to zero, artificial new portion of interface could be created in the computational domain. In this scenario, a reinitialization of the level set function is still required. As the level set function will be updated it is more accurate to use the term $\frac{\phi}{|\nabla\phi|}$ for the computation of H_ε, ζ_ε.

Modification of the transport equation

An other idea proposed in [72] is to replace the advection equation of ϕ by the following equation:

$$\partial_t\phi(x,t) + (U\cdot\nabla\phi)(x - \phi\nabla\phi, t) = 0 \tag{1.33}$$

With this advection equation, the interface $\phi = 0$ is still evolved with the velocity field U and ϕ is at each time a distance function. However, the non-locality of this equation makes it difficult to use in the context of fluid mechanics.

Fast marching method

The fast marching method is a very robust method to reinitialize the level set function as a signed distance function. It has been presented in 1996 by Sethian in [156]. The principle of the fast marching method is to compute the distance function from the

smallest values (close to the 0 value of the interface) to the greatest ones. The distance is propagated from the interface to far regions by taking care of never recomputing the points already assigned by a distance value. More precisely, we look for the solution of the following eikonal equation:

$$\begin{cases} |\nabla \phi| = 1 & \text{in } \Omega \\ \phi = 0 & \text{on } \Gamma \end{cases} \tag{1.34}$$

The numerical resolution of this equation presented in [156] is inspired by the Djikstra method introduced by Tsitsiklis in [172]. The Djikstra algorithm developed in [172] allows to compute the minimal short path between two points, by treating only one time each grid point of the domain, its computational complexity is $o(n \log(n))$ where n represents the number of grid points.

The numerical scheme used to solve the equation (1.34) is the one introduced in [151]. In the two-dimensional case, denoting by Δx, Δy the horizontal and vertical discretization space steps, the numerical scheme is formulated as follows:

$$\max\left(\max(D_x^- \phi_{ij}, 0)^2, \min(D_x^+ \phi_{ij}, 0)^2\right) + \max(\max(D_y^- \phi_{ij}, 0)^2, \min(D_y^+ \phi_{ij}, 0)^2)\big) = 1 \tag{1.35}$$

$$D_x^- \phi_{ij} = \frac{\phi_{ij} - \phi_{i-1,j}}{\Delta x}, D_x^+ \phi_{ij} = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x},$$

$$D_y^- \phi_{ij} = \frac{\phi_{ij} - \phi_{i,j-1}}{\Delta y}, \; D_y^+ \phi_{ij} = \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y}.$$

This is a first-order scheme.

The scheme (1.35) implies that the information moves in one direction only : from the lowest values of ϕ to the highest ones. In order to propagate the distance inside and outside the interface object, one can perform the algorithm twice, that is first compute the distance function outside (the points inside the object are considered as Alive), then multiply by -1 the distance function to propagate the distance inside.

To achieve the unique pass algorithm, the computational domain Ω is partitioned into three subdomains: Alive, Narrow-Band and Far Away points (see Figure 1.10). Those sets of points are defined by:

- Alive (A): set of points on which the values of ϕ are known
- Narrow-Band (NB): set of points were the values of d are unknown and the distance to A is less than one space discretization step.
- Far away (F): all others points of the domain and at those points we set $\phi = -\infty$

In order to have a consistent initialization of the fast marching procedure, we take the following boundary conditions for the eikonal equations:

$$\phi^r = \frac{\phi}{|\nabla \phi|} \tag{1.36}$$

The procedure is sketched as in the following algorithm.

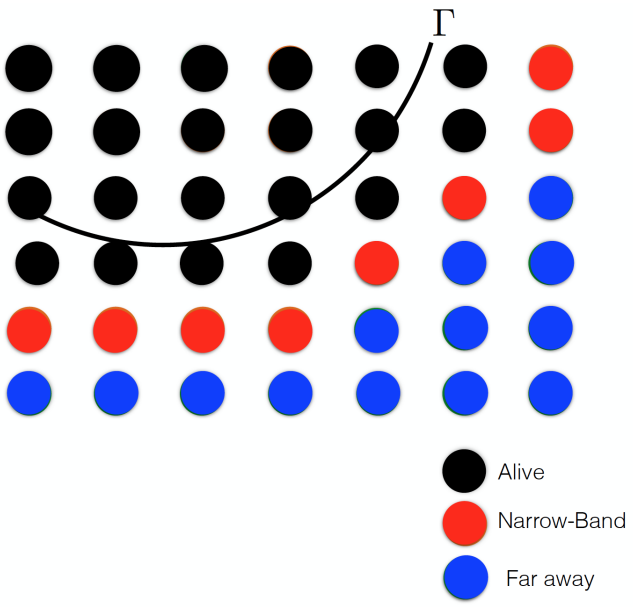


Figure 1.10: Partition of the computational domain.

Algorithm 1 *Initialization phase*

- Closest points at the interfaces are tagged as Alive and at those points the values of ϕ are fixed by (1.36).
- Add the neighbors of the Alive points in the Narrow Band.
- At the Far Away points, assign the value ∞ to ϕ.

Algorithm 2 *Iterative phase*

- Find the point x_m such as $\phi(x_m) = \min_{(x \in NB)} \phi(x)$.
- Delete x_m from the NB.
- Compute the new values of ϕ at the neighbors of x_m.
- If the neighbors of x_m are in the Far away set, add them in the NB.

The speed of the fast marching procedure relies on the sorting algorithm to find the minimum value of the Narrow-Band set. The Narrow-Band set is defined as a Fibonnaci binary heap. Thus, the first element of the NB heap directly gives the point x_m at which ϕ have the minimum value of the Narrow-Band set: $\phi(x_m) = \min_{\{x \in NB\}} \phi(x)$.
Two procedures are required: one to suppress the minimum value of the NB and one to add elements in the Narrow-Band. Numerically, a circular linked list is used to define the

NB heap, allowing to reduce or increase the size of the heap and it provides access to the first and last element directly.
When the first element is suppressed, the last element of the heap (having the maximum distance value of NB) is placed at the head on the heap. Then, a heap sort procedure is achieved, the element goes down in the list until it reaches its right position.
When an element is added to the NB, it is placed at end of the list, and the element goes up in the list until its right position.
A detailed description of the implementation of this procedure can be found for instance in [181].

The complexity of the Fast marching algorithm is $O(n\log(n))$ where n denotes the number of grid points of the domain, this complexity is given in the worst case that is when all the points of the domain are contained in the Narrow-Band. Denoting by n_b the number of grid points in the Narrow-Band of one interface, the complexity is $O(n_b\log(n_b))$, if we propagate k interfaces this complexity becomes $O(kn_b\log(kn_b))$, in this case it seems better to achieve k fast marching procedures as the complexity is reduced $O(kn_b\log(n_b))$. However, depending on the number of interfaces k, the fast marching procedure for the k interfaces will stop propagation most fastly, as in the second case each interface propagates in the all domain whereas in the first one the interfaces are propagated simultaneously.

Resolution of a Hamilton Jacobi equation

An other popular method to reinitialize the level set function ϕ consists in solving a Hamilton-Jacobi equation on the level set field. This method has been introduced in [167] and has been largely used and improved since then like for instance in [49; 165; 181]. The Hamilton-Jacobi equation can be formulated as:

$$\partial_t\phi - \mathrm{sgn}(\phi)(1 - \nabla\phi\cdot\frac{\nabla\phi}{|\nabla\phi|}) = 0 \tag{1.37}$$

The level set function is transported with a velocity $\dfrac{\nabla\phi}{|\nabla\phi|}$ where the sign function sgn permits to propagate the front in the two directions (inside and outside), in [167] it is defined as:

$$\mathrm{sgn}(\phi) = \frac{\phi}{\sqrt{\phi^2+\varepsilon_s^2}}$$

where ε_s is a smoothing parameter.

An advantage of this approach, is that it is easy to implement as the same schemes used to transport the level set function are used. However, different parameters has to be chosen like ε_s, the frequency of reinitialization and the time step used to discretize (1.37). No explicit values exist, thus different numerical tests have to be performed in order to find the right parameters.
Moreover, a significant loss of mass is observed after long time simulations.

Additionnal numerical procedure

Reinitialization procedures are not always sufficient to conserve the volume of the structures. An additional numerical procedure introduced in [163] can be used and consists

in using an iso line of the level set function in the vicinity of $\phi = 0$ in order to keep a constant volume without changing the form of the interface. At each time $t^{n+1} = (n+1)\Delta t$, the distance between the two isolines is computed as:

$$C_\phi^{n+1} = \frac{V^{n+1} - V^n}{\int_{\Gamma_1} d\Gamma_1^n} \tag{1.38}$$

where $V^{n+1} - V^n = \int_{\Omega_1^{n+1}} dx - \int_{\Omega_1^n} dx$ the new level set function is computed as:

$$\phi^{n+1} = \phi^n + C_\phi^{n+1} \tag{1.39}$$

Indeed, at each time using the regularized Heaviside function, the volume is given by:

$$V(\phi) = \int_\Omega H_\varepsilon(-\phi)dx$$

and for $C \in \mathbb{R}$ close to zero we have:

$$V(\phi + C) = V(\phi) + C\frac{d}{dC}V(\phi + C)_{|C=0} + o(C)$$

where :

$$\frac{d}{dC}V(\phi + C)_{|C=0} = \int_\Omega -\frac{1}{\varepsilon}\zeta_\varepsilon(\phi)dx$$

This last term corresponds to the length of the regularized interface. Thus this condition is avalaible if C_ϕ is sufficiently small and the level set function has to be a signed distance function close to the interface. This technique avoid the errors accumulations which can lead to a great mass loss.

1.4.5 Hybrid methods

Hybrid level set methods were developed in order to overcome the main weakness of the level set method which lies in its inaccuracy to achieve volume conservation. In regions where the fluid flow is under-resolved, the level set function can not show accurately if characteristics separate, merge or are parallel which leads to volume loss. To overcome this problem, several techniques have been designed.

The coupled Level set/VOF methods [166] combines the advantage of the VOF methods for volume conservation while preserving those of the level set method for the geometric properties of the curve. Both a level set function and a volume fraction are advected by the velocity field. Then, the VOF interface is reconstructed by using the normals computed with the level set function. Finally, the level set function is reinitialized as a signed distance function to the reconstructed interface.

Introduced in [49], the Particle Level set Method consists in using a level set function to capture the interface and a set of a Lagrangian particles are seeded in a vicinity of the interface to correct the location of the interface. The Lagrangian particles are initially located on each side of the interface and are labelled with the sign of ϕ. Then, those markers are transported in a Lagrangian manner, and the level set function is transported with the continous velocity field. When a particle switchs to the other side of the interface (for instance if a particle labelled by $\phi > 0$ changes its label), the level set function is corrected using an algorithm which takes into account the particle positions.

1.5 Numerical resolution

We present here the numerical resolution of the Navier-Stokes equations and of the transport equation of the level set functions with the fluid velocity. For a sake of simplicity we place ourselves in two dimension $\Omega \in \mathbb{R}^2$ and we consider the Navier-Stokes equations for variable densities and viscosities :

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U) - \nabla \cdot (\mu D(U)) + \nabla P = f & \text{in } \Omega \times (0,T) \\ \nabla \cdot U = F & \text{in } \Omega \times (0,T) \\ \partial_t \phi + U \cdot \nabla \phi = 0 & \text{in } \Omega \times (0,T) \end{cases} \tag{1.40}$$

where ρ and μ are defined in (1.20).

f is a force which depends on the level set function ϕ. Here, we do not take into account the immersed structures, only their interfaces are captured and evolved by the level set functions.

1.5.1 Discretization MAC and space discretization

The system is discretized by a finite difference method on a staggered grid (MAC type). The pressure function is located at the center of the mesh cells and the velocity at the center of the sides and the level set function is either located at the center of the cells for a coarser grid or we use a finer grid for the level set function. The finer grid is used when the model requires high derivatives of the level set functions see Figure 1.11 in particular we use this discretization for applications to elastic membranes.

The divergence free condition is computed at the pressure point which enforces the volume constraint accurately.

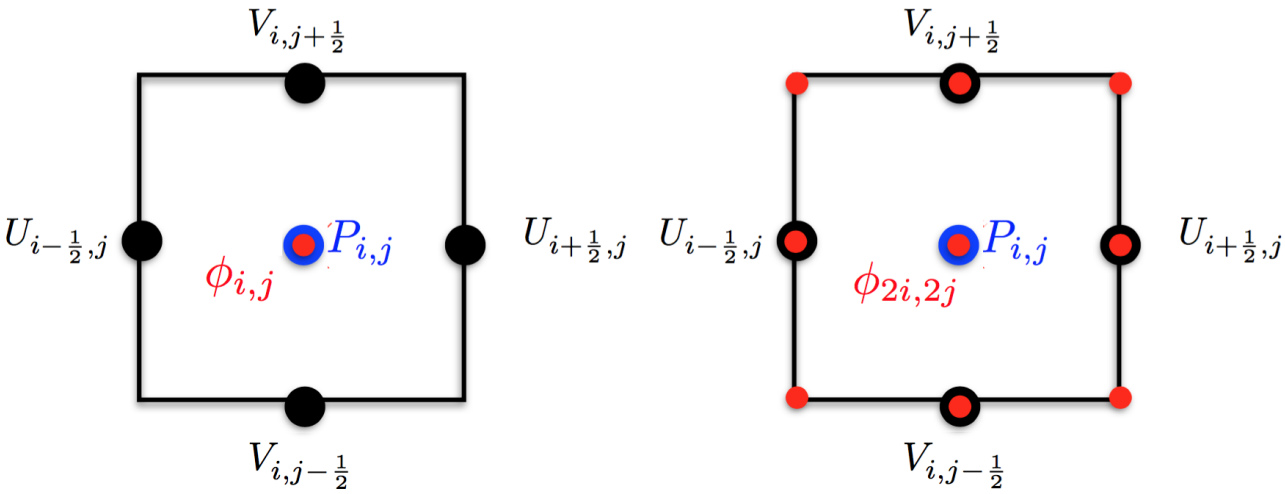


Figure 1.11: Eulerian coarse (left) and fine (right) grid for the level set function.

1.5.2 WENO schemes for the transport equation

The level set function is transported by means of the equation (4.2). This equation is either discretized with a WENO5 scheme or with ENO 3 scheme and an Euler explicit scheme or a Runge Kutta of order 2 scheme in time are used.

The ENO schemes introduced in [159; 160] use different stencils of the solution in order to compute the derivatives, depending on the local regularity of the solution one of this approximation is chosen. Later, in [112] the WENO schemes have been proposed to improve the ENO schemes which suffers from unstable computation of the solution in the regular regions. This is achieved by taking a convex combinaison of the three different approximations of the derivatives. The computation of the weights depends continuously on the regularity of the solution on each candidate stencil. In the regions where the solution is irregular the interpolations are affected with a weight close to zero, giving the robust ENO schemes. In the regions where the solution is regular, the weights are homogeneous for the three approximations. The weight depends on the local regularity of the solution.

To describe the WENO and ENO schemes we consider an explicit Euler scheme in time of the transport equation in the two dimensional case. Let $\Delta x, \Delta y$ be the horizontal and vertical discretization step of the Cartesian mesh. We denote by Δt the time discretization step, $u_{i,j}$ and $v_{i,j}$ the horizontal and vertical component of the velocity at the grid point (i,j) and by $\phi_{i,j}$ the level set function at the grid point (i,j). The first order explicit scheme of the transport equation 4.2 is formulated as:

$$\frac{(\phi_{i,j}^{n+1} - \phi_{i,j}^{n})}{\Delta t} + u_{i,j}^{n}\partial_x(\phi_{i,j}^{n}) + v_{i,j}^{n}\partial_y(\phi_{i,j}^{n}) = 0$$

This scheme is stable under the CFL condition:

$$\Delta t < \frac{\Delta x}{\max(|u|,|v|)}$$

Adopting an upwing approach and denoting by $D_x^-\phi$ the derivatives at left and $D_x^+\phi$ the derivatives at right we get:

$$\partial_x\phi_{i,j} \simeq \begin{cases} D_x^-\phi_{i,j} & \text{if } u_{i,j} < 0 \\ D_x^+\phi_{i,j} & \text{if } u_{i,j} > 0 \end{cases}$$

For the computation of the derivative $D_x^-\phi$, the ENO 3 schemes choose one of the three following discretizations:

$$D_x^-\phi_{i,j}^1 = \frac{1}{3}v_1 - \frac{7}{6}v_2 + \frac{11}{6}v_3$$

$$D_x^-\phi_{i,j}^2 = -\frac{1}{6}v_2 + \frac{5}{6}v_3 + \frac{1}{3}v_4$$

$$D_x^-\phi_{i,j}^3 = -\frac{1}{3}v_3 + \frac{5}{6}v_4 - \frac{1}{6}v_5$$

where:

$$v_1 = \frac{(\phi_{i-2,j} - \phi_{i-3,j})}{\Delta x},\ v_2 = \frac{(\phi_{i-1,j} - \phi_{i-2,j})}{\Delta x},\ v_3 = \frac{(\phi_{i,j} - \phi_{i-1,j})}{\Delta x}$$

$$v_4 = \frac{(\phi_{i+1,j} - \phi_{i,j})}{\Delta x},\ v_5 = \frac{(\phi_{i+2,j} - \phi_{i+1,j}}{\Delta x}$$

For the computation of the derivative $D_x^+\phi$, the ENO 3 schemes choose one of the three following discretizations:

$$D_x^-\phi_{i,j}^1 = \frac{1}{3}v_1 - \frac{7}{6}v_2 + \frac{11}{6}v_3$$

$$D_x^- \phi_{i,j}^2 = -\frac{1}{6}v_2 + \frac{5}{6}v_3 + \frac{1}{3}v_4$$
$$D_x^- \phi_{i,j}^3 = -\frac{1}{3}v_3 + \frac{5}{6}v_4 - \frac{1}{6}v_5$$

where:

$$v_1 = \frac{(\phi_{i+3,j} - \phi_{i+2,j})}{\Delta x},\ v_2 = \frac{(\phi_{i+2,j} - \phi_{i+1,j})}{\Delta x},\ v_3 = \frac{(\phi_{i+1,j} - \phi_{i,j})}{\Delta x}$$

$$v_4 = \frac{(\phi_{i,j} - \phi_{i-1,j})}{\Delta x},\ v_5 = \frac{(\phi_{i-1,j} - \phi_{i-2,j})}{\Delta x}$$

whereas the WENO 5 schemes choose a convex combinaison of these three discretizations, the computation of $\partial_x \phi$ with the WENO 5 scheme is provided by the algorithm 3.

1.5.3 Resolution of the fluid flow equations

The inertial terms of the Navier Stokes equation $U \cdot \nabla U$ are either discretized with a WENO5 schemes described above or with a centered scheme, the diffusion terms with a second order centered schemes and a projection method is used to decouple the fluid velocity field and the pressure field.

Explicit projection method

The projection method introduced in the 1960's by Chorin and Temam, is a very popular and widely used method to solve the incompressible Navier-Stokes equations.
This efficient approach allows to decouple the fluid velocity and the pressure. Thanks to the decoupling, fast elliptic solvers can be used to compute the equation on the pressure field.
Instead of computing simultaneously the momentum equation and the incompressibility constraint, this technique consists in first computing the momentum equation for an intermediate velocity and then project this intermediate velocity on a space of divergence-free velocities. Given a time step Δt, we set $t^n = n\Delta t$ and $U^n \approx U(., t^n)$. The first order explicit scheme is given by:

- **Step 1** : Compute an intermediate state U^*

$$U^* = U^n + \Delta t(-U^n \cdot \nabla U^n + \frac{\nabla \cdot (\mu D(U^n))}{\rho} + \frac{f}{\rho}) \tag{1.41}$$

- **Step 2** : Solve the pressure

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho}\right) = \frac{\nabla \cdot U^*}{\Delta t} \tag{1.42}$$

- **Step 3** : Projection of the velocity on a space of divergence-free velocities.

$$U^{n+1} = U^* - \frac{\Delta t}{\rho}\nabla p^{n+1} \tag{1.43}$$

Algorithm 3 *WENO 5 scheme*

To compute $D_x^- \phi$ set:

$$v_1 = \frac{(\phi_{i-2,j} - \phi_{i-3,j})}{\Delta x},\ v_2 = \frac{(\phi_{i-1,j} - \phi_{i-2,j})}{\Delta x},\ v_3 = \frac{(\phi_{i,j} - \phi_{i-1,j})}{\Delta x}$$

$$v_4 = \frac{(\phi_{i+1,j} - \phi_{i,j})}{\Delta x},\ v_5 = \frac{(\phi_{i+2,j} - \phi_{i+1,j}}{\Delta x}$$

To compute $D_x^+ \phi$ set:

$$v_1 = \frac{(\phi_{i+3,j} - \phi_{i+2,j})}{\Delta x},\ v_2 = \frac{(\phi_{i+2,j} - \phi_{i+1,j})}{\Delta x},\ v_3 = \frac{(\phi_{i+1,j} - \phi_{i,j})}{\Delta x}$$

$$v_4 = \frac{(\phi_{i,j} - \phi_{i-1,j})}{\Delta x},\ v_5 = \frac{(\phi_{i-1,j} - \phi_{i-2,j})}{\Delta x}$$

Compute the regularity coefficients as:

$$\begin{cases} S_1 &= \frac{13}{12}(v_1 - 2v_2 + v_3)^2 + \frac{1}{4}(v_1 - 4v_2 + 3v_3)^2 \\ S_2 &= \frac{13}{12}(v_2 - 2v_3 + v_4)^2 + \frac{1}{4}(v_2 - v_4)^2 \\ S_3 &= \frac{13}{12}(v_3 - 2v_4 + v_5)^2 + \frac{1}{4}(3v_3 - 4v_4 + v_5)^2 \end{cases}$$

Setting:

$$a_1 = \frac{1}{10}\frac{1}{\varepsilon + S_1},\ a_2 = \frac{1}{10}\frac{1}{\varepsilon + S_2},\ a_3 = \frac{1}{10}\frac{1}{\varepsilon + S_3}$$

The weights are given by:

$$\begin{cases} w_1 &= \frac{a_1}{a_1 + a_2 + a_3} \\ w_2 &= \frac{a_2}{a_1 + a_2 + a_3} \\ w_3 &= \frac{a_3}{a_1 + a_2 + a_3} \end{cases}$$

Finally, the horizontal derivatives are computed by setting:

$$D_x^{+,-}\phi = w_1\left(\frac{v_1}{3} - \frac{7v_2}{6} + \frac{11v_3}{6}\right) + w_2\left(-\frac{v_2}{6} + \frac{5v_3}{6} + \frac{v_4}{3}\right) + w_3\left(\frac{v_3}{3} + \frac{5v_4}{6} - \frac{v_5}{6}\right)$$

This ensures that $\mathrm{div}(U^{n+1}) = 0$. For a constant density, the equation (1.42) is a Poisson equation and can be solved using fast Poisson solvers. The solver used is FISHPACK which is adapted for MAC grid, it is a Poisson solver based on the Fast Fourier Transform ([168]). This solver allows to resolve numerically problems of the form: $-\Delta U + BU = F$ where B is a space constant function. In the case of a variable density function, the equation (1.42) is solved with a preconditioned conjugate gradient method.

Boundary conditions have to be imposed to the pressure and intermediate velocity in order to impose the correct boundary conditions on the exact velocity U^{n+1}.
For instance, using the explicit scheme for the Navier Stokes equations with a homogeneous boundary condition on the velocity and constant density, we set:

$$\begin{cases} U^* & = 0 \ \text{ on } \Gamma \\ \dfrac{\partial p^{n+1}}{\partial n} & = 0 \ \text{ on } \Gamma \end{cases} \tag{1.44}$$

giving:

$$U^{n+1} \cdot n = 0 \ \text{ on } \Gamma$$

where n is the normal to Γ. Thus, an error of order one on time is introduced as $U^{n+1}.t \neq 0$ where t is the tangential vector. As the diffusive terms are explicited, a condition on the time step of the form $\Delta t \leq CRe\Delta_x^2$ has to be used. This type of schemes can be used for moderate Reynolds number.

Implicit projection method

When the Reynolds number is small, in particular for biological applications, an implicit scheme is required.
The first order implicit scheme for the diffusion term is written by:

$$U^* = U^n + \Delta t \left(-(U^n \cdot \nabla U^n) + \frac{(\nabla \cdot (\mu D(U^*)) + f)}{\rho} \right) \tag{1.45}$$

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho} \right) = \frac{\mathrm{div}\, U^*}{\Delta t} \tag{1.46}$$

$$U^{n+1} = U^* - \frac{\Delta t}{\rho} \nabla p^{n+1} \tag{1.47}$$

When the viscosity is variable and the density function is constant $\rho = \rho_c$ or if we can apply a Boussineq Approximation, a trick is to decompose the viscous terms as:

$$\nabla \cdot (\mu D(U^*) \simeq \Delta U^* + \nabla \cdot ((\mu - 1) D(U^n)) \tag{1.48}$$

so that using the FISCHPACK library, we can solve the following equation on U^*:

$$U^* - \frac{\Delta t}{\rho_c} \Delta U^* = U^n + \Delta t(-(U^n \cdot \nabla U^n) + \frac{\Delta t}{\rho_c} \left(\nabla \cdot ((\mu - 1) D(U^n)) + \frac{f}{\rho}) \right) \tag{1.49}$$

If on the contrary the viscosity is constant between two fluids we define the average of the density as $\rho_0 = (\rho_1 + \rho_f)/2$ and we solve:

$$\nabla \cdot (\mu D(U^*) \simeq \frac{\mu}{\rho_0} \Delta U^* + (\frac{\mu}{\rho^n} - \frac{\mu}{\rho_0}) \Delta U^* \tag{1.50}$$

Finally we can compute U^* as:

$$U^* - \Delta t \frac{\mu}{\rho_0} \Delta U^* = U^n - \Delta t(-(U^n \cdot \nabla U^n) + (\frac{\mu}{\rho^n} - \frac{\mu}{\rho_0})\Delta U^* + \frac{f}{\rho_0}) \tag{1.51}$$

When using the implicit scheme, one has to be careful with the imposed boundary conditions imposed to the pressure and U^*.
Indeed, for the case of homogeneous boundary conditions, if we impose:

$$U^* = 0, \quad \frac{\partial p^{n+1}}{\partial n} = 0 \ \text{ on } \Gamma \tag{1.52}$$

and denotes by n the normal to the boundary Γ and using the equations, we obtain:

$$U^{n+1} = 0 \text{ or } \frac{\partial p^{n+1}}{\partial n} = 0$$

This induces significant errors close the boundary Γ, see for instance [36; 66; 186]. To remedy to this problem, a large amount of researchs has been dedicated to the improvement of the Chorin Temam projection method. For instance, in [99; 130] the boundary condition of the velocity U^* is modified by introducing a pressure gradient. In [96; 136], projection schemes based on an exact condition on the pressure field are developed.

Conclusion

In this chapter, we have described some numerical methods for fluid/structure interaction. In this work, we adopt a fully eulerian formulation of the fluid/structure interaction. Within this framework, we used the Level set method to capture interfaces. Among the various Front-capturing methods, the level set technique is the most adapted to this work for several reasons. First, the level set function can be defined as a signed distance function providing informations on the distance to the interface. Secondly, geometrical properties of the curve like the normal and curvature are explicitly provided by the derivatives of the level set function. Moreover, other properties like the stretching are explicitly computed by means of the level set function as we will see in Chapter 6. An other advantage of this method for this work, is that one level set function can captures several interfaces. Moreover, as we have seen, the level set method is easy to implement and the obtained solvers can be fast if the appropriate precautions are taken. We also present the numerical resolution of the fluid equations and the transport equation of the level set function. A time-independent Cartesian mesh is used to discretize the equations with two different resolutions for the level set function: the finer grid will be used for the simulation of immersed vesicles as high derivatives of the level set function are required. The resolution of the incompressible Navier-Stokes equations is achieved by means of a projection method of Chorin type, the implicit scheme for the diffusion term is used for the simulation of immersed elastic membranes, whereas for moving rigid bodies the explicit scheme is more convenient.

Chapter 2

Domain Labeling

Contents

In this chapter, we present the model that allows to locate each structure and its neighbor in the all fluid/structure domain. This approach is an adaptation of the multi geometric deformable model (MGDM) [11] that was introduced by J. Bogovic for image analysis. The MGDM method is a multi-object extension to the conventional geometric level set formulation of active contours [23].

We consider a bounded domain $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) that contains N structures immersed in a fluid. We denote by $(\Omega_i)_{i=1,..,N}$ the N structures and the surrounding fluid is denoted by Ω_{N_f} where $N_\mathrm{f} = N + 1$. The entire fluid/structures domain Ω is partitioned into $N + 1$ objects as follows:

$$\left\{ \begin{array}{rcl} \forall i \neq j, \Omega_i \cap \Omega_j & = & \emptyset \\ \Omega_{N_\mathrm{f}} & = & \Omega \backslash \overline{\{\bigcup_{i=1}^{N} \Omega_i\}} \\ \Gamma_{N_\mathrm{f}} & = & \bigcup_{i=1}^{N} \Gamma_i \bigcup \partial\Omega. \end{array} \right. \tag{2.1}$$

2.1 Level set models to capture multiple interfaces

As discussed in the last chapter, the level set method is an efficient approach to capture and evolve structures in a domain. When dealing with several structures one has to choose the right level set model to capture all the structures and alleviate the computational cost due to an increasing number of particles. Moreover, being able to distinguish each structure in the domain and preventing numerical contacts is necessary for the applications we aim at dealing with. We present here some existing level set models which allows to capture and evolve multiple interfaces.

2.1.1 One level set function

One level set function can capture an arbitrary number of interfaces between the cells and the fluid. Let ϕ be the level set function which captures the union of the N cells Ω_i, $i \in \{1, ..., N\}$, it is defined by:

$$\phi(x) = \begin{cases} -d(x, \cup_{i=1}^{N} \Gamma_i) & x \in \cup_{i=1}^{N} \Omega_i \\ d(x, \cup_{i=1}^{N} \Gamma_i) & \text{elsewhere.} \end{cases}$$

Using one single level set function yields a very low computing time as it is independent on the number of cells. In the fluid/structure framework, the level set function can be transported by the fluid velocity. However, while an advantage of the level set method is to handle automatically changes of topology, in our case this property is problematic as splitting or merging of interfaces are directly taken into account by the level set function. Indeed, if two cells are too close the curvature and the normal are not well computed and a strong drawback is that we do not have any information on the distance between cells. Moreover, it is not possible to specify a different velocity model and/or a force for each cell only based on the level set value. So this capturing approach can lead to collision and merging of the structures.

2.1.2 Level set decomposition

Another way to capture multiple interfaces is to use one level set function for each cell. Then, each cell interface Γ_i is captured by one level set function ϕ_i. Thus, it is possible to specify a different speed or force to a cell Ω_i by using the level set function ϕ_i and we get the distance between all the cells.
Indeed, the distance between two cells Ω_j and Ω_i is given by the level set functions ϕ_i and ϕ_j as follows:

$$\forall x \in \Omega_i, \phi_j = d(x, \Gamma_j) \quad \text{or} \quad \forall x \in \Omega_j, \phi_i = d(x, \Gamma_i)$$

This multiple level set decomposition has been widely used in image segmentation and in fluid/structures interaction. A big inconvenient is its computational cost when the number of cells increases as N transport equations and N reinitialization are required to evolve the structures and to maintain the N level set functions as signed distance functions.

2.1.3 $\log_2(N)$ level set

An interesting approach has been introduced in [184] , $\log_2(N)$ level set functions are used to capture N interfaces. This idea is based on the four colour theorem. For instance for seven particles, we have eight regions and these regions can be described by three level set functions using the sign of these functions, see Figure 2.1. However, each level set function captures multiple interfaces. As a result we do not have information on all the distance between the particles.

For image segmentation where the distance is not required it is a very desirable model as it reduces the computational cost compared to the level set decomposition (N level set functions).

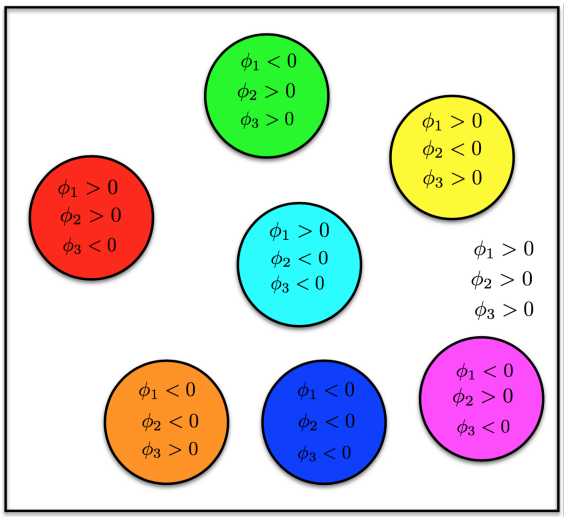


Figure 2.1: $\log_{2N}$ level set model.

2.2 Starting point of the method

As stated above, one can consider N level set functions to capture the N structures. Let ϕ_i denotes the level set function which captures the interface Γ_i.
The starting idea of the MGDM method [11] is based on one simple observation: each level set function $\phi_i(x)$ can be decomposed into several distance functions depending on the location of x in the domain Ω.
Let us take the example of three bodies immersed in a fluid Ω_{N_f}, as illustrated in Figure 2.2.
Let the level set function ϕ_1 be a signed distance function :

$$\forall x \in \Omega,\ \phi_1(x) = \begin{cases} -d(x,\Gamma_1) & x \in \Omega_1 \\ d(x,\Gamma_1) & \text{otherwise} \end{cases} \tag{2.2}$$

where:

$$d(x,\Gamma_1) = \min_{y\in\Gamma_1} \|x-y\|.$$

If a point x is in Ω_1, then $\phi_1(x) = -d(x,\Gamma_1) = -d(x,\Gamma_{N+1})$. Therefore, the absolute value of $\phi_1(x)$ corresponds to the distance to the first closest object Ω_{N+1}, that is the object corresponding to the fluid.

If x is in Ω_2, $\phi_1(x)$ is either the distance to the second closest object to x (in the left region of Ω_2) or the distance to the third closest object to x (in the right region of Ω_2).

Similarly, for a point x belonging to the body Ω_3, $\phi_1(x)$ identifies the distance to the third closest object.

Therefore, depending on the location of x in the domain, the level set function ϕ_1 is either the distance to the first, the second or the third closest objects. In the same way, ϕ_2 and ϕ_3 can be decomposed into these three distance functions.
To extend this approach to an arbitrary number of structures N, one can introduce a set of label maps L_0, L_1, L_2,..,L_N as:

$$\forall x \in \Omega, \forall i \in \{1, ..., N+1\}, \quad \begin{cases} L_0(x) & = i \text{ if } x \in \Omega_i \\ L_1(x) & = \arg\min_{j \neq L_0(x)} d(x, \Gamma_j) \\ L_2(x) & = \arg\min_{j \notin \{L_0(x), L_1(x)\}} d(x, \Gamma_j). \\ : & : \\ . & . \\ L_N(x) & = \arg\min_{j \neq L_k(x), k=0,...,N-1} d(x, \Gamma_j). \end{cases}$$

The label map L_0 provides a partition of the whole computational domain Ω into $N+1$ different objects. The label maps $L_k(x), k = 1, .., N$ identifies the index of the kth closest object for any point x of the whole computational domain Ω. For each point x in Ω, if $L_k(x) = i$ then the object Ω_i is the kth closest object to x.
By means of these label maps one can recover the N signed distance functions as follows.

$$\forall x \in \Omega, \forall i \in \{1, ..., N+1\}, \quad \phi_i(x) = \begin{cases} -d(x, \Gamma_{L_1(x)}) & \text{if } i = L_0(x) \\ d(x, \Gamma_{L_1(x)}) & \text{if } i = L_1(x) \\ d(x, \Gamma_{L_2(x)}) & \text{if } i = L_2(x) \\ : & : \\ . & . \\ d(x, \Gamma_{L_N(x)}) & \text{if } i = L_N(x) \end{cases}$$

For a point x in Ω, $d(x, \Gamma_{L_i(x)})$ is the distance to the ith closest object to x. In the

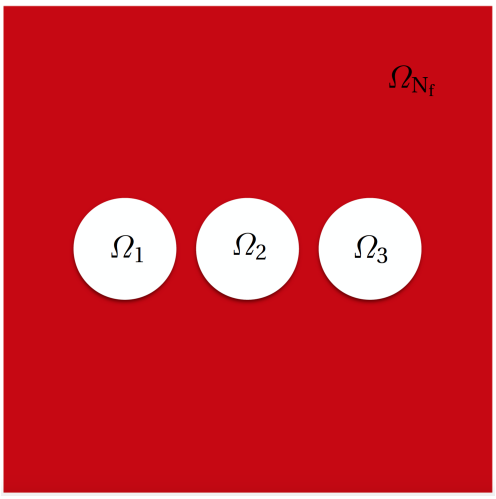


Figure 2.2: Partition of the domain Ω for a configuration of three bodies.

case of two immersed bodies, the label maps L_0, L_1 ,L_2 allow to recover exactly the two level set functions. The proposed model only use the three label maps L_0, L_1 and L_2 and there associated distance functions, in particular to recover the N level set functions in a neighborhood of interfaces.

2.3 The three label maps

At every point x of the fluid/structure domain Ω, the label maps L_0, L_1 and L_2 are defined as:

$\forall x \in \Omega, \forall i \in \{1, ..., N+1\}$,

$$\begin{cases} L_0(x) & = i \text{ if } x \in \Omega_i \\ L_1(x) & = \underset{j \neq L_0(x)}{\arg\min}\, d(x, \Gamma_j) \\ L_2(x) & = \underset{j \notin \{L_0(x), L_1(x)\}}{\arg\min}\, d(x, \Gamma_j). \end{cases}$$

or equivalently:

$$\begin{cases} L_0(x) = i & \text{if } x \in \Omega_i \\ L_1(x) = j & \text{if the first closest object to } x \text{ is } \Omega_j \\ L_2(x) = k & \text{if the second closest object to } x \text{ is } \Omega_k. \end{cases}$$

The label map $L_0(x)$ gives the object number to which the point x belongs. The label map $L_1(x)$ and $L_2(x)$ identifies the index of the first and second closest object for any x in the whole computational domain Ω. As a consequence, the label map function L_2 gives the index of the first closest cell for any x in the whole computational domain. Moreover, for each point of a solid the label map L_1 corresponds to the index of the fluid.

$$\forall x \in \Omega, \forall i \in \{1, ..., N\}, \begin{cases} L_0(x) & = i \text{ if } x \in \Omega_i \\ L_1(x) & = N_f \\ L_2(x) & = k \text{ if } \Omega_k \text{ is the first closest structure to } x. \end{cases}$$

The Figure 2.3 gives an illustration of the label maps L_0, L_1 and L_2 related to the Figure 2.2. The red color represents the object related to the fluid. For each point of a cell the label map L_1 corresponds to the red color. Moreover, the label map L_2 gives the index of the first closest cell for any x in the whole computational domain. For instance, if a point x belongs to the green body ,the label map L_2 corresponds to the closest cell that is either the yellow one (in the left region) or the blue one (in the right region). For all points x belonging to the blue body, the first closest cell is the green one. Likewise, for all points x in the yellow body, the label map L_2 corresponds to the green one.

The Figure 2.4 shows an illustration of the three label maps in the case of five structures.

We can see that for all points belonging to the surrounding four cells (green, yellow, purple and light blue objects), the closest structure is always the blue one. On the contrary, the label map L_2 has partitioned the blue object into four regions, each of them giving the color of the closest structure. Therefore, the three label maps provide an interesting local description of the entire fluid/structure domain Ω. At each point of the fluid/structure domain we have the index of the first and second closest object. This local representation of objects has to be completed with the two distance functions associated to L_1 and L_2.

Figure 2.3: Illustration of the three label maps L_0, L_1, L_2 from left to right.

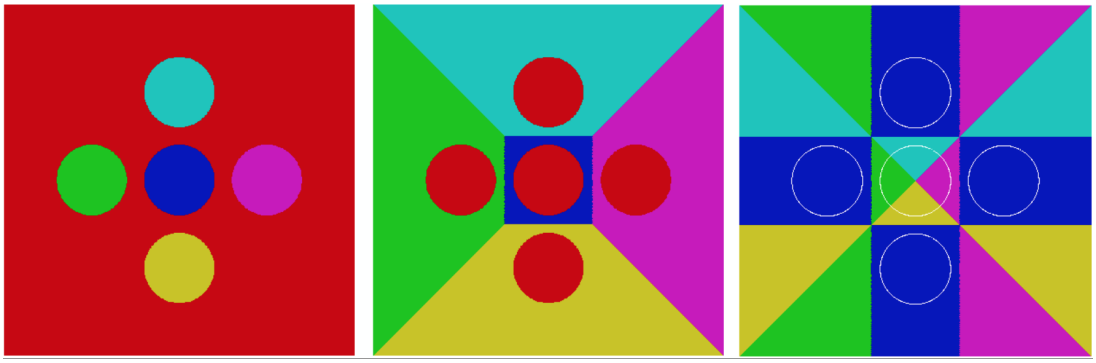


Figure 2.4: Illustration of the three label maps for a configuration of five cells, from left to right: L_0, L_1 and L_2. The white contour represents the boundary of the cells.

2.4 Distance functions

The distance functions associated to the first and second closest object are:

$$\forall x \in \Omega, \begin{cases} \varphi_1(x) = d(x, \Gamma_{L_1(x)}) \\ \varphi_2(x) = d(x, \Gamma_{L_2(x)}). \end{cases} \tag{2.3}$$

The distance $\varphi_1(x)$ is the distance from x to the first closest object's boundary $\Gamma_{L_1(x)}$ and $\varphi_2(x)$ is the distance from x to the second closest object's boundary $\Gamma_{L_2(x)}$. At any point of the domain Ω, the distance function φ_1 captures the union of all cells interfaces and φ_2 provides the distance to the first closest cell. As a consequence, on each point of a cell, we have the distance to the closest one allowing to define a collision model to the closest interface. The Figure 2.5 shows an illustration of these two distance functions associated to the label maps 2.3. The distance φ_1 captures all interfaces and provides the distance to the closest one at all points of the domain. For a point x belonging to the green object, the distance function φ_2 provides either the distance to the blue object (in the left region where the label map L_2 corresponds to the blue object) or to the yellow one (in the right region where the label map L_2 corresponds to the blue object). Similarly, for all points belonging to the blue and the green object, φ_1 is the distance to the green one.

Figure 2.6 shows an illustration of the two distance functions related to the label maps of the Figure 2.4. If we take a point x belonging to the blue object, as this object is partitioned into four regions by the label map L_2 (see figure 2.4), φ_1 is in the yellow region the distance to the yellow object, in the green region the distance to the green object and so forth. For all points belonging to the four surrounding bodies that is the yellow, green,

purple and light blue objects, φ_1 is the distance to the blue object.

The three label maps, L_0, L_1, L_2 and the two distance functions φ_1 and φ_2 provide information concerning the location of each structure in the domain Ω and the distance to the first closest structure (φ_2).

These informations are sufficient to handle numerical collisions between cells as will be shown later. In fact, avoiding contacts is equivalent to require:

$$\forall x \in \Omega \ , \quad \varphi_2(x) > 0.$$

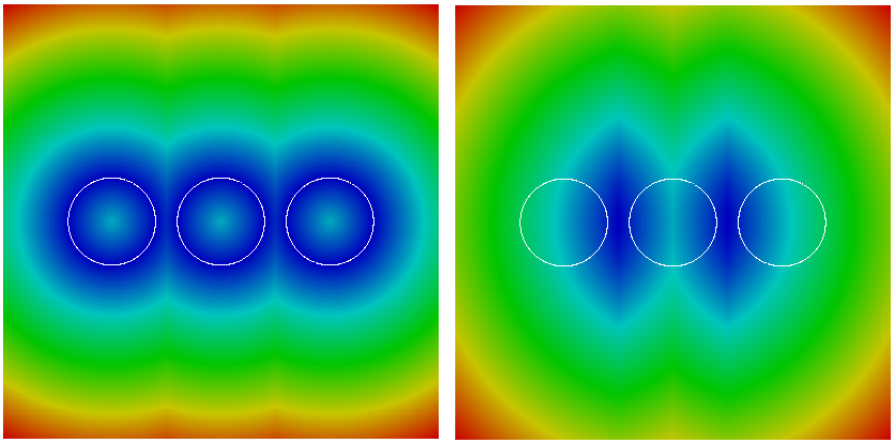


Figure 2.5: Illustration of the two distance functions for a configuration of three circular cells, from left to right: φ_1 and φ_2. The white contour represents the boundary of the cells.

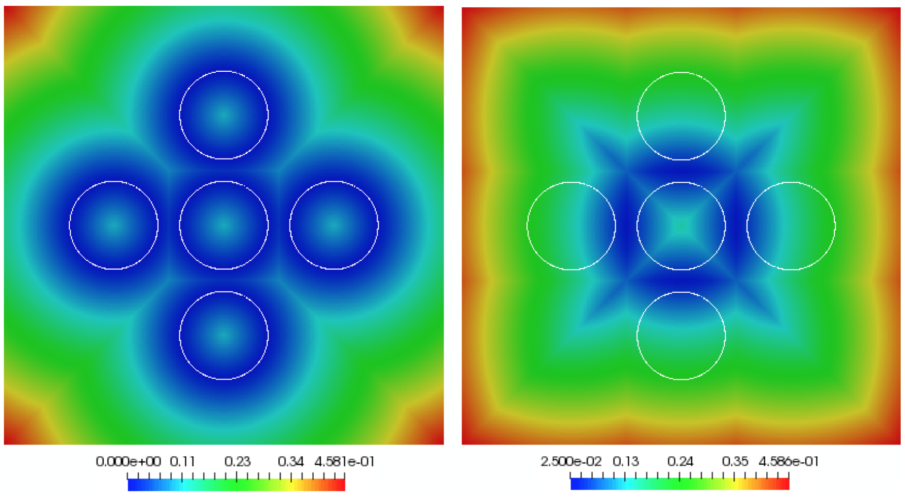


Figure 2.6: Illustration of the two distance functions for a configuration of four ellipse, from left to right: φ_1 and φ_2. The white contour represents the boundary of the circular cells.

2.5 Reconstituted level set functions

For some applications, we need to specify a speed or a force independently for each structure, in that case it is more convenient to recover the N level set functions close to

the bodies interfaces.
For instance, an application to rigid bodies requires the computation of the rigid velocities for each structure.
Using the two distance functions the N level set functions can be defined as:

$$\forall x \in \Omega, \forall i \in \{1, ..., N+1\}, \quad \phi_i(x) = \begin{cases} -\varphi_1(x) & \text{if } i = L_0(x) \\ \varphi_1(x) & \text{if } i = L_1(x) \\ \varphi_2(x) & \text{if } i = L_2(x) \\ C & \text{otherwise} \end{cases}$$

where $C >> 1$. These functions are negative inside each object, positive outside and Lipschitz continuous. Therefore, they are still level set functions, and in a neighbourhood of each object, each level set function is a signed distance function. These functions are explicitly defined by the distance φ_1 and φ_2, thus instead of evolving each level set function one can also evolve the two distance functions and the three label maps.
An illustration of these functions is given in the Figure 2.7 in the case of three bodies having the same configuration as in Figures 2.3 and 2.5. The white contour represents the boundary of the cells. The constant C is fixed to one for illustration. We can see that on the left region of the fluid/structure domain Ω, ϕ_1 is a true signed distance function to the boundary Γ_1. In fact, inside the body Ω_1 (the blue object) ($L_0 = 1$), $\phi_1 = -\varphi_1$ where φ_1 is the distance to the interface Γ_1. Then, while Ω_1 is the first closest object (L_1 corresponds to the blue object), or the second closest object (L_2 corresponds to the blue object), the level set function ϕ_1 corresponds to the signed distance function to the boundary Γ_1. Likewise, on the right part of the domain Ω, ϕ_3 corresponds to the signed distance function associated to the boundary of Ω_3. In this configuration, the level set field ϕ_2 is a true signed distance function in the all computational domain, as for all point x in $\Omega \backslash \Omega_2$, the green object Ω_2 corresponds either to the first closest object ($L_1(x) = 2$) or to the second one ($L_2(x) = 2$) (see Figure 2.3).

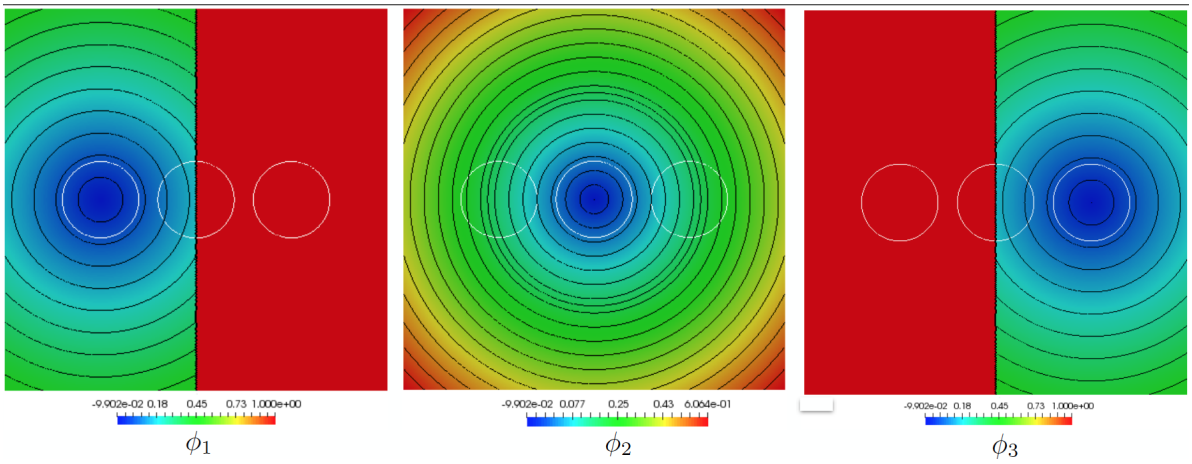


Figure 2.7: Illustration of the three reconstituted level set functions for a configuration of three cells (Figure 2.2).

Conclusion

In this chapter, we have described the interface capturing method proposed by J. Bogovic and exploited it in the context of multiple structures immersed in a fluid. The label L_0 provides the location of each structure in the domain, the distance function φ_1 allows to capture the union of all interfaces by means of the label map L_1 which gives the indice of the first closest object. The distance field φ_2 associated to L_2 provides the distance between the closest structures at all points of the fluid/structure domain. These five functions will enable us to define a collision model in order to avoid numerical contacts between the structures.

Chapter 3

Collision strategy

Contents

For the numerical simulation of more than one particle evolving in a fluid, a numerical strategy is required to avoid contacts and overlaps of the smooth particles although theory [77; 78] demonstrated that two perfectly smooth particles can not enter in contact in finite time, this phenomenon is due to the presence of a thin-layer of viscous fluid in the interstitial gap between two close particles.

This chapter is organized as follows:
Section 1 is dedicated to the hydrodynamical interactions, in particular we present some existing methods developed to prevent numerical contacts. In Section 2, the collision model introduced in [31] is described. Section 3 is devoted to the description of the proposed collision model to avoid contacts between the particles. Finally, in the last section we give a comparison of the collision model [31] and the proposed model, an estimation of the error between the two forces is given and the error induces on the velocity flow is investigated by considering two Stokes problem.

3.1 Lubrication forces

Within a dense suspension of particles immersed in a fluid, different forces and phenomena are responsible for the motion of the particles.

When the effect of the fluid is neglible, only the gravity force and the interaction between particles are responsible for their motion. This is called the frictional regime, and corresponds to the case where the energy dissipated by friction is of the same order than the energy dissipated by the relative motion of the particles. Two types of interactional forces are possible between particles: the collision force and the friction force which obeys

to the Coulomb Law.

Otherwise, the viscous regime corresponds to the case where the particles are driven by the fluid flow. The velocity field of each moving particle is transmitted to the fluid flow which induces forces and torques on the other particles. In the context of several immersed structures, the hydrodynamical forces play an important role in the dynamic of the bodies. In particular, these forces are responsible for the increase of the viscosity in dense suspensions.

The Hydrodynamical Interactions (HIs) have different characteristics which render them difficult to deal with. First, the behaviour of these forces is different if two particles are close or far from each other. These forces are long-range attractive forces and become repulsive forces when the gap between two particles is small. These repulsive forces are called lubrication forces. Moreover, the HIs are changed between a pair of particles if a third one is present in their vicinity. From a numerical point of view, in the context of an Eulerian fluid/structure model, the HIs are well computed as long as the particles are far enough from each other. We focus here on these short-range repulsive forces. We first discuss the origins of the HIs. Then, we describe some theoretical and numerical models used in the litterature to modelize these forces and to include them in numerical simulations.

3.1.1 Origins of the lubrication forces

Lubrication forces play a key role when two or several bodies are close to each other. When two particles are approaching each other, the interstitial fluid can not be evacuated, as a consequence high pressure gradient develops in the interstitial gap in order to squeeze out or push in the fluid, opposing the relative movement of the particles. High pressure gradients develop in the form of Poiseuille-type flow and depend on the relative motion of the particles, an illustration is given in Figure 3.1. The action of this force is dissipative as there is no force if the particles are steady.

This force is proportional to the inverse of the separation distance between the particles, to the viscosity of the fluid, to the relative velocity and to the size of the particles. Indeed, when the separation distance becomes smaller and the viscosity greater, the opposition of the fluid to the relative movement of the particles is more important. Moreover, the greater the relative velocity between the particles is, the greater the force will be and its sign is associated to the relative velocity. Its dependence on the size of the particles is straightforward, as if we take the case of a particle Ω_2 falling on an immobile sphere Ω_1, the larger the size of Ω_2 is, more it is difficult for the interstitial fluid to get away leading to an increase of the resistance of the fluid flow.

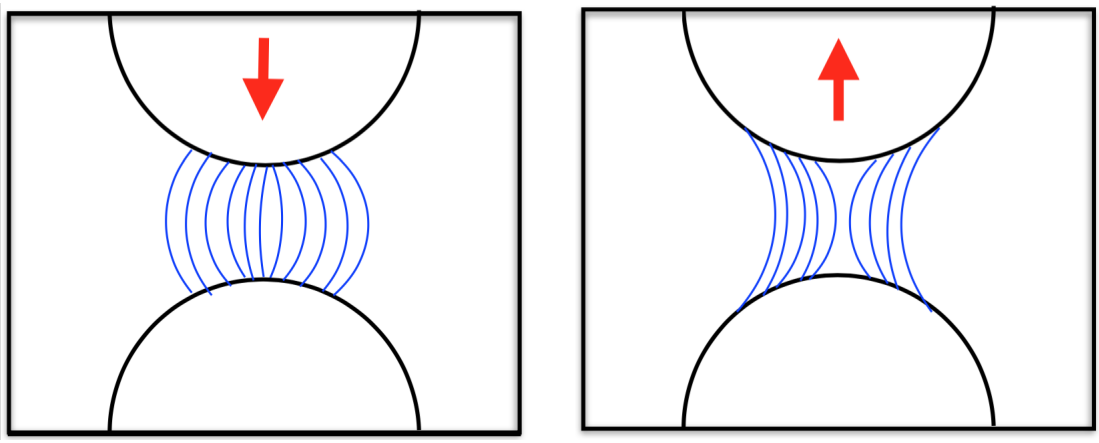


Figure 3.1: Fluid flow between two smooth spherical particles near contact, for two different relative motions.

3.1.2 Analytical models

To understand the lubrication interactions and to integrate them in numerical simulations, analytical models were developed to compute these lubrication forces correctly. These models were mostly developed in the context of low Reynolds number in order to use Stokes equations. The basic idea is to expand the analytical solutions for the lubrication forces when the distance between two solids tends to zero. Thanks to the linearity of Stokes equations, the hydrodynamic forces/torques are linear functions of the translational and rotational velocity U of the particle.

The first analytical models were developed for the interaction of a sphere and a plane-wall immersed in a Stokes flow.
In [16] and [120] the exact expression of the lubrication force is given, a development at large distance values is achieved in the case of a sphere close to a wall evolving in a Stokes flow.
In [69], an asymptotic development of the solutions of the Stokes equation for translational and rotational motions of a sphere is achieved. The authors provide the second term of the lubrication force when the distance between the sphere and the plane wall tends to zero. The test case of a sphere falling on a inclined wall is numerically investigated, and results confirm that the sphere can not be in contact with the wall.
All these models provide an exact solution of the lubrication forces in the dimension two with a first term formulated as:

$$F_{\text{lub,w,2D}} = -6\pi\mu\frac{rU}{\delta} \tag{3.1}$$

where μ is the dynamic viscosity of the fluid, r is the radius of the particle, δ the separation distance between the particle and the wall and U the velocity of the particle.
In fact, the first term $-6\pi\mu rU$ corresponds to the drag applied on a sphere moving in a quiescent fluid given by Stokes in 1851. In [135] and [30] a correction of this approximation to $\delta\ln(\delta)$ is proposed.

The three-dimensional case has also been studied in [69], for a separation distance

$\delta << r^2$, the authors obtain the following approximation:

$$F_{\text{lub,w,3D}} \sim -6\pi\mu r^2 \frac{U}{\delta} \tag{3.2}$$

The lubrication force is proportional to the size of the particles, the viscosity of the fluid, the inverse of the separation distance and the relative velocity of the particles. Several experiments (see for instance[2; 193]) confirm these approximations.

For two spheres approaching each other, in [35], an asymptotic development on the power of the separation distance gives an equivalent of this force when it tends to zero. For an immobile sphere of radius r_1 and sphere of radius r_2 moving with a velocity U_2, one gets the first term of this force as:

$$F_{\text{lub,p,2D}} = -6\pi\mu \frac{r_1^2 r_2^2}{(r_1+r_2)^2} \frac{U_2}{\delta} + O(\ln(\delta)) \tag{3.3}$$

where μ is the dynamic viscosity of the fluid and δ the separation distance between the two spheres. Correction to $\ln(\delta)$ acting on the spheres have been determined in [117; 134]. Then, the authors in [89] provide an asymptotic solution at order $\delta\ln(\delta)$ for two converging spheres of different radii. The case of relative translation and rotation has also been considered in [30; 134]. For two disks Ω_1 and Ω_2 having the same radius $r_1 = r_2 = r$, the normal lubrication force [30] due to relative translation is expressed as:

$$F_n^{2,1} = -6\pi\mu r \left(\frac{1}{2\delta} - \frac{9}{20}\ln(\delta) - \frac{3}{56}\delta\ln(\delta) + O(1) \right) \times (U_{2,1}\cdot e_{1,2})e_{1,2} \tag{3.4}$$

where $e_{1,2}$ denotes the normal vector along the line of the centers of the particles, δ is the separation distance and $U_{2,1} = U_2 - U_1$ is the relative velocity.
Following [134], the tangential component of the lubrication force due to the relative translation along the line of centers is:

$$F_{t,n}^{2,1} \simeq -6\pi\mu r (U_{2,1} - (U_{2,1}\cdot e_{2,1}).e_{2,1})\ln(\delta) \tag{3.5}$$

Denoting by w_1 and w_2 the rotational velocity of the particles and $w_{1,2}$ the relative rotation about an axis perpendicular to the line of centers, the associated tangential force is:

$$F_{t,w}^{2,1} = \frac{\pi\mu r^2}{5}(w_{1,2} - 4(w_1+w_2)) \times e_{2,1})\ln(\delta) \tag{3.6}$$

It appears that the component of the lubrication forces having the most effect on the particles is the normal one as it is proportional to $\frac{1}{\delta}$ whereas the dominant term of the translational and rotational component is $\ln(\delta)$. Thus, it tends more slowly to infinity when the distance δ tends to zero.
In fact, the tangential force exerted by the fluid on a body is a viscous shear force. That is this force depends on the viscosity of the fluid and the roughness of the surface of the particle. In this work, we only consider smooth particles.

Following [100] , the normal and tangential lubrication force acting on a particle Ω_i by the body Ω_j can be expressed as:

$$F_{i\to j} = -F_{j\to i} = -k(D_{ij})(U_{ij}.e_{ij}).e_{ij} - k^{\perp}(D_{ij})(U_{ij}.e_{ij}^{\perp}).e_{ij}^{\perp} \tag{3.7}$$

where:

$$U_{ij} = U_i - U_j, \quad k(D_{ij}) = \frac{C_{ij}}{D_{ij}}, \quad k^{\perp}(D_{ij}) = C_{ij}^{\perp} \ln \frac{\varepsilon}{D_{ij}}$$

The coefficient C_{ij} depends on the local radii of curvature of the particles and on the viscosity of the fluid, ε corresponds to the characteristic size of the bodies and gives the distance of activation of the forces. For two spherical particles Ω_i and Ω_j of radius r_i and r_j immersed in a fluid of viscosity μ, the coefficient C_{ij} is:

$$C_{ij} = 6\pi\mu \frac{r_i^2 r_j^2}{(r_i + r_j)^2}$$

This first order approximation of the lubrication force has been used for instance in [121]. The first step of the method proposed in [121] is to compute the distance between the particles, then the lubrication forces are computed one by one for each particle and finally the velocities and particles positions are updated. In the case of a large number of particles, this approach can induce a large computational cost and can lead to overlap.

3.1.3 Numerical strategies

The lubrication forces avoid collisions of bodies, during numerical calculations, the accumulation of space and time discretization errors does not allow to capture accurately these forces.

A first approach proposed in [81] is to refine the mesh in the inter particle gap in order to resolve accurately these lubrication forces. However, this strategy leads to a high computational cost as several refinements are necessary and the frequency of refinements is not known a priori. Moreover, the interparticle distance can be very small (around 10^{-5} times the size of the particles [128]). Consequently, using this approach in the case of a large number of particles in particular in three dimensions looks impossible, the only application in three-dimensions has been achieved in [90] for five spheres. Even with the body-fitted mesh approach, dealing with a large number of particles induces a high computational cost.

An other technique is to add the theoretical lubrication forces only when the gap between the bodies is less than a given distance. Thus, when the particles are far from each other the hydrodynamical forces are well computed by the fluid solver, and when the intertitial gap becomes small a correction of the uncomputed lubrication forces is added. The value of the smaller gap width for which we can not compute correctly these lubrication effects has been determined in the case of a sphere approaching a wall [16] or for two spheres [30] by using asymptotic expansions of the analytical solutions. In [121], the lubrication forces are added to the numerical model and act like a correction of the lubrication forces when the particles get close. The considered lubrication forces are an extension of the asymptotic expansion introduced in [100] for the case of multiple particles. In this approach overlap of the particles can occur and if it happens the particles are artificially separated and kept at a certain small distance.

The same approach is used in [39], the lubrication forces are computed using the distances between particles and then incorporating into the fluid equations. Only the pair-wise particles being at a critical distance are considered, in order to reduce the computational cost. However, this approach is not sufficient to deal with numerical contacts. Indeed, as the lubrication forces are singular, the obtained system is stiff and numerical errors due

to time discretization leads to numerical contacts. To remedy this problem, a short-range repulsive force is added.
The correction of the lubrication forces has been largely used in [17; 98; 133; 170; 192]. Most of these models authorize the physical contacts (rough particle are considered for instance). So models to take into account particle collisions are added.

For N particles, all these approachs approximate these interactions as a sum of pair-wise particles interactions. As the theoretical lubrication forces are only known for two particles, the interactions between the N particles is considered as the sum of the all possible interactions between all the possible pair of particles.
Even if these interactions are short range, it has been shown in [26], that in a system with many particles, making this assumption leads to unprecise computation of the forces. In particular, they show that the contribution of three particles is affected. Ignoring the multi body effects in a dense colloidal suspension can completely change the dynamics. This kind of problem has been observed in the case of the structure of some materials.

In [188], a more accurate method is proposed to compute correctly these forces for three spheres immersed in a viscous fluid. The main drawback is that the proposed approach is difficult to apply for more than three particles.

In [108], the authors propose a method to compute very accurately the lubrication forces without any approximation based on [107] . The main idea of this approach is to decompose the initial velocity flow into a singular part which contains the lubrication forces and a regular part . The regular part is solved using a fluid solver and the singular one is decomposed over the set of pair-wise close particles. To compute the singular lubrication part, the same approach as in [47] is used. Then, the velocities are corrected using the initial velocities to obtain the original problem.

Other collision strategies less respectful of the underlying physics have been developed. In [122], a minimal distance is imposed between cells to forbid overlaps, this was achieved by using a minimization procedure on a global functional of the cells position, this technique represents an efficient way to control the distance between structures. The velocities are projected into a space of admissible velocities which ensures that no contact between the structures or overlaps will occur, the velocities are thus forced to stay into this space at each time step. At each time step, distance between particles are calculated, if this distance is smaller than a certain value, the minimization procedure is activated to compute the new velocities and finally update the particles positions. The advantage of this method is that a minimal distance between the particles is maintained in a global way contrary to the approach in [121] for which the repulsive force are computed one by one for each particle. This method can deal with multiple contacts and large time step. For N particles in two dimension, the system of equations to solve consists of $2N + N((N-1)/2$ components.
In [124] a scheme for inelastic collisions is developed allowing to impose a minimal distance between the cells. It has been used to model crowd motion ([127]) for active suspensions [42], for rigid particles[50; 180]. In [125] a gluey particle model is introduced for rigid bodies immersed in a viscous, incompressible fluid. This model takes into account the viscous effects between the particles. It is based on a first order approximation of the lubrication forces and is built by considering a vanishing viscosity of the fluid.
The extension to multiple particles has been achieved in [107] and is based on the pro-

jection of the velocities onto a set of admissible velocities. In order to reduce the computational cost due to a large number of particles, the projection of the velocities is only activated for close particles.

Other methods less popular exist like in [118] where the size of the particles are changed in order to increase the size of the gap between the structures.

An other way to impose a minimal distance between particles is to add short range repulsive forces. The short range repulsive forces try to mimic the behaviour of the lubrication forces by avoiding them from entering in contacts.
In [65] artifical short range repulsive forces between spherical rigid bodies are introduced and avoid overlapping of the particles. Smooth collisions are considered meaning that at the contact points of the particles the rigid bodies velocities coincides.
In [162], a modified repulsive force is proposed and allows the particles to overlap slightly each other, the repulsive force is only activated when the particles overlap. At each grid point of the overlapping regions, the constraint of the closer particle to that point is imposed. Different parameters have to be chosen like the intensity of the force, the distance of activation or whether the value of the rebound. This is a big drawback as according to the chosen parameters, the effect of these forces can completely change the dynamics of the particles [19; 38; 46; 191]. In [43] a non-parametric model is proposed to treat collisions. Following [65], a collision model is developed in [31] for rigid bodies having more general shapes.

3.2 A collision model in the case of a level set decomposition

We present in this section the collision model proposed in [31]. This model was developed within a level set framework in the context of fluid/rigid bodies interaction. The level set decomposition is used in such a way that each body interface is captured by one level set function. The advantage of capturing the N structures with N level set functions, is that we know exactly at all points of the computational domain, the distance between each structure. Moreover, the repulsive forces can be distributed on each body interface. When two solid bodies immersed in a viscous fluid get in contact, a part of the energy is diffused into the fluid in a neighborhood of the particles (boundary layer). This model allows the conservation of the energy.
Moreover, in the case of small number of particle no parameters is required.

3.2.1 One dimensional model

These repulsive forces has been introduced by considering a one dimensional model like in [65]. Let $z(t) > 0$ defined the position of a solid body and $z = 0$ represents the location of a solid wall. The dynamical system chosen in [31] is:

$$\ddot{z} = \frac{k}{z} \exp(-\frac{z}{\varepsilon_b})$$

This dynamical model acts on a width of order ε_b at the wall $z = 0$. It is short range and singular at the contact point in order to avoid overlaps. Its potential energy is given

by:

$$E_p(z) = \int_{\frac{1}{\varepsilon_b}}^{\frac{z}{\varepsilon_b}} \frac{k}{s} \exp(-s) ds$$

The coefficient ε_b is of the same dimension of the width of the rebound. Indeed, let us take the following system :

$$\begin{cases} \ddot{z} & = \frac{k}{z} \exp(-\frac{z}{\varepsilon_b}) \\ z(0) & = 1 \\ \dot{z}(0) & = v_0 < 0 \end{cases} \tag{3.8}$$

If we denote by $z_{\min}$ the minimal location of the particle (a point at which the velocity is zero), the energy conservation gives:

$$E_p(z_{\min}) = -\frac{1}{2} v_0^2 \tag{3.9}$$

leading to:

$$z_{\min} = \varepsilon_b F^{-1}(\frac{v_0^2}{2k}), \text{ for } \varepsilon_b << 1 \tag{3.10}$$

where F is given by:

$$F(z) = \int_{z}^{\frac{1}{\varepsilon_b}} \frac{1}{s} \exp(-s) ds$$

Thus the coefficient k scales as the square of a velocity. Let $z(t)$ be immersed in a fluid of viscosity μ, $z = 0$ represents the location of the wall. Let r be the radius of the solid $z(t)$, using the first order development of the lubrication force and the fundamental principle of the dynamics we have:

$$m\ddot{z} = -6\pi\mu r^2 \frac{\dot{z(t)}}{z(t)} + mgz(t)$$

Using this model, A. Lefebvre shows in [106] that no contact occurs in finite time.

Consequently, in order to mimic the behavior of the exact lubrication force when the immersed solid body is approaching the wall, the coefficient k has to be proportional to the size of the particle and to the relative velocity.

3.2.2 Level set framework

In the level set framework, multiple level set functions are used in order to partition the entire computational domain with each level set representing a distinct structure.
Consider N bodies immersed in a fluid and denote $F_{j,i}$ the force applied by the body Ω_j to the body Ω_i and ϕ_i the level set function which captures the boundary Γ_i of the body Ω_i. The distance of a point x of Ω_i to the body Ω_j is given by $\phi_j(x)$ and the direction of the force $F_{j,i}$ is obtained directly by $\nabla\phi_j$.
Moreover, to localize the interface Γ_i one can use the delta function ζ regularized on a thickness ε on each part of the interface. The short range repulsive force is formulated as:

$$\forall x \in \Omega,\ F_{j,i}(x) = \frac{k}{\varepsilon} \rho(x) \zeta\left(\frac{\phi_i(x)}{\varepsilon}\right) \frac{\nabla\phi_j(x)}{\phi_j(x)} \exp\left(-\frac{\phi_j(x)}{\varepsilon_b}\right) \tag{3.11}$$

Consequently, we obtain the following collision model:

$$\forall x \in \Omega,\ F_{\text{global}}(x) = \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \rho(x) \frac{k}{\varepsilon} \zeta(\frac{\phi_i(x)}{\varepsilon}) \frac{\nabla \phi_j(x)}{\phi_j(x)} \exp\left(-\frac{\phi_j(x)}{\varepsilon_b}\right) \tag{3.12}$$

where k is a repulsive constant proportional to the square of the relative velocities of the corresponding bodies just before collision.

The density function ρ is given by:

$$\rho = \rho_f + \sum_{i=1}^{N} (\rho_f - \rho_i)\chi_i.$$

where χ_i is the regularized characteristic function of the body Ω_i:

$$\chi_i = 1 - H(\frac{\phi_i}{\varepsilon})$$

The coefficient ε represents the half thickness of the interface on which we apply the repulsive force. The interaction forces tend to zero out of a cut-off radius reducing the number of interacting neighbours (see Figure 3.2).
The coefficient ε_b represents the rebound coefficient. In [31], this value is fixed to ε. In fact, when the distance is equal or higher to ε_b the repulsive force is almost zero like we represent on Figure 3.2, thus increasing ε_b increase the activation distance of the force leading to a higher minimal distance between particles.

The shape of the bodies is taken into account as it is applied on the surfaces of the bodies. Thus, for two spheres Ω_i and Ω_j with radii $r_i > r_j$, the support of the regularized delta function $\zeta_\varepsilon(\phi_i)$ is higher than the one of $\zeta_\varepsilon(\phi_j)$, thus the smallest particle will be pushed with a higher intensity.

To a certain extent, this collision model try to mimic the lubrication forces as the repulsive forces are short-range, the coefficient is proportional to the relative velocities of the solid bodies and the shape of the bodies is taken into account.
This collision model accounts for all possible interactions between the N bodies. The major drawback of this model is that for N bodies captured by N level set functions, N^2 computations of the repulsive forces are required, which represents a huge computational effort.

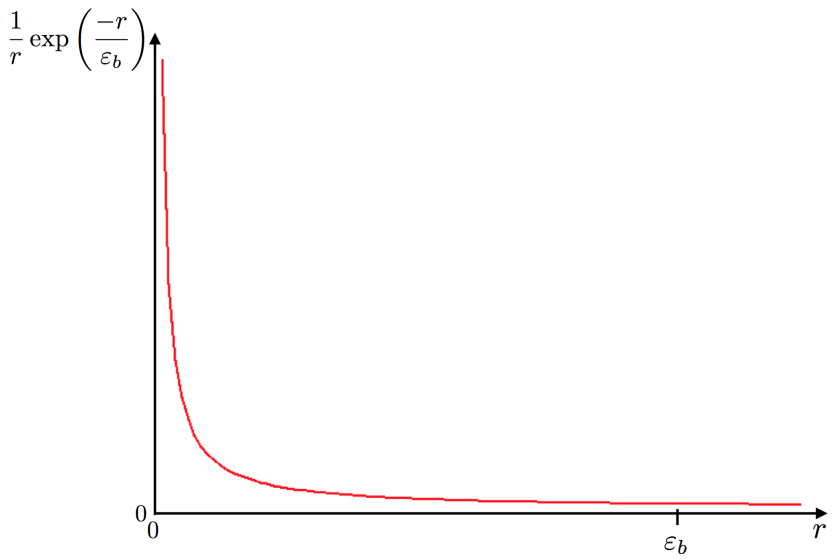


Figure 3.2: Behaviour of the repulsive force according to the distance r and the rebound ε_b.

3.3 The proposed collision model using the two distance functions

We present in this section the collision model used in this work. This collision model is an adaptation of the collision model described above. We first present the direct application of the collision model using the label maps and distance functions. Then, we describe how the repulsive coefficient is computed for several solid bodies and using the distance function φ_2. The short-range repulsive force avoid particles from merging, a minimal distance is always kept between two particles.

3.3.1 Short-range repulsive force

To reduce the high computational cost of (3.12) we propose to adapt F_{global} so that it depends only on the two distance functions φ_1 and φ_2. It comes:

$$\forall x \in \Omega,\ F_{\text{label}}(x) = \frac{k}{\varepsilon}\rho(x)\zeta_\varepsilon\left(\varphi_1(x)\right)\frac{\nabla\varphi_2(x)}{\varphi_2(x)}\exp\left(-\frac{\varphi_2(x)}{\varepsilon_b}\right) \tag{3.13}$$

The term $\zeta_\varepsilon\left(\varphi_1(x)\right)$ gives the location of the union of all interfaces. Thus, this force has its support on a subset $\Gamma_\varepsilon = \{x \in \Omega, \varphi_1(x) \leq \varepsilon\}$. Moreover, it takes into account the interaction between the closest cells at all points. Indeed, as φ_2 is the distance to the second closest object at all points of the fluid/structures domain, if a body is surrounded by other bodies the interaction of the other structures are taken into account on different part of its interface.

For instance, in Figure 3.3, two different configurations of bodies are illustrated.
In the first configuration, the repulsive force applied to the blue surrounded body comes from the four surrounding bodies as the label map L_2 has partitioned it into four parts. On the yellow part, the force applied is the one exerted by the yellow cell and so forth. On the four surrounding cell, only the forces exerted by the blue body is applied.
For the second configuration, only the green, purple and white blue particles exerted

forces on the blue cell. As the yellow particle is far, the only forces exerted on it is the one coming from the white blue particle.

The advantage of this formulation is that we get rid of the sum in (3.12) leading to a considerable saving of the computation as there is only one repulsive force for an arbitrary number of objects.

As proved in proposition 1 below, if two cells are at a distance greater than ε, the force becomes small. Moreover, numerical tests provide evidence that, in that case, this force does not change the dynamics of the bodies.

The parameter ε_b corresponds to the distance of activation of the force, numerically this

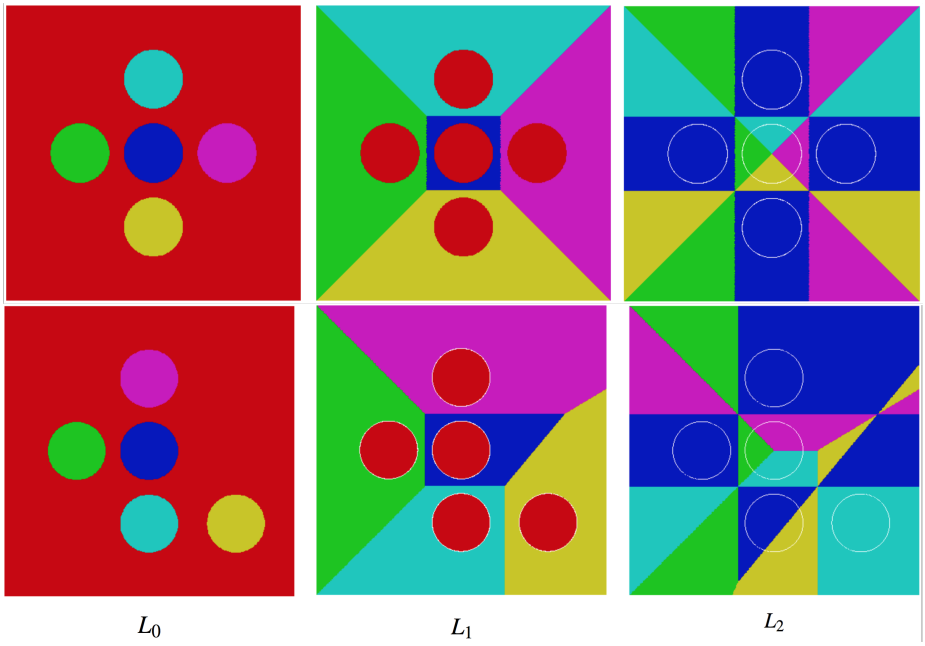


Figure 3.3: Illustration of the three label maps for two different configuration of five cells (top and bottom), from left to right: L_0, L_1 and L_2. The white contour represents the boundary of the cells.

parameter is fixed to the minimal distance between two particles. The intensity of the force is controlled by the repulsive parameter k.

3.3.2 Computation of the intensity of the repulsive force

The choice of the parameters involved in collision models is a difficult task, as according to the choice of the parameters one can obtain different dynamics of the solid bodies, in particular when one deals with dense suspensions.

Numerically, in [31] the repulsive coefficient k is chosen equal to the relative velocities just before collision, then this coefficient is fixed for the rest of the simulation. It has been applied to two immersed spherical rigid bodies falling under gravity and in the case of an ellipsoid rotating around a fixed sphere. This approach gives good results for two interacting bodies.

We want to extend this approach to the case of multiple interacting bodies, and using the label maps and distance functions. In the case of multiple interacting bodies, it is not

possible to compute the relative velocities just before collision and to keep this value until the end of the simulation. Indeed, multiple collisions can occur between two bodies and the relative velocities are different before each of them. During the numerical calculations, for instance in the case of sedimenting particles, these relative velocities get smaller and smaller. Consequently, numerical contacts and overlaps can occur. Thus, in addition of taking into account the relative velocities we have to impose a minimal coefficient when some particles are almost at rest.

The relative velocities are computed just before collision and kept until the distance between the concerned bodies becomes higher than a certain distance for which the repulsive force is negligible.
To achieve this we use the reconstituted level set functions ψ and the label map $L_{0,1}$.

The first step consists in computing the minimal value of φ_2 distance $dist_{\min}(L_{0,1}, L_2)$, on each regularized interface in order to know if the coefficient remains constant for two closest bodies or has to be computed again.
To do this, we define the regularized interface as:

$$\Gamma^{\varepsilon}_{L_{0,1}} = \left\{ y \in \Omega, |\psi_{L_{0,1}(y)}(y)| \leq \varepsilon \right\}$$

The minimal distance is directly given by:

$$\forall x \in \Omega, \mathrm{dist}_{\min}(L_{0,1}(x), L_2(x)) = \min_{x \in \Gamma^{\varepsilon}_{L_{0,1}}} (\varphi_2(x))$$

Numerically, we observe that when the particles are approaching slowly from each other, taking a coefficient proportional to the relative velocity before collision is sufficient. Otherwise, we observed that the minimal intensity has to be proportional to the square of the relative velocities.
At each time step ,the coefficient is computed as:
if $(\mathrm{dist}_{\min}(L_{0,1}, L_2) \leq D_{min})$

$$K(L_{0,1}, L_2) = \max(K_{min}, |(U_{L_2} - U_{L_{0,1}}) \cdot \nabla\varphi_2|, ((U_{L_2} - U_{L_{0,1}}) \cdot \nabla\varphi_2))^2) \tag{3.14}$$

otherwise :

$$K(L_{0,1}, L_2) = \max(|(U_{L_2} - U_{L_{0,1}}) \cdot \nabla\varphi_2|, ((U_{L_2} - U_{L_{0,1}}) \cdot \nabla\varphi_2))^2) \tag{3.15}$$

The minimal separation distance D_{min} is of the order of a mesh grid and the minimum coefficient K_{min} is fixed during numerical calculation.

Finally, the complete short range repulsive force considered for rigid bodies is:

$$\forall x \in \Omega,\ F_{\mathrm{label}}(x) = \frac{K_{L_{0,1},L_2}}{\varepsilon} \rho(x) \zeta_{\varepsilon}(\varphi_1(x)) \frac{\nabla\varphi_2(x)}{\varphi_2(x)} \exp\left(-\frac{\varphi_2(x)}{\varepsilon_b}\right) \tag{3.16}$$

In a certain extent, the proposed collision model 3.16 is an approximation of the normal component of the lubrication force provided in [100] in the case of small relative velocities and between closest solid bodies.

3.4 Error estimation between the two collision models

This section is devoted to the comparison of the two collision models introduced above: the collision model (3.12) and the short range repulsive force 3.16 and the influence of

these forces on N disks immersed in Stokes fluid flow.

We consider a bounded domain $\Omega \in \mathbb{R}^2$ which contains N disks evolving in an incompressible fluid. We denote by R the radius of each disk, by k the intensity of the two collision models and the value of the rebound parameter ε_b is fixed to ε^γ where $\gamma > 1$. In order to study the influence of the two collision models on the dynamic of N disks , we study two Stationnary Stokes models. The first one corresponds to the Stationnary Stokes equations supplemented by the collision model (3.12). The second model corresponds to the Stationnary Stokes equations with the short range repulsive force 3.16 as source term. For both models, the N gravity centers of the disks are transported by the fluid velocity field, and N level set function are reconstructed as signed distance functions. For all the estimations, we assume that:

- (H1): the distance between disks is superior or equal to ε
- (H2): $p > 2$

3.4.1 Stokes model involving the collision model (3.12)

The first problem corresponds to the steady Stokes equations with the collision model (3.12) as source term. For each disk Ω_i^1, we denote by ϕ_i^1 the level set function that captures the boundary Γ_i^1 and by X_i^1 its gravity center. At each time, the signed distance function ϕ_i^1 to the interface Γ_i^1 is defined as a signed distance function:

$$\forall x \in \Omega, \phi_i^1(x,t) = |x - X_i^1(t)| - R \tag{3.17}$$

We denote by F^1_{global} the global collision model which takes into account the repulsive forces between the N disks:

$$\forall x \in \Omega,\ F^1_{\text{global}}(x,t) = \sum_{\substack{i,j=1 \\ i\neq j}}^{N} \frac{k}{\varepsilon}\zeta\left(\frac{\phi_i^1(x,t)}{\varepsilon}\right)\frac{\nabla\phi_j^1(x,t)}{\phi_j^1(x,t)}\exp\left(-\frac{\phi_j^1(x,t)}{\varepsilon_b}\right) \tag{3.18}$$

so that denoting by U_1 the fluid velocity field, P_1 the pressure field and μ the viscosity of the fluid, for a time $T > 0$, we obtain the following problem:

$$\begin{cases} -\mu\Delta U_1 + \nabla P_1 = F^1_{\text{global}} & \text{in } \Omega\times]0,T) \\ \forall i \in \{1,..,N\}, \dfrac{d}{dt}X_i^1 = U_1(X_i^1) & \text{in } \Omega\times]0,T) \\ \nabla \cdot U_1 = 0 & \text{in } \Omega \\ U_1 = 0 & \text{on } \partial\Omega \end{cases} \tag{3.19}$$

An estimation of the velocity field is provided by the following proposition.

Proposition 1. *Considering:*

$$U_1 \in W^{2,p}(\Omega),\ P_1 \in W^{1,p}(\Omega),\ 2 < p < \infty \tag{3.20}$$

are solutions of the Stokes problem (3.19) and that the distance between the disks is

higher or equal to ε. There exists a constant $C(p,\mu,\Omega)$ such that:

$$\|U_1\|_{L^\infty(0,T,W^{2,p}(\Omega))} \leq C\frac{k}{\varepsilon^{\frac{2p-1}{p}}}(N-1)(N(2\pi R+o(\varepsilon)))^{\frac{1}{p}}\exp(-\varepsilon^{1-\gamma}) \tag{3.21}$$

where $\gamma > 1$.

Proof. Let ψ_1 denotes the level set function which captures the union of all interfaces at each time t, by definition we have:

$$\forall x\in\Omega,\ \psi_1(x,t)=\min(\phi_1^1(x,t),..,\phi_N^1(x,t))=\min(|x-X_1^1(t)|,..,|x-X_n^1(t)|)-R \tag{3.22}$$

Using the assumption on the distance between disks, since the support of the cut-off functions do not intersect, it holds:

$$\forall x\in\Omega,\quad \zeta_\varepsilon\left(\varphi_1(x,t)\right)=\sum_{j=1}^{N}\zeta_\varepsilon(\phi_j^1(x,t)),$$

Moreover, denoting by $L_i(x,t)$ the i^{th} closest object to x at time t, the model (3.12) can be formulated as:

$$\forall x\in\Omega,\quad F^1_{\text{global}}(x,t)=\sum_{i=2}^{N}\frac{k}{\varepsilon}\zeta_\varepsilon(\psi_1(x,t))\frac{\nabla\phi_{L_i^1(x,t)}(x,t)}{\phi_{L_i^1(x,t)}(x,t)}\exp\left(-\frac{\phi_{L_i^1(x,t)}(x,t)}{\varepsilon_b}\right). \tag{3.23}$$

As all the $\phi_{L_i^1}$ are distance functions we have at each time t:

$$\|\nabla\phi_{L_i^1(\cdot,t)}(\cdot,t)\|_{L^\infty(\Omega)}\leq 1$$

So that we get:

$$\|F^1_{\text{global}}(\cdot,t)\|_{L^p(\Omega)}\leq\|\frac{k}{\varepsilon}\zeta_\varepsilon(\varphi_1(\cdot,t))\|_{L^p(\Omega)}\|\sum_{i=2}^{N}\frac{1}{\phi_{L_i^1(\cdot,t)}}\exp(-\frac{\phi_{L_i^1(\cdot,t)}}{\varepsilon})\|_{L^\infty(\Omega)}.$$

Moreover, using the definition of the label maps one gets:

$$\forall x\in\Omega,\forall i\in\{3,..,N\},\quad \phi_{L_i^1(x,t)}(x,t)\geq\phi_{L_2^1(x,t)}(x,t).$$

leading to :

$$\|F^1_{\text{global}}(\cdot,t)\|_{L^p(\Omega)}\leq(N-1)\|\frac{k}{\varepsilon}\zeta_\varepsilon(\varphi_1)\|_{L^p(\Omega)}\|\frac{1}{\phi_{L_2^1}}\exp(-\frac{\phi_{L_2^1}}{\varepsilon_b})\|_{L^\infty(\Omega)}.$$

It is a classical result from level-set theory that $\frac{1}{\varepsilon}\zeta_\varepsilon(\varphi_1(x))$ is an approximation of the length of the zero level-set of φ_1. Namely, in the case of N disks, we can prove that there exists a constant $C(p)$ such that:

$$\|\frac{1}{\varepsilon}\zeta(\frac{\varphi_1}{\varepsilon})\|_{L^p(\Omega)}\leq\frac{C}{\varepsilon^{1-\frac{1}{p}}}(N(2\pi R+\mathcal{O}(\varepsilon)))^{\frac{1}{p}}, \tag{3.24}$$

so that we get:

$$\|F^1_{\text{global}}(\cdot,t)\|_{L^p(\Omega)}\leq\frac{Ck}{\varepsilon^{1-\frac{1}{p}}}(N-1)(N(2\pi R+\mathcal{O}(\varepsilon)))^{\frac{1}{p}}\left\|\frac{1}{\phi_{L_2^1(\cdot,t)}}\exp(-\frac{\phi_{L_2^1(\cdot,t)}}{\varepsilon})\right\|_{L^\infty(\Omega)}.$$

Using the assumption that $\varepsilon_b = \varepsilon^\gamma$ and that the minimal distance between the particles is at each time t higher than ε, that is :

$$\forall x \in \Omega, \forall t \in [0,T],\ \phi_{L_2^1(x,t)}(x,t) \geq \varepsilon,$$

we obtain:

$$\|F^1_{\text{global}}(\cdot,t)\|_{L^p(\Omega)} \leq (N-1)\frac{Ck}{\varepsilon^{\frac{2p-1}{p}}}(N(2\pi R + \mathcal{O}(\varepsilon)))^{\frac{1}{p}} \exp(-\varepsilon^{1-\gamma}). \tag{3.25}$$

Following [169], we have: If $U_1 \in W^{2,p}(\Omega),\ P_1 \in W^{1,p}(\Omega)$ are solution of the Stokes model (3.19) and $F_{\text{label}} \in L^p(\Omega)$, there exists a constant $c_0(p,\mu,\Omega)$ such as:

$$\|U_1(\cdot,t)\|_{W^{2,p}(\Omega)} + \|P_1(\cdot,t)\|_{W^{1,p}(\Omega)} \leq c_0 \|F^1_{\text{global}}(\cdot,t)\|_{L^p(\Omega)}$$

Taking the L^∞ norm on $[0,T]$ and using (3.25) we obtain the announced estimation. □

3.4.2 Model involving the short range repulsive force (3.16)

The second model consists in the Stokes equations supplemented by the local short-range repulsive force F^2_{label}. The evolution of each particle Ω^2_i is based on the transport of the gravity center X^2_i by the fluid velocity at this point. Each particle interface Ω^2_i is captured by a level set field ϕ^2_i. Let ψ_2 be the level set function which captures the union of all interfaces, at each time t this field is defined as:

$$\forall x \in \Omega,\ \psi_2(x,t) = \min(|x - X^2_1(t)|, .., |x - X^2_n(t)|) - R \tag{3.26}$$

and denoting by φ_2 the distance between the closest structures, the short-range repulsive force F^2_{label} is formulated as:

$$\forall x \in \Omega,\ F^2_{\text{label}}(x,t) = \frac{k}{\varepsilon}\zeta\left(\frac{\psi_2(x,t)}{\varepsilon}\right)\frac{\nabla\varphi_2(x,t)}{\varphi_2(x,t)}\exp\left(-\frac{\varphi_2(x,t)}{\varepsilon_b}\right) \tag{3.27}$$

Denoting by U_2 the fluid velocity field, P_2 the pressure field and μ the viscosity of the fluid, for a time $T > 0$, the second model is provided by:

$$\begin{cases} -\mu\Delta U_2 + \nabla P_2 = F^2_{\text{label}} & \text{in } \Omega \times [0,T) \\ \forall i \in \{1,..,N\}, \dfrac{d}{dt}X^2_i = U_2(X^2_i) & \text{in } \Omega \times [0,T) \\ \nabla \cdot U_2 = 0 & \text{in } \Omega \\ U_2 = 0 & \text{on } \partial\Omega \end{cases} \tag{3.28}$$

As for the first problem, we obtain an estimation which depends on ε, R and N:

Proposition 2. *Considering:*

$$U_2 \in W^{2,p}(\Omega),\ P_2 \in W^{1,p}(\Omega),\ 2 < p < \infty \tag{3.29}$$

are solutions of the Stokes problem (3.28) and that the distance between the disks is higher to ε. There exist a constant $C(p,\mu,\Omega)$ such that:

$$\|U_2\|_{L^\infty(0,T,W^{2,p}(\Omega))} \leq C\frac{k}{\varepsilon^{\frac{2p-1}{p}}}(N(2\pi R + O(\varepsilon)))^{\frac{1}{p}} \exp(-\varepsilon^{1-\gamma}) \tag{3.30}$$

The proof is similar to the one provided for Proposition 1.

3.4.3 Error estimation between the two Stokes models

The model of interest is the comparison of the two models above, setting $U = U_1 - U_2$ and $P = P_1 - P_2$, one gets the following system:

$$\begin{cases} -\mu\Delta U + \nabla P = F & \text{in } \Omega \\ \nabla \cdot U = 0 & \text{in } \Omega \\ \dfrac{d}{dt}(X_i^1 - X_i^2) = U_1(X_i^1) - U_2(X_i^2) & \\ U = 0 & \text{on } \Gamma \end{cases} \tag{3.31}$$

where $F = F^1_{\text{global}} - F^2_{\text{label}}$ is the difference between the two collision models. The global force F^1_{global} depends on the level set functions $\phi^1_{L_i^1}$ and ψ_1 whereas F^2_{label} depends on ψ_2 and φ_2, thus the term F can not be estimated directly.
Instead, we define a short-range repulsive force F^1_{label} associated to the N disks Ω^1_i, denoting by φ_1 the distance function $\phi^1_{L_2}$, this force is expressed as:

$$\forall x \in \Omega, \forall t \in [0,T], F^1_{\text{label}}(x,t) = \frac{k}{\varepsilon}\zeta\left(\frac{\psi_1(x,t)}{\varepsilon}\right)\frac{\nabla\varphi_1(x,t)}{\varphi_1(x,t)}\exp\left(-\frac{\varphi_1(x,t)}{\varepsilon_b}\right) \tag{3.32}$$

so that the estimation of F will involve to estimate the two following terms:

$$\|F(\cdot,t)\|_{L^p(\Omega)} \leq \|F^1_{\text{global}}(\cdot,t) - F^1_{\text{label}}(\cdot,t)\|_{L^p(\Omega)} + \|F^1_{\text{label}}(\cdot,t) - F^2_{\text{label}}(\cdot,t)\|_{L^p(\Omega)} \tag{3.33}$$

The first term corresponds to an error estimation between the two collision models for a same configuration of disks ($\forall i \in \{1,..,N\}, \Omega^1_i = \Omega^2_i$). The second term provides an estimation of the short-range repulsive force F_{label}, the disks dot not have necessarily the same configuration.

In fact, we can demonstrate the following proposition:

Proposition 1. *Considering:*

$$U \in W^{2,p}(\Omega),\ P \in W^{1,p}(\Omega),\ 2 < p < \infty \tag{3.34}$$

are solutions of the Stokes problem (3.31) and that propositions 1,2 hold. There exist a constant $C(p,\mu,\Omega)$ such that:

$$\|U\|_{W^{2,p}(\Omega))} \leq |I_1 + I_2|_\infty \left(1 + \frac{|\nabla G|_\infty}{R+\varepsilon}\left(\frac{\exp((|F^1_{global}| + \frac{|\nabla G|_\infty}{R+\varepsilon})t) - 1}{|F^1_{global}| + \frac{|\nabla G|_\infty}{R+\varepsilon}}\right)\right) \tag{3.35}$$

where:

$$\frac{|\nabla G|_\infty}{R+\varepsilon} \leq \frac{Ck}{(R+\varepsilon)\varepsilon}\exp(-\frac{1}{\varepsilon^{\gamma-1}})$$

$$|F^1_{global}| \leq C\frac{k}{\varepsilon^{\frac{2p-1}{p}}}(N-1)(N(2\pi R + o(\varepsilon)))^{\frac{1}{p}}\exp(-\varepsilon^{1-\gamma})$$

$$\|F^1_{global} - F^1_{label}\|_{L^p(\Omega)} \leq \frac{k(N-2)}{\alpha R \varepsilon^{\frac{p-1}{p}}} \left(N(2\pi R + \mathcal{O}(\varepsilon))\right)^{\frac{1}{p}} \exp\left(-\frac{\alpha R}{\varepsilon^{\gamma}}\right)$$

$$|I_1 + I_2|_{\infty} \leq \frac{Ck}{(R+\varepsilon)\varepsilon} \exp(-\frac{1}{\varepsilon^{\gamma-1}})(\|\varphi_2\|_{L^p(\Omega)} + RC)) + \|F^1_{global} - F^1_{label}\|_{L^p(\Omega)}$$

In what follows, we provide a proof of this proposition.

3.4.4 Estimation of the term $\|F^1_{\text{global}} - F^1_{\text{label}}\|_{L^p(\Omega)}$

The following result provides an estimate of the difference introduced by using the label force (3.16) instead of the collision model (3.12) for the same configuration. As expected, the difference is smaller for large bodies and/or small ε. Moreover, it depends on the intensity of the force.

Proposition 3. *Assuming that the N bodies are disks, such as:*

$$\forall x \in \Omega, \forall t \in [0,T], \quad \varphi_2(x,t) \geq \varepsilon$$

then:

$$\|F^1_{global} - F^1_{label}\|_{L^p(\Omega)} \leq \frac{k(N-2)}{\alpha R \varepsilon^{\frac{p-1}{p}}} \left(N(2\pi R + \mathcal{O}(\varepsilon))\right)^{\frac{1}{p}} \exp\left(-\frac{\alpha R}{\varepsilon^{\gamma}}\right)$$

where $\alpha = \frac{\sqrt{13}-3}{2} \approx 0.3$.

Proof. We have:

$$F^1_{\text{global}} = \sum_{i=3}^{N} \frac{k}{\varepsilon} \zeta_\varepsilon(\psi_1) \frac{\nabla \phi_{L_i}}{\phi_{L_i}} \exp(-\frac{\phi_{L_i}}{\varepsilon_b}). \tag{3.36}$$

As $\forall i \in \{2,..,N\}, \phi_{L_i}$ are distance functions we have :

$$|\nabla \phi_{L_i}|_{L^\infty_\Omega} \leq 1$$

Moreover, the definition of the label maps gives:

$$\forall x \in \Omega, \forall i \in \{4,..,N\}, \quad \phi_{L_i(x)}(x) \geq \phi_{L_3(x)}(x).$$

and we have seen that there exists a constant $C(p)$

$$\|\frac{1}{\varepsilon}\zeta(\frac{\varphi_1}{\varepsilon})\|_{L^p(\Omega)} \leq \frac{C}{\varepsilon^{\frac{p-1}{p}}} (N(2\pi R + \mathcal{O}(\varepsilon)))^{\frac{1}{p}},$$

leading to :

$$\|F^1_{\text{global}} - F^1_{\text{label}}\|_{L^p(\Omega)} \leq \frac{kC}{\varepsilon^{\frac{p-1}{p}}} (N-2)(N(2\pi R + \mathcal{O}(\varepsilon)))^{\frac{1}{p}} \|\frac{1}{\phi_{L_3}} \exp(-\frac{\phi_{L_3}}{\varepsilon_b})\|_{L^\infty(\Omega)}.$$

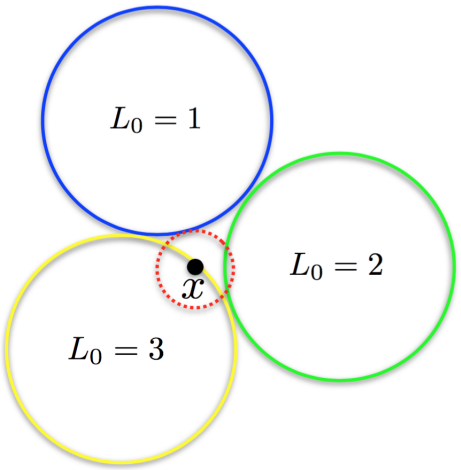


Figure 3.4: Configuration of three bodies. The circles represented are of radius $R + \varepsilon$.

We are looking for a lower bound of ϕ_{L_3} depending on the radius R. To that aim, we consider the worst case of three disks which are enlarged by ε and are in contact like shown in Figure 3.4. This configuration gives the minimum distance required between two disks. Taking a point x on the boundary of one disk (without loss of generality, Γ_1) as shown in Figure 3.4, we compute the minimal distance of x to the second closest disk. Once again, the worst case is obtained in the situation where x is as in Figure 3.4. Through a simple calculation, we obtain as distance:

$$d = \frac{\sqrt{13} - 3}{2} R$$

that is a lower bound for ϕ_{L_3}. As $r \to \frac{1}{r}\exp(-\frac{r}{\varepsilon})$ is decreasing, we obtained the announced estimate. □

Let us point out to the reader that this difference tends to zero when R tends to ∞ or when ε tends to zero. Consequently, we can adjust ε depending upon R such as this difference becomes negligible. Numerically, we take an ε that depends on the discretization space step, thus the grid mesh size is selected in order to lower the difference.
Moreover, the influence of the repulsive force imposed by the first closest cell at all points F^1_{label}, is the most influential on the dynamics of the cells as it is the largest. Therefore, numerically it is better to compare the relative difference:

$$\frac{\|F^1_{\text{global}} - F^1_{\text{label}}\|_{L^p(\Omega)}}{\|F_{\text{label}}\|_{L^p(\Omega)}}.$$

3.4.5 Estimation of $\|F^1_{\text{label}} - F^2_{\text{label}}\|_{L^p(\Omega)}$

The repulsive force F^1_{label} depends on ψ_1, φ_1 and $\nabla\varphi_1$ whereas F^2_{label} is associated to the second problem and thus depends on the level set function ψ_2, the distance function φ_2 and its gradient $\nabla\varphi_2$. We define two spaces $L^p_\varepsilon(\Omega))$ and $W^{1,p}_\varepsilon(\Omega)$ as:

$$L^p_\varepsilon(\Omega) = \{f \in L^p(\Omega), \forall x \in \Omega, f(x) \geq \varepsilon\}$$

and :

$$W^{1,p}_{\varepsilon}(\Omega) = \{f \in L^p_{\varepsilon}(\Omega), \nabla f \in L^p(\Omega), \|\nabla f\|_{L^p} \leq 1\}$$

and the functions :

$$\begin{cases} \mathcal{G} : (W^{1,p}(\Omega), W^{1,p}_{\varepsilon}(\Omega), L^p_{\varepsilon}(\Omega)) & \longrightarrow L^p(\Omega) \\ (\psi, \ \varphi, \ \nabla\varphi) & \longmapsto G(\psi, \varphi, \nabla\varphi) = \dfrac{k}{\varepsilon}\zeta(\dfrac{\psi}{\varepsilon})\dfrac{\nabla\varphi}{\varphi}\exp(-\dfrac{\varphi}{\varepsilon_b}) \end{cases} \tag{3.37}$$

So that using this function, the term $\|F^1_{\text{label}} - F^2_{\text{label}}\|_{L^p(\Omega)}$ can be valued as:

$$\|F^1_{\text{label}} - F^2_{\text{label}}\|_{L^p(\Omega)} \leq \|\nabla G\|_{L^\infty(\Omega)} |\|\psi_1 - \psi_2\|_{L^p(\Omega)}, \|\varphi_1 - \varphi_2\|_{L^p(\Omega)}, \|\nabla\varphi_1 - \nabla\varphi_2\|_{L^p(\Omega)}|_{L^\infty} \tag{3.38}$$

We provide in what follows an estimation of $|\|\psi_1 - \psi_2\|_{L^p(\Omega)}, \ \|\varphi_1 - \varphi_2\|_{L^p(\Omega)}, \|\nabla\varphi_1 - \nabla\varphi_2\|_{L^p(\Omega)}|_{L^\infty}$ and then $\|\nabla G\|_{L^\infty(\Omega)}$

Estimation of $\|\psi_1 - \psi_2\|_{L^p(\Omega)}$, $\|\varphi_1 - \varphi_2\|_{L^p(\Omega)}$,$\|\nabla\varphi_1 - \nabla\varphi_2\|_{L^p(\Omega)}$

For each time t, there exists a constant $C(p, \Omega)$ such as:

$$\begin{cases} \|\psi_1(\cdot,t) - \psi_2(\cdot,t)\|_{L^p(\Omega)} & \leq C\max\limits_i |X^1_i(t) - X^2_i(t)| \\ \|\varphi_1(\cdot,t) - \varphi_2(\cdot,t)\|_{L^p(\Omega)} & \leq C\max\limits_i |X^1_i(t) - X^2_i(t)| \\ \|\nabla\varphi_1(\cdot,t) - \nabla\varphi_2(\cdot,t)\|_{L^p(\Omega)} & \leq \dfrac{2}{R+\varepsilon}\Big(2(\|\varphi_2(\cdot,t)\|_{L^p(\Omega)} + CR) + \max\limits_i |X^1_i(t) - X^2_i(t)|)\Big) \end{cases} \tag{3.39}$$

• The level set functions which captures the union of all interfaces are defined by:

$$\forall x \in \Omega, \forall t \in [0,T], \ \psi_1(x,t) = \min_i(|x - X^1_i(t)|) - R, \ \psi_2(x,t) = \min_i(|x - X^2_i(t)|) - R$$

For a point $x \in \Omega$ and a time t, let $\psi_1(x,t)$ and $\psi_2(x,t)$ be given by:

$$\psi_1(x,t) = |x - X^1_{i_1}(t)| - R, \ \ \psi_2(x,t) = |x - X^2_{i_2}(t)| - R \tag{3.40}$$

this means that:

$$\forall i \in \{1, \cdot\cdot, N\}, |x - X^1_i| \geq |x - X^1_{i_1}(t)|, \ |x - X^2_i| \geq |x - X^2_{i_2}(t)|$$

if $i_1 = i_2$, the inequality is immediate, otherwise we have:

$$\forall i, |X^1_i - X^2_i| \geq |X^1_i - x| - |x - X^2_i| \geq (\psi_1(x,t) + R) - |x - X^2_i|$$

in particular setting $i = i_2$ we obtain:

$$\max_i |X^1_i - X^2_i| \geq \max |X^1_{i_2} - X^2_{i_2}| \geq \psi_1(x,t) - \psi_2(x,t)$$

By symmetry and taking the L^p norm of $\varphi_1 - \varphi_2$, we obtain :

$$\|\psi_1(\cdot,t) - \psi_2(\cdot,t)\|_{L^p(\Omega)} \leq |\Omega|^{\frac{1}{p}} \max_i |X^1_i(t) - X^2_i(t)|$$

• The second term depends on the distance functions whichs are associated to the label maps $L^1_2(x,t)$ and $L^2_2(x,t)$, mainly:

$$\varphi_1(x,t) = |x - X^1_{L^1_2(x,t)}(t)| - R, \ \ \varphi_2(x,t) = |x - X^2_{L^2_2(x,t)}(t)| - R \tag{3.41}$$

The level set functions $\psi_1(x,t)$ and $\psi_2(x,t)$ are signed distance functions to the closest interfaces, thus by definition we have:

$$\varphi_1(x,t) \geq \psi_1(x,t), \quad \varphi_2(x,t) \geq \psi_2(x,t)$$

in particular, setting:

$$\varphi_1(x,t) = |x - X^1_{j_1}(t)| - R, \quad \varphi_2(x,t) = |x - X^2_{j_2}(t)| - R \tag{3.42}$$

we have that $i_1 \neq j_1$, $i_2 \neq j_2$ and:

$$\forall i \neq i_1, |x - X^1_i| \geq |x - X^1_{j_1}|, \quad \text{and } \forall i \neq i_2, |x - X^2_i| \geq |x - X^2_{j_2}|$$

If $L^1_1(x) = L^2_1(x)$ or $L^1_0(x) = L^2_0(x)$ this means that i_1 and i_2 are equal and we get:

$$\forall i \neq i_1, |X^1_i - X^2_i| \geq |X^1_i - x| - |X^2_i - x| \geq \varphi_1(x,t) + R - |X^2_i - x|$$

and as $i_2 \neq j_2$, we have:

$$\max_i |X^1_i(t) - X^2_i(t)| \geq |X^1_{j_2} - X^2_{j_2}| \geq \varphi_1(x,t) - \varphi_2(x,t)$$

by symmetry one gets the right estimation.

Otherwise if $i_1 \neq i_2$ we have:

$$\forall i \neq i_2, |X^1_i - X^2_i| \geq |X^2_i - x| - |X^1_i - x| \geq \varphi_2(x,t) + R - |X^1_i - x|$$

Taking $i = i_1$, we get:

$$\max_i |X^1_i(t) - X^2_i(t)| \geq |X^1_{i_1} - X^2_{i_1}| \geq \varphi_2(x,t) - \psi_1(x,t)$$

and as $\psi_1 - \varphi_1 \leq 0$ one gets:

$$\varphi_2 - \varphi_1 \leq \varphi_2 - \psi_1 - (\psi_1 - \varphi_1) \leq \max_i |X^1_i(t) - X^2_i(t)| \tag{3.43}$$

The same reasoning provides:

$$\varphi_1 - \psi_2 \leq \max_i |X^1_i(t) - X^2_i(t)| \tag{3.44}$$

so that taking the L^p norm, one finally obtains:

$$\|\varphi_1(\cdot,t) - \varphi_2(\cdot,t)\|_{L^p(\Omega)} \leq |\Omega|^{\frac{1}{p}} \max_i |X^1_i(t) - X^2_i(t)|$$

•The last term involves the gradient of the distance functions φ_1 and φ_2, for a point x and a time t where these functions are defined as 3.42, the difference of the gradients are provided by:

$$|\nabla\varphi_1(x,t) - \nabla\varphi_2(x,t)| = \frac{x - X_{j_1^1}}{|x - X_{j_1^1}|} - \frac{x - X_{j_2^2}}{|x - X_{j_2^2}|}$$

and as ε is a lower-bound of φ_1 and φ_2 , we have $|x - X_{j_1^1}| \geq R + \varepsilon$ and $|x - X_{j_2^2}| \geq R + \varepsilon$ so that defining a function h as:

$$\forall x \geq R + \varepsilon, h(x) = \frac{x}{|x|}$$

we get:

$$|\nabla\varphi_1 - \nabla\varphi_2| \leq |\nabla h|_\infty |X_{j_1^1} - X_{j_2^2}| \leq \frac{2}{R+\varepsilon}|X_{j_1^1} - X_{j_2^2}|$$

Using the upper-bound of $\varphi_1 - \varphi_2$, we finally obtain:

$$|\nabla\varphi_1 - \nabla\varphi_2| \leq \frac{2}{R+\varepsilon}(2(\varphi_2 + R) + \max_i |X_i^1(t) - X_i^2(t)|) \tag{3.45}$$

As $\varphi_2 \in L^p(\Omega)$ one can take the L^p norm to obtain the announced estimation. Finally, we obtain:

$$\begin{aligned} |\|\psi_1 - \psi_2\|_{L^p(\Omega)}, \|\varphi_1 - \varphi_2\|_{L^p(\Omega)}, \|\nabla\varphi_1 - \nabla\varphi_2\|_{L^p(\Omega)}|_\infty \\ \leq \frac{2}{R+\varepsilon}(2(\varphi_2 + R) + \max_i |X_i^1(t) - X_i^2(t)|) \end{aligned} \tag{3.46}$$

Estimation of $|\nabla G|_{L^\infty}$.

The function G is defined as:

$$\begin{cases} G : (\mathbb{R}, \mathbb{R} \setminus B(0,\varepsilon), B(0,1)) & \longrightarrow \mathbb{R}^2 \\ (x,y,z) & \longmapsto \frac{k}{\varepsilon}\zeta(\frac{x}{\varepsilon})\frac{z}{y}\exp(-\frac{y}{\varepsilon_b}) \end{cases} \tag{3.47}$$

The first partial derivative of this function is $\dfrac{\partial G}{\partial x}$ is :

$$|\frac{\partial G}{\partial x}|_{L^\infty} = |\frac{k}{\varepsilon^2}\zeta'(\frac{x}{\varepsilon})\frac{z}{y}\exp(\frac{-y}{\varepsilon_b})| \leq |\frac{k}{\varepsilon}\zeta'(\frac{x}{\varepsilon})|_{L^\infty}|z|_{L^\infty}|\frac{\exp(\frac{-y}{\varepsilon_b})}{y}|_{L^\infty}$$

As $z \in B(0,1)$, $y \geq \varepsilon$ and the cut off function ζ_ε is bounded, using $\varepsilon_b = \varepsilon^\gamma$ we obtain:

$$|\frac{\partial G}{\partial x}|_{L^\infty} \leq C\frac{k}{\varepsilon^2}\exp(-\frac{1}{\varepsilon^{\gamma-1}}) \tag{3.48}$$

Similarly, we have:

$$|\frac{\partial G}{\partial y}|_{L^\infty} = |\frac{k}{\varepsilon}\zeta(\frac{x}{\varepsilon})z(\frac{1}{y^2} + \frac{1}{\varepsilon_b})\exp(\frac{-y}{\varepsilon_b})| \leq C(\frac{k}{\varepsilon^2} + \frac{k}{\varepsilon^\gamma})\exp(-\frac{1}{\varepsilon^{\gamma-1}}) \tag{3.49}$$

and the last partial derivative is given by:

$$|\frac{\partial G}{\partial z}|_{L^\infty} = |\frac{k}{\varepsilon}\zeta(\frac{x}{\varepsilon})\frac{z}{y}\exp(\frac{-y}{\varepsilon_b})| \leq kC\frac{\exp(-\frac{1}{\varepsilon^{\gamma-1}})}{\varepsilon} \tag{3.50}$$

Thus, there exists a constant C such as:

$$|\nabla G|_{L^\infty} \leq \frac{Ck}{\varepsilon}\exp(-\frac{1}{\varepsilon^{\gamma-1}}) \tag{3.51}$$

Final estimation

The transport equation of the gravity centers gives:

$$\frac{d}{dt}|X_i^1 - X_i^2| \leq |U_1(X_i^1) - U_2(X_i^2)| \leq |U_1(X_i^1) - U_1(X_i^2)| + |U_1(X_i^2) - U_2(X_i^2)|$$

As $U_1 \in W^{2,p}(\Omega)$ and $U \in W^{2,p}(\Omega)$, using the injection $W^{2,p}(\Omega) \hookrightarrow L^\infty(\Omega)$:

$$\frac{d}{dt}|X_i^1 - X_i^2| \leq \|\nabla U_1\|_{L^\infty}|X_i^1 - X_i^2| + |U|_{L^\infty(\Omega)} \tag{3.52}$$

Moreover as $W^{2,p}(\Omega) \hookrightarrow L^\infty(\Omega)$, we have:

$$|U|_{L^\infty(\Omega)} \leq \|U\|_{W^{2,p}(\Omega)} \leq \|F^1_{\text{global}} - F^2_{\text{label}}\|_{L^p(\Omega)}$$

Using proposition 1 and the injection $W^{2,p}(\Omega) \hookrightarrow W^{1,\infty}(\Omega)$ we get:

$$\|\nabla U_1\|_{L^\infty([0,T],L^\infty(\Omega))} \leq |F^1_{\text{global}}|_{L^\infty(0,T,L^p(\Omega))}$$

leading to:

$$\frac{d}{dt}|X_i^1 - X_i^2| \leq |F^1_{\text{global}}|_{L^\infty(0,T,L^p(\Omega))}|X_i^1 - X_i^2| + \|F^1_{\text{global}} - F^2_{\text{label}}\|_{L^p(\Omega)} \tag{3.53}$$

Using a Gronwall inequality and as $X_i^1(0) - X_i^2(0) = 0$ we get:

$$|X_i^1(t) - X_i^2(t)| \leq \int_0^t \|F^1_{\text{global}} - F^2_{\text{label}}\|_{L^p}(s) \exp(|F^1_{\text{global}}|_{L^\infty,L^p}(t-s))ds \tag{3.54}$$

in particular:

$$\max_i|X_i^1(t) - X_i^2(t)| \leq \int_0^t \|F^1_{\text{global}}(\cdot,s) - F^2_{\text{label}}(\cdot,s)\|_{L^p} \exp\left(|F^1_{\text{global}}|_{L^\infty_T,L^p}(t-s)\right) \tag{3.55}$$

and as:

$$\|F^1_{\text{global}} - F^2_{\text{label}}\|_{L^p} \leq \frac{|\nabla G|_\infty}{R+\varepsilon}\max_i|X_i^1(t) - X_i^2(t)| + \frac{|\nabla G|_\infty}{R+\varepsilon}(\|\varphi_2\|_{L^p} + RC) + \|F^1_{\text{global}} - F^1_{\text{label}}\|_{L^p}$$

denoting by $I_1(t)$ and $I_2(t)$ the terms:

$$I_1(t) = \frac{|\nabla G|_\infty}{R+\varepsilon}(\|\varphi_2(\cdot,t)\|_{L^p} + RC), \quad , I_2 = \|F^1_{\text{global}} - (\cdot,t)F^1_{\text{label}}(\cdot,t)\|_{L^p}$$

and using the equation (3.55) we obtain:

$$\|F^1_{\text{global}} - F^2_{\text{label}}\|_{L^p} \leq \frac{|\nabla G|_\infty}{R+\varepsilon}\int_0^t \|F^1_{\text{global}}(\cdot,s) - F^2_{\text{label}}(\cdot,s)\|_{L^p} \exp(|F^1_{\text{global}}|_{L^\infty_T,L^p}(t-s))ds + I_1 + I_2$$

Using a second Gronwall inequality, we obtain:

$$\|F^1_{\text{global}} - F^2_{\text{label}}\|_{L^p} \leq I_1(t) + I_2(t)$$
$$+ \frac{|\nabla G|_\infty}{R+\varepsilon}\exp(|F^1_{\text{global}}|t)\int_0^t (I_1(s) + I_2(s))\exp(-|F^1_{\text{global}}|s)\exp(\frac{|\nabla G|_\infty}{R+\varepsilon}(t-s))ds \tag{3.56}$$

so that taking the L^∞ norm on $(0,T)$ we finally obtain:

$$\|F^1_{\text{global}} - F^2_{\text{label}}\|_{L^p} \leq |I_1 + I_2|_\infty\left(1 + \frac{|\nabla G|_\infty}{R+\varepsilon}\left(\frac{\exp((|F^1_{\text{global}}| + \frac{|\nabla G|_\infty}{R+\varepsilon})t) - 1}{|F^1_{\text{global}}| + \frac{|\nabla G|_\infty}{R+\varepsilon}}\right)\right)$$

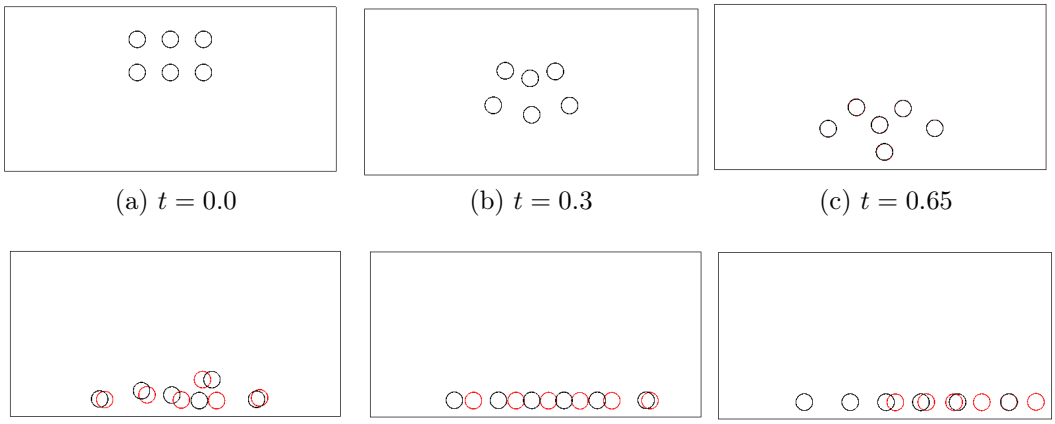
3.5 Numerical validation of the error estimation

This section is devoted to the validation through numerical simulation of the error estimation between the two stokes models. These models are supplemented by a gravity term ρg where ρ takes into account the density of the fluid and of the disks. We fixed the density of the fluid to one whereas the density of the disks is set to 2. The two stationnary Stokes models are discretized by a finite difference method on a staggered grid where the pressure and the level set function are located at the center of the mesh cells and the velocity at the center of the sides.
The steady stokes equations are solved by means of a projection method of Chorin type and a fixed point method. To solve the poisson equation on the intermediate velocity, the solver FISHPACK is used.
In order to advect the gravity centers, the velocities are interpolated from the Cartesian grid points to the gravity centers using the M_4' interpolation kernel introduced in [132] :

$$M_4'(x) = \begin{cases} 0 & \text{if } |x| > 2 \\ \frac{1}{2}(2 - |x|^2)(1 - |x|) & \text{if } 1 \leq |x| \leq 2 \\ 1 - \frac{5}{2x^2} + \frac{3|x|^3}{2} & \text{if } |x| \leq 1 \end{cases}$$

The support of this kernel is of size 4, it is locally third order and globally second-order. Owing to its regularity, accuracy and efficiency, this kernel has been widely used, in particular for the simulation of vortex flows (see for instance [61]).
The time stepping scheme used to advect the gravity centers is a two-order Runge Kutta scheme. We consider a rectangular computational domain $\Omega = [0, 2] \times [0, 1]$. The simulations are performed on a grid of size (512×256). The coefficient γ is set to 1×1.10^{-10}, the viscosity $\mu = 0.1$.

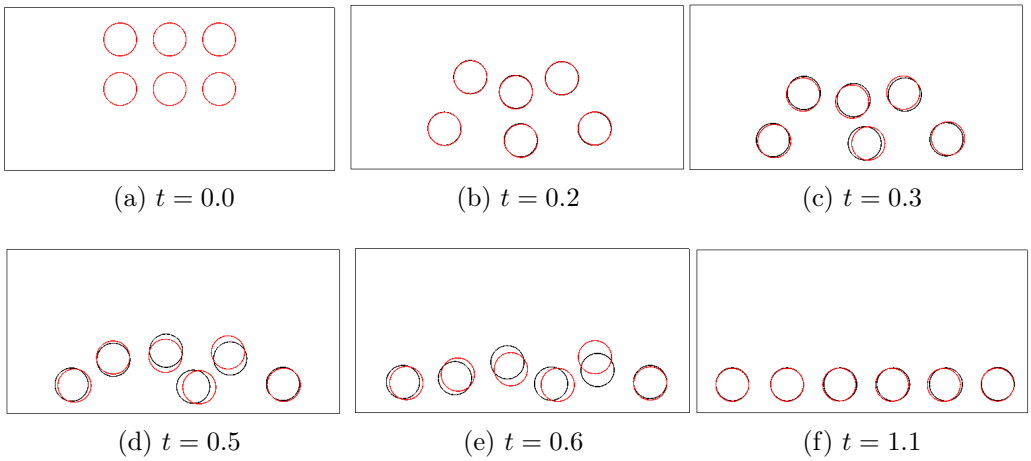
Three test cases are considered: the simulation of 6 disks of radius $R = 0.1$ and $R = 0.05$ and the sedimentation of 14 of radius $R = 0.05$. By comparing the numerical simulations obtained on Figure 3.5 and 3.6, we observe that for the disks of radius $R = 0.1$, the dynamics of the disks are similar for both stokes models. For the greatest particles, we can see that from time $t = 0.5$ to $t = 0.6$, we see a slight difference between the particles Ω_i^1 and Ω_i^2, then the particles reach a steady state and are located exactly at the same places (for all $i \in \{1, ..6\}$, $\Omega_i^1 = \Omega_i^2$). On the contrary, after collision the configuration of the smallest particles is completely different for the two Stokes models, thereafter the relative difference increases, and the obtained stationnary state is different for both set of particles. Contrary to what we expected, the estimation $\|F^1_{\text{global}} - F^1_{\text{label}}\|_{L^\infty((0,10),L^p(\Omega)}$ is smaller in the case of particles of radius $R = 0.05$ than for the case $R = 0.1$, this may be due to the fact that the particles for $R = 0.05$ stay far enough from each other.
The last simulation is shown on Figure 3.7, starting from $t = 1.5$ the particles take different trajectories until the end of the simulation, the obtained estimation $\|F^1_{\text{global}} - F^1_{\text{label}}\|_{L^\infty((0,10),L^p(\Omega)}$ is around twice higher than the one obtained when $N = 6$ which confirms the dependence of the estimation on the number of particles.



(a) $t = 0.0$ (b) $t = 0.3$ (c) $t = 0.65$

(d) $t = 1.2$ (e) $t = 2.25$ (f) $t = 3.0$

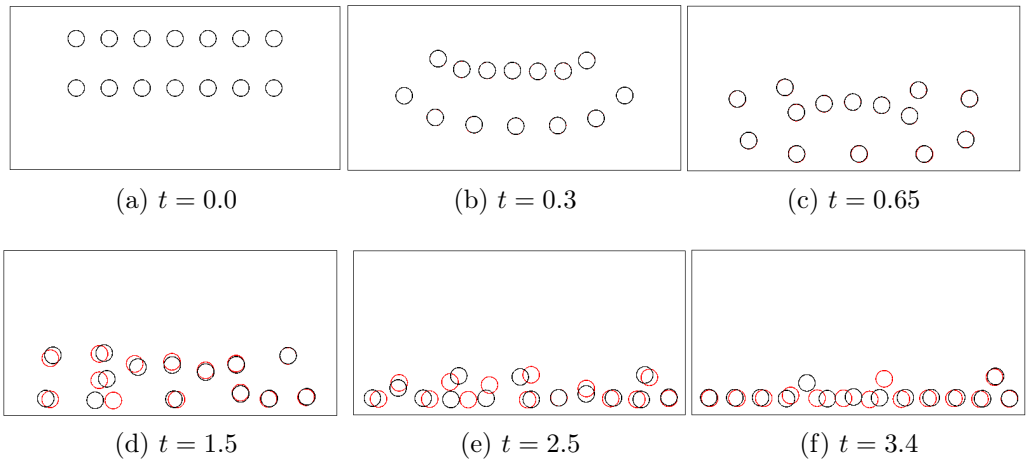
Figure 3.5: Sedimentation of 6 disks of radius $R = 0.05$, black lines correspond to the isoline $\psi_1 = 0$ and the red lines corresponds to the isoline $\psi_2 = 0$ associated to the Stokes model 2. We obtain the error estimation for the same configuration: $\|F^1_{\text{global}} - F^1_{\text{label}}\|_{L^\infty((0,10),L^p(\Omega)} = 3.48.10^{-4}$.



(a) $t = 0.0$ (b) $t = 0.2$ (c) $t = 0.3$

(d) $t = 0.5$ (e) $t = 0.6$ (f) $t = 1.1$

Figure 3.6: Sedimentation of 6 disks of radius $R = 0.1$, black lines correspond to the isoline $\psi_1 = 0$ and the red lines corresponds to the isoline $\psi_2 = 0$ associated to the Stokes model 2. We obtain the error estimation for the same configuration: $\|F^1_{\text{global}} - F^1_{\text{label}}\|_{L^\infty((0,10),L^p(\Omega)} = 2.22.10^{-3}$.



(a) $t = 0.0$ (b) $t = 0.3$ (c) $t = 0.65$

(d) $t = 1.5$ (e) $t = 2.5$ (f) $t = 3.4$

Figure 3.7: Sedimentation of 14 disks of radius $R = 0.04$, black lines correspond to the isoline $\psi_1 = 0$ and the red lines corresponds to the isoline $\psi_2 = 0$ associated to the Stokes model 2. We obtain the error estimation for the same configuration: $\|F^1_{\text{global}} - F^1_{\text{label}}\|_{L^\infty((0,10),L^p(\Omega)} = 7.54.10^{-4}$.

Conclusion

In this chapter, we have presented the numerical strategy adopted to handle contacts between structures.
The proposed collision model consists in a short-range repulsive force which depends on the two distance functions φ_1 and φ_2, it is an adaptation of the collision model introduced in [31] for the case of a larger number of particles. The collision model proposed in [31] takes into account the interaction between all the particles, whereas our model is only active between the closest particles.
The intensity of the force is either controlled by a constant coefficient which depends on the external forces exerted on the particles or is directly provided by the relative velocities of the closest particles thanks to the label maps L_2.
The short-range repulsive force with a constant intensity is compared to the collision model [31] in the case of N disks. The estimation error depends on the size and the number of disks. The impact of these forces on the solution of two Stokes problems is also investigated.

Chapter 4

Evolution of the label maps and distance functions

Contents

This chapter is devoted to the evolution of the three label maps and the two associated distance functions. We present the evolution algorithm in two-dimensions, the extension to the 3D case is straightforward.

The evolution is based on the transport of one level set function that captures the union of all interfaces, then a multi label fast marching method is performed enabling a re-initialization of the label maps and distance functions. For our applications, the informations of interest are:

- The location of each structure given by L_0 inside and L_1 outside close to the interface and the distance φ_1 which captures the union of the interfaces.
- The location of the first closest structure L_2 and the distance φ_2 when the particles are nearby to avoid numerical contacts.

The proposed collision model is a short-range repulsive force, it is thus negligible when the particles are far enough from each other. Consequently, we do not need the label and distance functions in the all computational domain. Taking this into account, we only evolved L_1 and φ_1 in a thin band close to the interfaces, we denote by NB_{L_1} the size of this narrow-band from each side of the interface. The computation of L_2 and φ_2 is activated only if these thin bands met (just before collision) on a distance NB_{L_2}. The final step consists in updating the level set function ϕ as a signed distance function using the new updated distance function φ_1 where φ is less than the distance NB_{L_1}

4.1 Evolution of the label maps L_0 and L_1

We describe the evolution of L_0 and L_1 which is based on the transport of a level set function ϕ which captures the union of all interfaces. We assume that the transport

equation on the level set function is discretized in time by an explicit scheme which involves that the displacement of a particle from one time step to another is less or equal to one space grid.

4.1.1 Global level set function

Since the distance function φ_1 and the label L_1 are only of interest in a thin band close to the interfaces, the level set function ϕ which captures the union of all interfaces is defined as:

$$\phi(x) = \begin{cases} -C & x \in \cup_{i=1}^{N} \Omega_i,\ d(x, \cup_{i=1}^{N} \Gamma_i) > NB_{L_1} \\ -d(x, \cup_{i=1}^{N} \Gamma_i) & x \in \cup_{i=1}^{N} \Omega_i,\ d(x, \cup_{i=1}^{N} \Gamma_i) \leq NB_{L_1} \\ d(x, \cup_{i=1}^{N} \Gamma_i) & x \notin \cup_{i=1}^{N} \Omega_i,\ d(x, \cup_{i=1}^{N} \Gamma_i) \leq NB_{L_1} \\ C & x \notin \cup_{i=1}^{N} \Omega_i,\ d(x, \cup_{i=1}^{N} \Gamma_i) > NB_{L_1} \end{cases} \tag{4.1}$$

where $C >> 1$. This level set field is a signed distance function close to the interfaces, it is negative inside each structure and positive outside. In the context of fluid/structure interaction, the level set function can be transported with the fluid velocity. Denoting by U the fluid velocity in the entire computational domain Ω. The level set function is defined as a signed distance function solution of the following scalar transport equation:

$$\begin{cases} \partial_t \phi + U \cdot \nabla \phi = 0 \\ \phi(x, 0) = \phi_0(x) \end{cases} \tag{4.2}$$

in a neighboorhood of the interfaces. To transport accurately the interface using a WENO5 scheme, the narrow-band size NB_{L_1} is fixed to a value larger than six mesh grid. At each time step and after the multi-label fast marching procedure, the level set function ϕ is updated as a signed distance function as:

$$\phi(x) = \begin{cases} -C & x \in \cup_{i=1}^{N} \Omega_i,\ d(x, \cup_{i=1}^{N} \Gamma_i) > NB_{L_1} \\ -\varphi_1(x) & x \in \cup_{i=1}^{N} \Omega_i,\ d(x, \cup_{i=1}^{N} \Gamma_i) \leq NB_{L_1} \\ \varphi_1(x) & x \notin \cup_{i=1}^{N} \Omega_i,\ d(x, \cup_{i=1}^{N} \Gamma_i) \leq NB_{L_1} \\ C & x \notin \cup_{i=1}^{N} \Omega_i,\ d(x, \cup_{i=1}^{N} \Gamma_i) > NB_{L_1} \end{cases} \tag{4.3}$$

4.1.2 Evolution of the label maps L_0

The evolution of the label map L_0 depends on the sign of the level set function. By definition the region where the level set function is non negative corresponds to the fluid providing at each time the label function L_0 in the fluid region, that is:

$$\forall x \in \Omega,\ \forall i \in \{1, ..., N\},\ L_0^n(x) = \begin{cases} i \neq N_{\mathrm{f}} & \text{if } \phi^n(x) < 0 \\ N_{\mathrm{f}} & \text{if } \phi^n(x) \geq 0 \end{cases} \tag{4.4}$$

At each time step, we change the label value L_0, near the interface at the points where the condition (4.4) is not verified. Namely, if the level set function is positive, we set L_0 to the label of the fluid ($L_0 = N_{\mathrm{f}}$) and if the level set function is negative and the label function L_0 is still the label of the fluid, we assign to L_0 the value of its neighbours which are different from N_{f}. For all x where the condition (4.4) is not true we look for the closest neighbours. The proposed procedure is only adapted in the case of small displacements.

Let dt denotes the time step. For all point $x = (i, j)$ we denote by Neighbours(x) the set which contains the four closest points to x:

$$\text{Neighbours(x)} = \{(i+1, j), (i-1, j), (i, j), (i, j-1), (i, j+1)\}$$

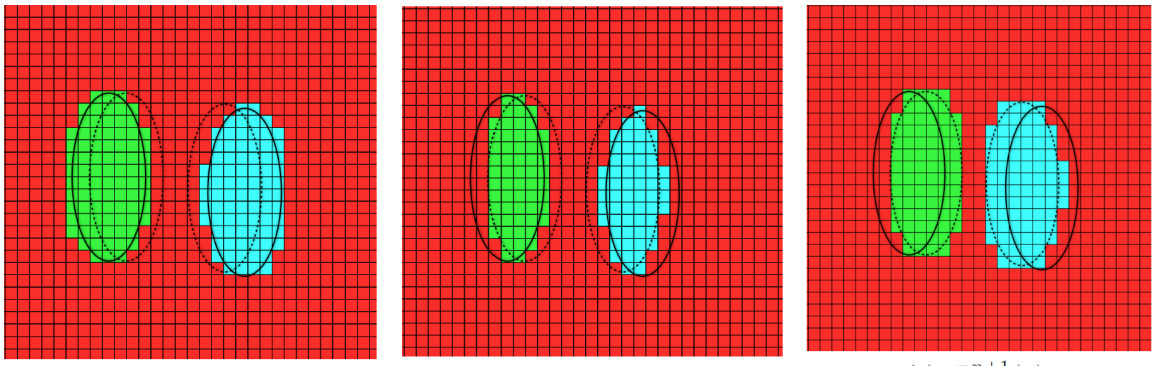
.

As the displacement is less than one discretization step, the label values are directly given by the algorithm (4). In the case of a displacement larger than one discretization step, one has to look for the closest neighbors of a grid cell in order to assign the right value of L_0 at this point.

Algorithm 4 *Redefinition of* L_0

```
for (x ∈ Ω) do
    if φ^{n+1}(x) ≥ 0 then
        L_0^{n+1}(x) = N_f
    else
        if (L_0^{n+1}(x) = N_f) then
            for (y ∈ Neighbours(x)) do
                if (L_0^{n+1}(y) ≠ N_f) then
                    L_0^{n+1}(x) = L_0^{n+1}(y)
                end if
            end for
        end if
    end if
end for
```

Let us note that this simple strategy is allowed only if the distance between two cells is strictly greater than one grid step. Here, this condition is imposed by our collision model.



(a) L_0^n. (b) if $(\phi^{n+1}(x) \geq 0)$ then $L_0^{n+1}(x) = N_f$ in the fluid. (c) $L_0^{n+1}(x)$.

Figure 4.1: Evolution of the label L_0 when two cells move. The red color corresponds to the object associated to the fluid.

4.1.3 Evolution of the label maps L_1

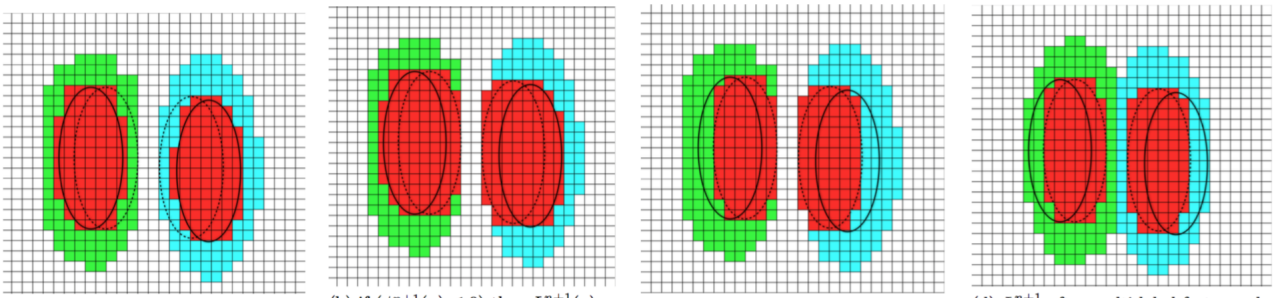
Contrary to the label map L_0 which is defined in the all computational domain, the label map L_1 is only defined close to the interface. At the points where L_1 is not defined

we assign to L_1 the value -1. The same procedure than the one applied for the label L_0 can be applied to L_1 close to the interfaces .

Algorithm 5 *Update of the label map* L_1

```
for (x ∈ Ω) do
    if (φ^{n+1}(x) ≤ 0) and (L_0^{n+1} ≠ N_f) then
        L_1^{n+1}(x) = N_f
    else
        if (L_1^{n+1}(x) = N_f) then
            for ( y ∈ Neighbours(x)) do
                if ((L_1^{n+1}(y) ≠ N_f)and(L_1^{n+1}(y) ≠ −1)  then
                    L_1^{n+1}(x) = L_1^{n+1}(y)
                end if
            end for
        end if
    end if
end for
```

With this procedure the label map L_1 has only been evolved close to the interface (see Figure 4.2) . To evolve the label map L_1 outside the particle and far from the interfaces a multi label fast marching method (MLBFMM) has to be performed.



(a) L_1^n. (b) if $(\phi^{n+1}(x) \leq 0)$ then $L_1^{n+1}(x) = N_f$ in the fluid. (c) Look for the closest neighbours $L_1^n \neq N_f$ (d) L_1^{n+1} after multi-label fast marching procedure.

Figure 4.2: Evolution of the label L_1 when two cells move. The red color corresponds to the object associated to the fluid. The white ones corresponds to the cells where L_1 is not defined.

4.2 Multi Label Fast marching

We present here the multi label fast marching procedure which allows to update φ_1, φ_2 and the label maps L_1 and L_2 in a vicinity of the structures' boundaries. This local fast marching is an extension of the fast marching method [156]. The idea of multi-label fast marching was first introduced in [161]. However, the method used only two labels and each of them was propagated individually (i.e. with a different speed). In [11], Bogovic extended this approach to the case of multiple object segmentation.
Here, this procedure is performed to update the label map L_1 and φ_1 close to the interfaces in a narrow-band of size NB_{L_1}. Then, if the narrow-band L_1 met the fast marching

algorithm to compute φ_2 and L_2 is activated.

To this end, we introduce a distance function d, for which we solve the eikonal equation 4.5 in the entire computational domain Ω. This equation is supplemented by an initial boundary condition which will depend on φ_1 or φ_2.

$$\forall x \in \Omega,\ |\nabla d(x)| = 1 \tag{4.5}$$

The discretization of this equation is achieved by a first order numerical scheme [151], so that denoting by Δx and Δy an horizontal and a vertical space step in two-dimensions, we solve:

$$\max(\max(D_x^- d_{ij}, 0)^2, \min(D_x^+ d_{ij}, 0)^2) + \max(\max(D_y^- d_{ij}, 0)^2, \min(D_y^+ d_{ij}, 0)^2) = 1 \tag{4.6}$$

where

$$D_x^- d_{ij} = \frac{d_{ij} - d_{i-1,j}}{\Delta x},\ D_x^+ d_{ij} = \frac{d_{i+1,j} - d_{i,j}}{\Delta x},\ D_y^- d_{ij} = \frac{d_{ij} - d_{i,j-1}}{\Delta y},\ D_y^+ d_{ij} = \frac{d_{i,j+1} - d_{i,j}}{\Delta y}.$$

4.2.1 Computation of the label map L_1 and φ_1

The first step of the multi label fast marching procedure is the computation of the distance function φ_1 and the values L_1 by solving the following equation:

$$\begin{cases} \forall x \in \Omega, & |\nabla d(x)| = 1 \\ \forall x \in \Gamma, & d(x) = \dfrac{\varphi_1(x)}{|\nabla \phi(x))|} \end{cases} \tag{4.7}$$

Initialization phase

At initialization, the function d is equal to $\dfrac{\varphi_1}{|\nabla \phi|}$ close to the interfaces, this allows to propagate all the interfaces simultaneously and to have a consistent initialization of the fast marching procedure.

The scheme (4.6) implies that the information moves in one direction only : from the lowest values of d to the highest values of d, the interfaces spread hence in two-directions: inside and outside cells. Moreover, each contour is associated to the number of objects. Thus, at initialization the computational domain Ω is partitioned into three sub domains:

- Alive (A) : set of points for which the values of d are known and at those points the values of φ_1 and L_1 will not be changed.
- Narrow-Band (NB): set of points were the values of d are unknown and the distance to A is less than one space discretization step.
- Far away (F) : all others points of the domain.

The Figure (4.3) provides an illustration of the domain decomposition at initialization for the spreading of one interface (boundary of the blue object). Inside the cell the boundary that propagates is the one associated to the fluid (red points) as it is the closest one. At initialization, the Alive set contains all the points of the N interfaces.The

Narrow-Band set consists of the closest points to the interfaces. All others points are considered as Far away and has to be computed and added to the narrow band if they are at a distance smaller than NB_{L_1}.
To achieve the storage and the sorting of the Narrow-Band according to the values of d, each cell of the heap contains the following informations:

- the integers (i, j) that localize the grid point
- the logical alive(i, j) which indicates if the point is in the Alive set
- the integer lab(i, j) which corresponds to the number of the interfaces that spread.
- the distance value $d(i, j)$: distance from the interface $\Gamma_{\text{lab}(i,j)}$.

The steps performed during the initialization phase are provided by the algorithm6.

Algorithm 6 *Initialization phase*

for $(x \in \Omega)$ **do**
  visit$(x) = 0$
  for $(y \in \text{Neighbours}(x))$ **do**
    if $(L_0(x) \neq L_0(y))$ **then** points at the interfaces
      if (alive$(y) =$ false0) **then**
        • Add y to the Narrow-Band
        • $d(y) = \dfrac{\varphi_1(y)}{|\nabla\phi(y))|}$
        • lab$(y) = L_0(x)$
        • alive$(y) =$ true The values $L_1(y) = lab(y)$ and $\varphi_1(y) = d(y)$ are fixed.
      end if
    else
      • lab$(x) = -1$
      • d$(x) = -1$
      • alive$(x) =$ false
    end if
  end for
end for

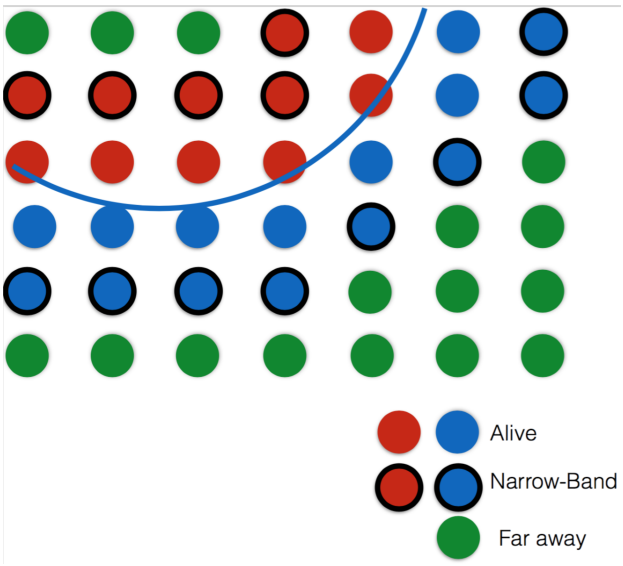


Figure 4.3: Partition of the computational domain for the multi-label fast marching at initialization. The interface spreads in two directions, inside and outside cells.

Iterative phase: Propagation of the interfaces

The direction of interfaces propagation depends on the values of the distance function d (from the lowest to the greatest values) and on the label value, namely one point is only added by its neighbors if it has the same label value.
During the propagation of the interfaces, the point x_m for which $d(x_m)$ is the lowest value of d in the narrow-band, is definitely deleted from the narrow-band and this point is considered as alive. This minimum value corresponds to the distance $\varphi_1(x_m)$ and we assign the label $\text{lab}(x_m)$ to $L_1(x_m)$. Then, the values of d at the neighbors of the point x_m are computed using the alive points which have the same label value, and these points are added to the Narrow-Band.The algorithm stops when the distance d has reached the narrow-band value NB_{L_1}. The steps performed during the iterative phase are provided by the algorithm 26.

Algorithm 7 *Iterative phase: computation of φ_1 and L_1*

while $((NB \neq \varnothing)$ and $(\text{dist} \leq NB_{L_1}))$ **do**
  Find x_m such as $d(x_m) = \min_{(x \in NB)} d(x)$
  $\text{dist} = \max(\text{dist}, d(x_m))$
  Delete x_m from the NB
  if $(L_1(x_m) = \text{lab}(x_m))$ **then**
    alive(x_m)=true
  end if
  if (alive(x_m)=false) **then**
    $\varphi_1(x_m) = d(x_m)$
    $L_1(x_m) = \text{lab}(x_m)$
    alive(x_m)=true
    for $(y \in \text{Neighbours}(x_m))$ **do**
      ajout=true
      if $(L_0(y) = \text{lab}(x_m))$ **then**
        ajout=false
      end if
      if $(L_1(y) = \text{lab}(x_m))$ **then**
        ajout=false
      end if
      if (ajout=true) **then**
        Compute $d(y)$ using Neighbors(y) such as L_1(Neighbors(y))=lab(x_m)
        Add y in NB
      end if
    end for
  end if
end while

4.2.2 Computation of the label map L_2 and φ_2

The same procedure is applied for the label map L_2 and φ_2, we solve the following equation:

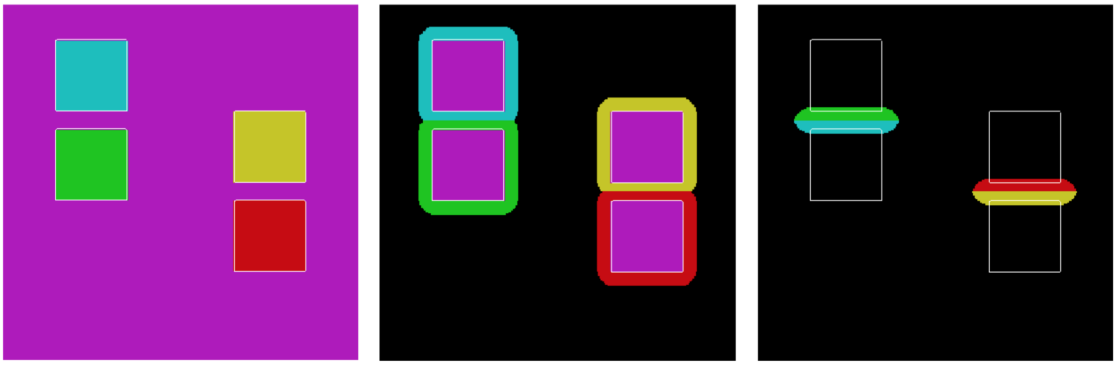
$$\begin{cases} \forall x \in \Omega, & |\nabla d(x)| = 1 \\ \forall x \in \Gamma_{L_1}, & d(x) = \varphi_2(x) \end{cases} \tag{4.8}$$

where a point $x \in \Gamma_{L_1}$ is defined by:

$$\forall y \in \text{Neighbours}(x), L_1(y) \neq L_1(x), L_0(x) = L_0(y)$$

Activation and computation of the boundary condition

The computation of L_2 and φ_2 is activated between two particles if the narrow-bands L_1 met, this means that the distance between the two particles is less or equal to $2NB_{L_1}$. If this happens, when L_1 and φ_1 are computed, the interfaces continue the propagation on one space grid in order to get the distance function φ_2 and the label map L_2 at the boundaries Γ_{L_1}. For instance, for a configuration of four objects like shown in Figure 4.4, the computation of L_2 is only activated between the pairwise of bodies.



L_0 L_1 L_2

Figure 4.4: Illustration of the multi label fast marching algorithm for a configuration of four particles. The black color corresponds to the undefined values of L_1 and L_2. The white contours corresponds to the zero value of the level set function ϕ.

Initialization phase

The initial narrow-band corresponds to the intersection of the closest objects in the fluid domain. Then these new values computed are frozen and will correspond to the alive points for which the values of φ_2 and L_2 are fixed. The initialization procedure is summarized in algorithm 8.

Algorithm 8 *Initialization phase*

for $(x \in \Omega)$ **do**
  alive(x)=false
  for $(y \in$ Neighbours$(x))$ **do**
    if $(L_1(x) \neq -1)$ and $(L_1(y) \neq -1)$ **then**
      if $(L_1(x) \neq L_1(y))$ and $(L_0(x) = L_0(y))$ **then** points at interfaces of closest objects in the fluid
        - Add y to the Narrow-Band
        - $d(y) = \varphi_2^{n+1}(y)$
        - lab$(y) = L_1(x)$
        - alive(y)=true The values $L_2(y) = lab(y)$ and $\varphi_2(y) = d(y)$ are fixed.

      else
        - lab$(x) = -\infty$
        - d$(x) = -\infty$
        - alive$(x) = 0$

      end if
    end if
  end for
end for

Iterative phase

We denote by NB_{L_2} the narrow-band distance for the computation of L_2 and φ_2. The iterative phase is provided by algorithm 9.

Algorithm 9 *Iterative phase: computation of φ_2 and L_2*

while $(NB \neq \varnothing)$ and $(\text{dist} \leq NB_{L_2})$ **do**
- Find x_m such as $d(x_m) = \min_{(x \in NB)} d(x)$

dist $= \max(\text{dist}, d(x_m))$
- Delete x_m from the NB

if $(L_1(x_m) = \text{lab}(x_m))$ **then**
alive(x_m)=TRUE
end if
if (alive(x_m)=false) **then**
- $\varphi_2(x_m) = d(x_m)$
- $L_2(x_m) = \text{lab}(x_m)$
- alive(x_m)=true

for ($y \in$ Neighbours(x_m)) **do**
ajout=true
if $(L_0(y) = \text{lab}(x_m)$ **then**
ajout=false
end if
if $(L_1(y) = lab(x_m))$ **then**
ajout=false
end if
if $(L_2(y) = lab(x_m))$ **then**
ajout=false
end if
if (ajout=true) **then**
- Compute $d(y)$
- Add y in NB

end if
end for
end if
end while

The present algorithm stops propagation when the distance function φ_2 reachs the distance value NB_{L_2}. Only the points close to the interface L_1 are visited. Figure 4.5 provides an illustration of the results obtained for the label map L_1 and L_2 for a configuration of 20 rectangular particles. The narrow-band size for L_1 is higher than the one for L_2. As you can see, that the computation of L_2 is achieved only for the closest structures allowing to reduce the CPU time.

4.2.3 Stop condition when NB_{L_1} and NB_{L_2} are not active

When the narrow-band is not active and so if the label maps and distance functions are computed in the entire domain, it is more efficient to propagate the label L_1 and L_2 simultaneously by using one common priority queue. Figure 4.7 provides an illustration of the multi-label fast marching procedure for three rectangular cells when the interfaces propagate in the whole computational domain. The background is the red object and in

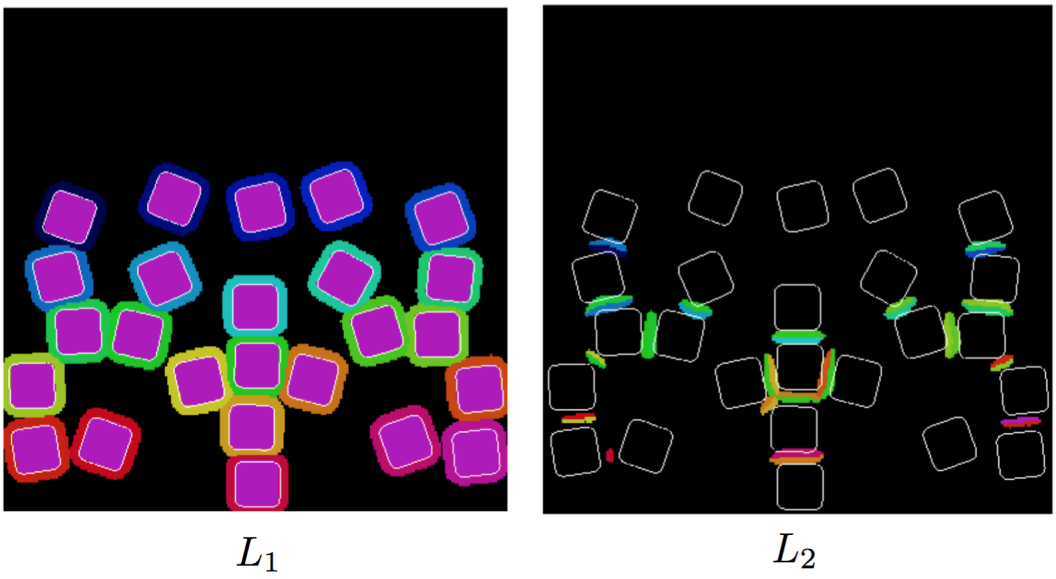


Figure 4.5: Illustration of the multi label fast marching algorithm for 20 objects. The black color corresponds to the undefined values of L_1 and L_2. The white contours corresponds to the zero value of the level set function ϕ

our case will be the fluid. The image 4.7 (b) shows the initial Narrow-Band and Alive points. The image (c) shows the expansion of the four interface at a step k. The red object boundary has finished its propagation giving inside the cells the L_1 map values (already known) and the new distance function φ_1. Outside the cells, the three bodies interfaces have propagated giving the L_1 map values and the white grid points correspond to the Far away points which have not been visited yet. In the case of dense suspensions of bodies, the stop condition can provide larger computational saving. This multi label fast marching is in a certain extent a local fast marching for different regions of the domain.

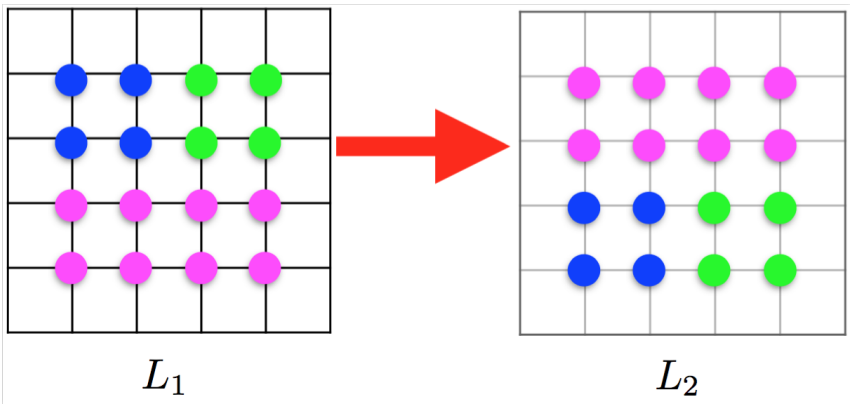
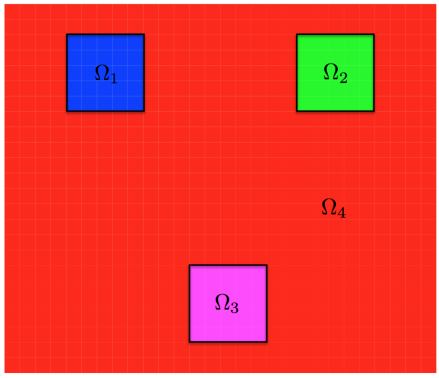
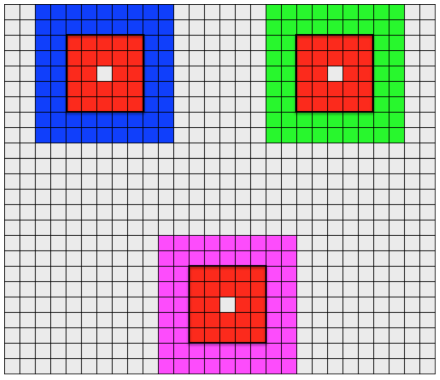


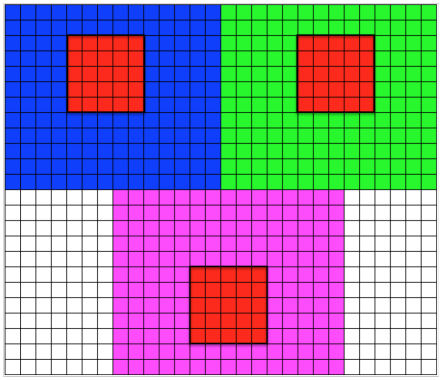
Figure 4.6: Step $k+1$. Values of the label map L_2 at intersection of the interfaces.



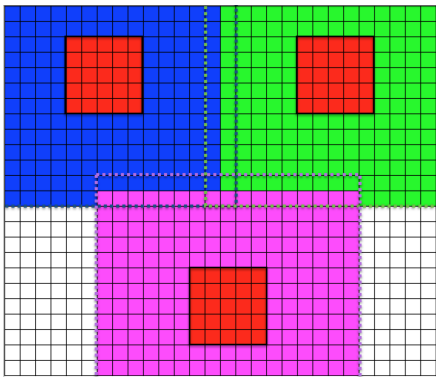
(a) Three rectangular cells immersed in a fluid Ω_4.



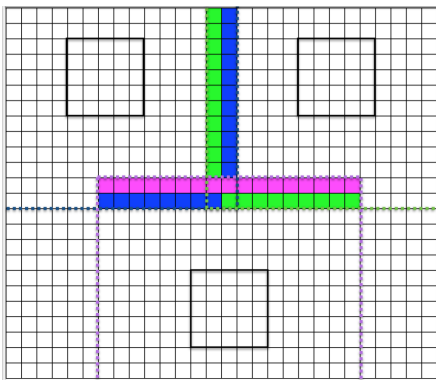
(b) Initialization phase: Narrow-band and Alive points.



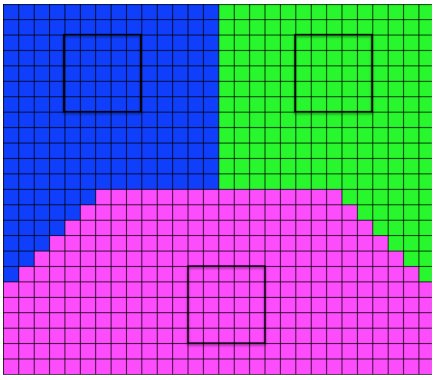
(c) Iteration k. The white cells represent the Far away points. Colours correspond to the L_1 map values.



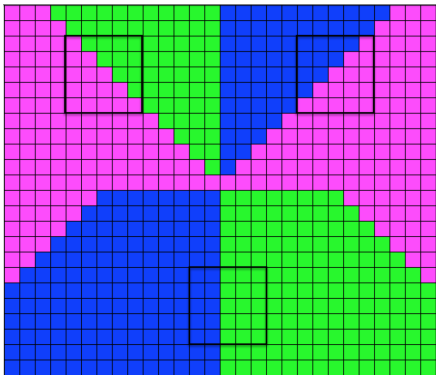
(d) Step k+1. The interfaces continue the propagation. Colours correspond to the L_1 map values.



(e) L_2 map values at step $k+1$.



(f) Final L_1 map.



(g) Final L_2 map.

Figure 4.7: Illustration of the multi label fast marching algorithm.

4.3 Computational complexity

This section focus on the computational complexity of the multi-label fast marching (MLFMM) algorithm presented above. The only dependence on the number of particles of the proposed approach to locate and deal with numerical contacts, comes from this procedure.
Depending on the implementation of the MLFMM, the CPU time can be decreased. To compare the CPU time of these procedures , we consider a rectangular computational domain $\Omega = [0, 2] \times [0, 1]$ and we add circular particles in the domain. The particles are added next to each other until the half of the domain is almost filled with 400 particles. The space grid is $h = 1.95.10^{-3}$ and the size of the particles corresponds to ten discretization space step.

4.3.1 Propagation in the whole computational domain

We first provide the CPU time of the procedure when the interfaces are propagated in the entire computational domain, this means that the five field functions are defined in the entire computational domain.

In the case of N particles and denoting by M the number of grid points in the domain, the worst computational complexity of the fast marching procedure to compute φ_1 and L_1 is $NM \log(NM)$.
The worst complexity of the fast marching procedure to compute φ_2 and L_2 is also $NM \log(NM)$ leading to a complexity of MLFMM of $2NM \log(NM)$. These algorithms rely significantly on the number of particles.

An other way is to consider N different linked lists to propagate the N interfaces, the same stop condition is applied in the sense that if the label map L_1 and L_2 are defined for all points, the procedure stops. The N interfaces are propagated in parallel and stops propagation when the label map L_2 is defined at all points. Consequently, the interface Γ_i of a particle only propagates in a neighborhood of the particle Ω_i. However, one has to check at each iteration of the procedure if the label map L_2 is defined at all points, a browsing of the list of all points of the domain is required, this induces a very high computational cost.

Instead, one can consider N lists which allows to propagate the N interfaces in the whole computational domain. This procedure is equivalent to achieve N fast marching procedures inside and outside the particles.
At initialization, each list $LIST_k$ corresponds to the interface Γ_k for which the label L_0 is equal to k inside or $L_1 = k$ outside the particle. The distance associated to each list $LIST_k$ corresponds to φ_1 if $L_0 = k$ or $L_1 = k$.
Then, when the interfaces are propagated in the all domain, the label L_1, L_2 and the distance functions φ_1 and φ_2 are recovered by using the values of the N obtained distances. We denote by $LIST_k$ the kth list which corresponds to the object Ω_k. Using the N lists,

the functions are explicitly provided by:

$$\forall x \in \Omega, \forall k \in [1,..,N], \begin{cases} L_1(x) & = \text{argmin}_{k \neq L_0} LIST_k(x) \\ L_2(x) & = \text{argmin}_{(k \neq L_0, k \neq L_1)} LIST_k(x) \\ \varphi_1(x) & = LIST_{L_1(x)}(x) \\ \varphi_2(x) & = LIST_{L_2(x)}(x) \end{cases} \tag{4.9}$$

In the case of N lists, each interface is spread until all points are visited.
For each list, adding a point in the binary heap has a complexity of the order $\mathcal{O}(\log(M))$ as well as the deleting procedure. Thus, the N fast marching complexity is of order $(NM)\mathcal{O}(\log(M))$ which is better than the MLFMM complexity.
Table 4.1 provides the CPU time of both algorithms. It appears that the procedure to compute L_1 using the MLFMM is faster than the N fast marching procedure.
The computation of the label map L_2 is the most time consuming procedure, it represents over 60% of the MLBFMM CPU time for 400 interfaces. For 400 particles, the CPU time of the MLFMB has exceeded the N FMM. To lower the computational cost, we propose to activate the computation of L_2 only for the closest particles and to propagate the interfaces in a vicinity of the particles' boundaries.

N	N FMM	MLFMM		
		L_1	L_2	Total
10	1.5	0.5	1.4	2
50	8	1	4.	5.
100	16.	2.4	10.6	13
200	38.4	10.	22.	32.
400	78.2	40	75	115

Table 4.1: Comparison of the CPU time of the N FMM and the multi label fast marching method.

4.3.2 Activation of the narrow-band NB_{L_1} and NB_{L_2}

We compare the N lists procedures and the MLFMM when the narrow-band condition NB_{L_1} and NB_{L_2} are activated. The N signed distances are computed in a narrow-band of size NB_{L_1} allowing to update the label maps and two distance functions inside these narrow-bands. Likewise, the computation of φ_1 and L_1 is achieved in a narrow-band of size NB_{L_1} and if the particles are at a distance $2NB_{L_1}$, the computation φ_2 and L_2 is activated in a narrow-band of size NB_{L_2}.

To compare the CPU time, we add the particles next to each other in the domain, the particles are close enough so that the computation of L_2 is activated as soon as there are two bodies in the domain.
Table 4.2 shows the CPU time of both algorithms for $NB_{L_1} = 10h$ and $NB_{L_2} = 5h$ where h is the space discretization step. Using the narrow-band restriction, the N lists procedure is highly less time consuming than the MLFMM. Moreover, the computation of L_2 with the MLFMM has significantly decreased.
The most optimal algorithm is hence to use the N lists procedure. However, after the N lists procedure, the label map L_2 is only defined at the overlapping regions of the narrow-bands NB_{L_1}, whereas with the MLFMM L_2 is defined as soon as the isolines NB_{L_1} are in

contact, the computation of L_2 is then activated and stops when φ_2 reachs the distance NB_{L_2}. This difference is illustrated in Figure 4.9 for two configurations of particles. The inconvenient is that as long as L_2 is not defined on an interface, the short-range repulsive force is not activated, and thus depending on the size NB_{L_1}, the numerical contacts may not be handled.
Instead of using the N lists procedure, we propose to use the N lists to compute φ_1 and L_1 and the MLFMM is activated if two particles are at a distance $2NB_{L_1}$ to compute L_2 and φ_2 on a distance NB_{L_2}. The CPU time of this last algorithm is represented on Figure4.8 for different number of particles close to each other for $NB_{L_1} = 10h$ and $NB_{L_2} = 5h$.

N	N FMM	MLFMM		
		L_1	L_2	Total
10	0.04	0.006	0.01	0.02
50	0.25	0.35	0.02	0.37
100	0.45	1.3	0.07	0.7
200	1.3	8.4	0.8	9.1
400	2.5	34	3.6	25

Table 4.2: Comparison of the CPU time of the N fast marching procedures and the multi label fast marching method in a thin band around the particles ($NB_{L_1} = 10h$,$NB_{L_2} = 5h$).

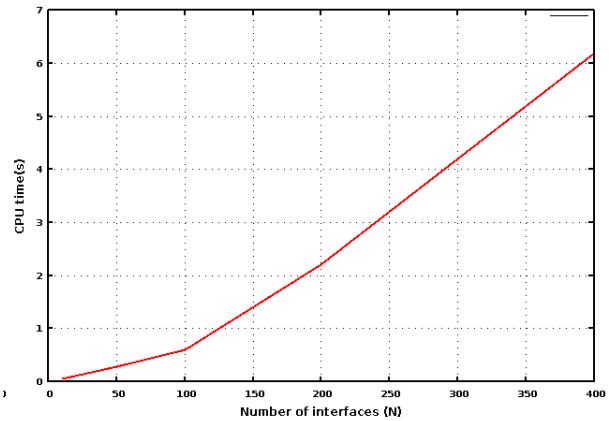
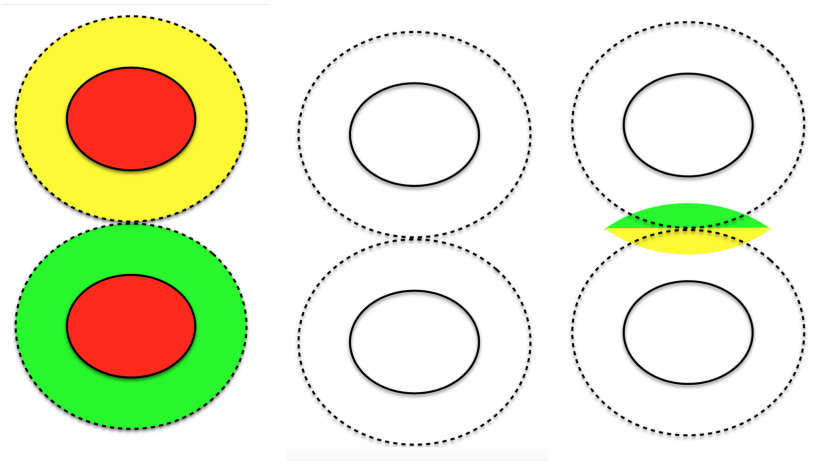
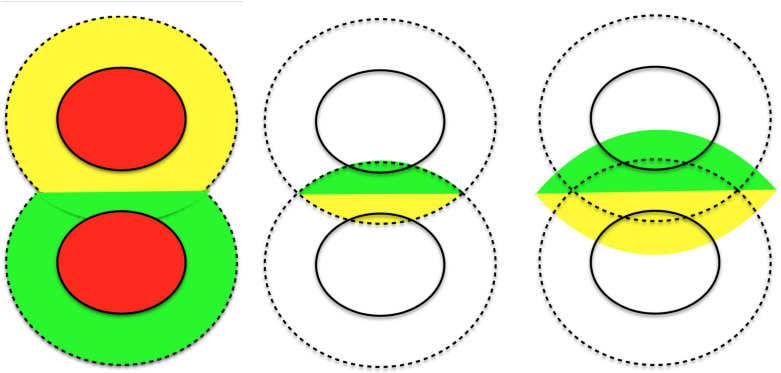


Figure 4.8: CPU time of the most efficient algorithm: Computation of L_1 and φ_1 with N lists and computation of L_2 and φ_2 like presented in section 5.3.2.



(a) L_1 after N FMM or MLFMM. (b) L_2 after N FMM. (c) L_2 after MLFMM.



(d) L_1 after N FMM or MLFMM. (e) L_2 after N FMM. (f) L_2 after MLFMM.

Figure 4.9: Label map L_2 obtained after a multi-label fast marching or after the procedure using N lists for two different configurations (a) and (d). The dashed lines represents the isolines NB_{L_1}.

Conclusion

In this chapter, we have described the numerical evolution of the label maps and distance functions. A level set function ϕ which captures the union of all interfaces is introduced, this level set field is a signed distance function in a vicinity of the interfaces, at a distance NB_{L_1} from the interfaces. Inside these narrow-bands, the level set function ϕ is transported with the fluid or structures velocities, allowing to evolve explicity the label functions L_0 and L_1 close to the interfaces. Then, a multi-label fast marching algorithm is performed in the vicinity of the particles.
First, the L_1 map and distance function φ_1 are updated by propagating the N interfaces over a distance NB_{L_1}, using N lists. When two particles are at a distance smaller than $2NB_{L_1}$, that is when two L_1 narrow-bands are in contact or overlapped, a fast marching procedure is activated to define the Label function L_2 and the distance function φ_2. The interfaces of the second closest object L_2 are propagated on a distance NB_{L_2} which has to be large enough to apply the short range repulsive force to avoid numerical contacts between the particles.

Chapter 5

Application to immersed rigid bodies

Contents

This chapter proposes an application of the model to rigid bodies immersed in an incompressible fluid.
In a first place, we describe the model adopted for the numerical simulations as well as its numerical resolution. This model will be compared numerically to a penalization model formulated in the context of a level set decomposition. The second part will be dedicated to the numerical illustrations.
First, we will present a qualitative grid convergence in two and three dimensions through the numerical simulation of spherical particles falling under gravity, two cases are considered: a constant intensity of the repulsive force and one depending on the relative velocities between closest particles.
Then, the proposed model (5.17) and the model (5.11) which involves N level set functions are compared in terms of computational complexity and the influence of the two collision models on rigid bodies dynamics is explored. The proposed short-range repulsive force is also compared to an algorithm proposed in [124] which enables to impose a minimal distance between particles. Finally, some simulations of dense suspensions of circular rigid bodies in two and three dimensions are presented.

5.1 Penalization models for fluid/"many rigid body" coupling

A penalization method is used to take into account the rigid bodies immersed in the fluid. In a first section, we present the penalization model for several immersed rigid

bodies. Then, the penalization model formulated in a level set decomposition framework is presented. Finally, a penalization model reformulated by means of the label maps and distance functions is introduced. Two numerical algorithms to evolve the structures are described and compared in terms of computational complexity.

5.1.1 Penalization model for many rigid bodies

The FSI technique adopted to take into account the rigid bodies immersed in a fluid is the penalization method [4; 140]. We recall here the Penalized Navier Stokes equations for N rigid bodies evolving in a fluid.
The computational domain $\Omega \in R^d, d = 2, 3$ is decomposed as

$$\Omega = \Omega_{N_f} \cup \left(\bigcup_{i=1}^{N} \Omega_i \right)$$

where Ω_{N_f} is the fluid subdomain and Ω_i is the region occupied by the ith rigid body and Γ_i its boundary. The boundaries of the computational domain Ω are denoted by $\partial\Omega_i, i \in [1, .., 2d], d = 2, 3$.
The fluid density and viscosity are denoted by ρ_f and μ and U and P are the flow velocity and pressure.

The interaction between the structures and the fluid is modelled by the Navier Stokes equations for the fluid, the rigid motions in the solid bodies and the no slip boundary conditions on each rigid body interface. The penalization model for N rigid bodies is given by:

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U) - \nabla \cdot (\nabla U) + \nabla p = \rho g + \lambda(\sum_{i=1}^{N} \chi_i (U_i - U)) & \text{in } \Omega_T = \Omega \times (0, T) \\ \nabla \cdot U = 0 & \text{in } \Omega_T \end{cases} \tag{5.1}$$

where, $\lambda >> 1$ is the penalization coefficient, χ_i is the characteristic function of the solid Ω_i defined as:

$$\chi_i = \begin{cases} 0 & \text{in } \Omega_i \\ 1 & \text{in } \Omega_{N_f} \end{cases} \tag{5.2}$$

Denoting by ρ_i the density of the immersed bodies Ω_i, the density function is given by:

$$\rho = \rho_f + \sum_{i=1}^{N} (\rho_f - \rho_i)\chi_i. \tag{5.3}$$

Numerically, a regularized version χ_i^ε of this function is used. As the rigid bodies move with the fluid, the displacement of χ_i is:

$$\forall x \in \Omega, \forall i \in \{1, ..., N\}, \quad \partial_t \chi_i + U \cdot \nabla \chi_i = 0. \quad \text{or} \quad \partial_t \chi_i + U_i \cdot \nabla \chi_i = 0.$$

Each rigid velocity U_i is obtained by averaging translation and angular velocities over the solid body Ω_i ([140]):

$$\begin{cases} U_i(x) & = U_i^t + W_i \times (x - x_i^g) \\ & = \dfrac{1}{|\Omega_i|} \int_{\Omega_i} \rho \chi_i U dx + \left(J_i^{-1} \int_{\Omega_i} \rho \chi_i U \times (z - x_i^g) dz \right) \times (x - x_i^g) \end{cases} \tag{5.4}$$

where J_i is the inertial matrix of the body Ω_i and x_i^g its gravity center. These equations has to be supplemented by boundary conditions for the fluid velocity and initial conditions for the studied variables.

5.1.2 The penalization model using a level set decomposition

The penalization model using a level set decomposition consists in the penalized Navier Stokes equations supplemented by the collision model introduced in [31], we also add the repulsive forces exerted by the wall on the structures. As each solid body Ω_i is captured by one level set function ϕ_i, the regularized characteristic function χ_i^ε is provided by:

$$\chi_i^\varepsilon = 1 - H_\varepsilon(\phi_i). \tag{5.5}$$

The density function is expressed as:

$$\rho = \rho_f + \sum_{i=1}^{N} (\rho_f - \rho_i)\chi_i^\varepsilon. \tag{5.6}$$

Each level set function is transported either by the fluid or the rigid motion of the associated body.

$$\forall x \in \Omega,\ \forall i \in \{1, ..., N\}, \quad \partial_t \phi_i + U \cdot \nabla \phi_i = 0. \qquad \text{or} \qquad \partial_t \phi_i + U_i \cdot \nabla \phi_i = 0.$$

We recall that the repulsive forces exerted between the particles are provided by:

$$\forall x \in \Omega,\ F_{\text{global}}(x) = \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \frac{k}{\varepsilon} \zeta_\varepsilon(\phi_i(x)) \frac{\nabla \phi_j(x)}{\phi_j(x)} \exp(-\frac{\phi_j(x)}{\varepsilon_b}). \tag{5.7}$$

To avoid contacts between the particles and the walls, we define a repulsive force by using a level set function ϕ_{N+1} which captures the union of all interfaces, this level set function corresponds to the minimum of the N level set functions provided by:

$$\forall x \in \Omega, \phi_{N+1}(x) = \min(\phi_1(x), \phi_2(x), ..., \phi_N(x)) \tag{5.8}$$

Denoting by ψ_i the distance functions to the boundary $\partial\Omega_i$ as:

$$\forall x \in \Omega, \psi_i(x) = d(x, \partial\Omega_i) \tag{5.9}$$

We define the repulsive forces exerted by the walls by:

$$\forall x \in \Omega, F_{\text{wall}}(x) = \sum_{i=1}^{2d} \frac{k}{\varepsilon} \zeta_\varepsilon(\phi_{N+1}(x)) \frac{\nabla \psi_i(x)}{\psi_i(x)} \exp\left(-\frac{\psi_i(x)}{\varepsilon_b}\right). \tag{5.10}$$

Finally, the system involving the N level set functions and the collision model F_{global} is as the following:

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U) - \nabla \cdot (\mu \nabla U) + \nabla p = \rho g + \lambda(\sum_{i=1}^{N} \chi_i^\varepsilon (U_i - U)) + F_{\text{global}} + F_{\text{wall}} & \text{in } \Omega_T \\ \nabla \cdot U = 0 & \text{in } \Omega_T \\ \forall i,\ \partial_t \phi_i + U_i \cdot \nabla \phi_i = 0 & \text{in } \Omega_T \end{cases} \tag{5.11}$$

5.1.3 The proposed penalization model

The system (5.1) depends on the number of bodies N contained in the fluid.
Indeed, the computation of N rigid velocities as well as the N transport equations are required in order to evolve the N structures.
In addition, the penalization term $\lambda(\sum_{i=1}^{N}\chi_i(U_i - U))$ and the density function depends also on the number of bodies.
In order to alleviate the high computational cost due to a large number of particles N we take advantage of the label maps and distance functions to reformulate this model. To achieve this, we define a label map $L_{0,1}$ which allows to partition the computational domain into N subdomains, each subdomain is associated to the first closest structure, this function is defined as:

$$\forall x \in \Omega, \quad L_{0,1}(x) = \begin{cases} L_0(x) & \text{if} \quad (L_0(x) \neq N_f) \\ L_1(x) & \text{otherwise} \end{cases} \tag{5.12}$$

The regularized characteristic functions of the solid bodies are then expressed as:

$$\forall x \in \Omega, \forall y \in \Omega, \quad \chi_{L_{0,1}(x)}(y) = 1 - H(\frac{\phi_{L_{0,1}(y)}}{\varepsilon})$$

Denoting by $\rho_{L_{0,1}(x)}$ the density of the body $\Omega_{L_{0,1}(x)}$ we obtain the following density function:

$$\rho_x = \rho_f(1 - \chi_{L_{0,1}(x)}) + \chi_{L_{0,1}(x)}\rho_{L_{0,1}(x)}$$

Computation of the rigid velocities

To compute the penalization term, the values of the rigid velocities are only required inside the particles and in a vicinity of the particle at a distance ε. Taking this in consideration and the advantage of the formulation above, we define a global rigid velocity which includes the N rigid velocities $U_i, i \in \{1, ..., N\}$.
More precisely, for each $x \in \Omega$, $y \to U_{L_{0,1}(x)}(y)$ is the rigid velocity of the solid body $\Omega_{L_{0,1}(x)}$ obtained by averaging the translation and angular velocities over the solid.

Setting:

$$\forall x \in \Omega, \quad |\Omega_{L_{0,1}(x)}| = \int_{\Omega_{L_{0,1}(x)}} \rho(z)dz = \int_{\Omega} \rho(z)\chi_{L_0^{\varepsilon}(x)}(z)dz \tag{5.13}$$

we obtain the following formulation .
$\forall x \in \Omega,\ \forall y \in \Omega,$

$$\begin{aligned} U_{L_{0,1}(x)}(y) = &\frac{1}{|\Omega_{L_{0,1}}|}\int_{\Omega} \rho_x(z)\chi_{L_{0,1}(x)}(z)U(z)dz \\ &+ \left(J^{-1}_{L_{0,1}(x)}\int_{\Omega} \rho_x(z)\chi_{L_{0,1}(x)}(z)U(z) \times (z - x^g_{L_{0,1}(z)})dz\right) \times (y - x^g_{L_{0,1}(y)}). \end{aligned} \tag{5.14}$$

where $J_{L_{0,1}(x)}$ and $x^g_{L_{0,1}(x)}$ are the inertial matrix and center of gravity of solid $\Omega_{L_{0,1}(x)}$.
In a neighboorhood of a particle Ω_i that is when $L_{0,1} = i$, the defined global velocity

corresponds exactly to the rigid velocity U_i.

On the implementation side, all rigid velocities are computed incrementally, involving only one iteration on the mesh grid.
To compute the coefficient of the repulsive forces, which depends on the relative velocities of the closest particles, we extend the velocities in each portion where the label map L_2 is defined that is by setting:
$\forall x \in \Omega,\ \forall y \in \Omega,$

$$U_{L_2(x)}(y) = U^t(L_2(x)) + W(L_2(x)) \times (y - x^g(L_2(y))) \tag{5.15}$$

where U^t is the translational velocity and W the angular velocity.

Evolution of the solid bodies

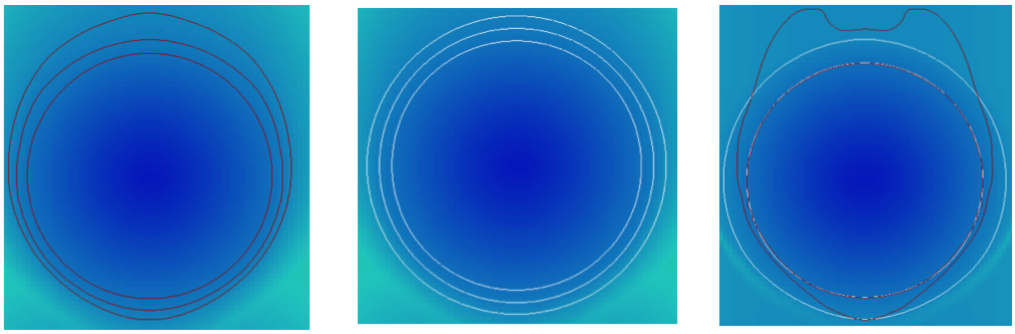
To evolve the rigid bodies two algorithms are considered. The first one consists in evolving the level set function which captures the union of all interfaces and then perform the evolution introduced in Chapter 3. The second one, consists in reconstructing the N level set functions by using the method proposed in [31] in order to evolve the structures and recover the label maps and distance functions.

Transport of the level set function ϕ

The advection of the level set function can be achieved either with the global rigid velocity or with the fluid velocity. Mainly, we have:

$$\partial_t \phi + U \cdot \nabla \phi = 0 \text{ or } \partial_t \phi + U_{L_{0,1}} \cdot \nabla \phi = 0$$

To compare the results obtained with these two advection equations, we consider the test case of a circular disk falling under gravity and we do not reinitalize the level set function as a signed distance function. Figure 5.1 shows the obtained level set function transported by the fluid velocity field and the rigid velocity for one particle, the isolines corresponds to $\phi = \varepsilon$ and $\phi = 0$. Inside the particle, the isolines are not deformed as the fluid velocity field match the rigid one, whereas outside the fluid velocity field deforms the isolines. It is thus more convenient to transport ϕ with the rigid velocity $U_{L_{0,1}}$ which is more regular than the fluid velocity field in a vicinity of the rigid bodies.



(a) $t = 0.05$ (b) $t = 0.05$ (c) $t = 0.1$

Figure 5.1: Isolines $\phi = \varepsilon$ and $\phi = 0$ transported by the fluid velocity (red lines) and by the rigid velocities $U_{L_{0,1}}$ (white lines).

Reconstruction of the N level set function

The transport equation of the level set function even with the rigid velocity does not conserve exactly the signed distance function, as spatial and time discretization errors are introduced during numerical calculations. As the considered bodies are rigid, an efficient technique proposed in [31] is to evolve the structures by using the gravity centers and the transformations over the solids. We present the technique in the case of one rigid body captured by a level set function ϕ in three dimensions, the extension to several bodies is straightforward.
Taking advantage of the rigidity property of the displacement, a simple interpolation of the initial level set function $\phi_0(x)$ is achieved using the characteristics of the velocity field. Denoting by $X(0,x,t)$ the characteristics of the velocity field we get:

$$\phi(x,t) = \phi_0(X(0,x,t))$$

Setting :

$$\theta^n = |w^n|\Delta t, \quad , \frac{w^n}{|w^n|} = (\alpha,\beta,\gamma)$$

where θ^n is the step rotation and $\dfrac{w^n}{|w^n|}$ corresponds to the normalized rotation vector around which the solid body turns, the rotational matrix is provided by:

$$R^n = \begin{pmatrix} 1-2b^2-2z^2 & 2ab-2cd & 2ac+2bd \\ 2ab+2cd & 1-2a^2-2c^2 & 2bc-2ad \\ 2ac-2bd & 2bc+2ad & 1-2a^2-2b^2 \end{pmatrix}$$

where

$$a = \alpha\sin(\frac{\theta^n}{2}), \quad , b = \beta\sin(\frac{\theta^n}{2}), \quad , c = \gamma\sin(\frac{\theta^n}{2}), \quad d = \cos(\frac{\theta^n}{2})$$

The position at time t^{n+1} of a point X which moves with the solid body can be deduced in function of its position at time t^n:

$$X^{n+1} = c^n + U^n\Delta t + R^n(X^n - c^n)$$

and by accumulation of rotations and translations we get:

$$X^{n+1} = \prod_{i=0}^{i=n} R^i(x - x_g^0) + x_g^{n+1}$$

The position of the gravity center is directly obtained by translation as:

$$x_g^{n+1} = x_g^n + \Delta t U^t$$

Thus, the computation of the level set function ϕ^{n+1} consists in first computing for all grid points x_i the characteristics $X(0,x_i,t^{n+1})$ and then interpolate $\phi_0(X(0,x_i,t^{n+1}))$ from the grid values by bilinear interpolations giving:

$$\phi(X^{n+1},t^{n+1}) = \phi_0\left(\frac{1}{\prod_{i=0}^{i=n} R^i}(X^{n+1} - x_g^{n+1}) - x_g^0\right) \tag{5.16}$$

At each time step, the error due to transport equation is constant and depends only on the choice of the interpolation function. Moreover, the computation of the transport equation is fast, only the interpolations can induces a high computational cost if high order interpolation scheme are used.
A linear interpolation is not sufficient for this work, as the distance between the bodies is required and has to be accurate. However, as the proposed collision model consists in a short-range repulsive force, the signed distance function are only required in a vicinity of the particles interfaces. In the case of a spherical particle, the gravity center is transported :

$$x_g^{n+1} = x_g^n + \Delta t U^t$$

and we define ϕ as the distance to the gravity center as the distance to a sphere is a very fast procedure. For more general shape like ellipsoids, the distance to the interface is not explicit and it induces a very high computational cost in particular when dealing with several bodies, it is thus faster to interpolate the initial level set function.

The complete penalization model

Finally, the proposed penalized model is provided by:

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U) - \mu\Delta U + \nabla P \\ \qquad = \rho g + \lambda(\chi_{L_{0,1}}(U_{L_{0,1}} - U)) + F_{\text{label}} + F_{\text{wall}} & \text{in } \Omega_T \\ \nabla \cdot U = 0 & \text{in } \Omega_T \end{cases} \tag{5.17}$$

Denoting by ϕ_i the distance functions to the boundary $\partial\Omega_i$ as:

$$\forall x \in \Omega, \phi_i(x) = d(x, \partial\Omega_i) \tag{5.18}$$

the repulsive forces exerted by the walls are given by:

$$\forall x \in \Omega, F_{\text{wall}}(x) = \sum_{i=1}^{2d} \frac{k}{\varepsilon} \zeta_\varepsilon(\phi(x)) \frac{\nabla\psi_i(x)}{\psi_i(x)} \exp\left(-\frac{\psi_i(x)}{\varepsilon_b}\right). \tag{5.19}$$

and we recall the particle-particle repulsive force:

$$\forall x \in \Omega,\ F^{\text{label}}(x) = \frac{K_{L_{0,1},L_2}}{\varepsilon} \rho(x) \zeta_\varepsilon(\phi(x)) \frac{\nabla\varphi_2(x)}{\varphi_2(x)} \exp\left(-\frac{\varphi_2(x)}{\varepsilon_b}\right) \tag{5.20}$$

where $K_{L_{0,1},L_2}$ is either a constant value fixed at the beginning of the simulation or it depends on the relative velocities between the particles. This model has to be completed with the evolution of the solid bodies and the initial and boundary conditions on the velocities and pressure.
Thanks to the label maps, we have suppressed the dependence on the number of bodies in the repulsive force, the penalization term, the density function and the computation of the rigid velocities.

5.1.4 Discretization and Numerical implementation

This part elucidates the numerical resolution of the system (5.17), two algorithms are considered for the evolution of the rigid bodies and we provide a comparison of these two algorithms in terms of computational complexity.

Resolution of a multi-fluid problem in the fluid/structures domain

The Navier-Stokes equations are solved using an incremental projection method of Chorin type. Given a time step Δt, we set $t^n = n\Delta t$ and $U^n \approx U(.,t^n)$. First, we compute an intermediate State U^* from:

$$U^* = U^n - \Delta t(U^n \cdot \nabla)U^n + \frac{\Delta t\mu}{\rho^n}\Delta U^n + \Delta t g - \frac{\Delta t \nabla p^n}{\rho^n}. \tag{5.21}$$

Then we solve the pressure from the equation:

$$\nabla \cdot (\frac{\nabla p^{n+1}}{\rho^n}) = \frac{\mathrm{div}(U^*)}{\Delta t} + \nabla \cdot (\frac{\nabla p^n}{\rho^n}) \tag{5.22}$$

so that the velocity:

$$\overline{U} = U^* - \frac{\Delta t}{\rho_0}(\nabla p^{n+1} - \nabla p^n) \tag{5.23}$$

is divergence free. The pressure equation (5.22) can be solved directly by a conjugate gradient algorithm or approximated with a relaxation procedure as follows:

$$\Delta p^{n+1} = \frac{\rho_0}{\Delta t}\,\mathrm{div}(U^*) + \Delta p^n$$

setting $1/\rho^n = 1/\rho_0 - (1/\rho_0 - 1/\rho^n)$ in order to get a Poisson equation much faster to solve. In that case, the step $\overline{u}$ satisfies Navier-Stokes equations with a modification of the pressure term $(1/\rho^n - 1/\rho_0)\nabla p^n + 1/\rho_0 \nabla p^{n+1}$ instead of $1/\rho^n \nabla p^{n+1}$. It remains to choose ρ_0 in order to have the best approximation. We have tested three different values $\rho_0 = \rho_f$, $\rho_0 = \rho_{L_{0,1}}$ and $\rho_0 = (\rho_{L_{0,1}} + \rho_f)/2$. The Poisson solver is based on a classical 5 or 7 points second order stencil according to the dimension, the viscous terms are discretized by a second order central scheme and the convection term is discretized by a 5^{th} order WENO scheme.
To find the best value for ρ_0, we perform a numerical test with six rigid disks falling under gravity in two dimensions. The Figure 5.2 shows that the results obtained with $\rho_0 = \rho_f$ gives the worst approximation whereas the two other values are very close to the approximation of the exact solution computed with the conjugate gradient. In the following we set $\rho_0 = (\rho_{L_{0,1}} + \rho_f)/2$.

Concerning the penalization term an implicit treatment is adopted in order to use larger penalization coefficient λ and therefore the interface boundary condition is satisfied with better accuracy.

Algorithm 1

In the general case the algorithm performs the following steps:

1. Compute and add the repulsive force $U_{\mathrm{col}} = \overline{U} + \Delta t F^{\mathrm{lab}}_{\mathrm{col}}$,

2. Compute the translation velocity $U^t_{L_{0,1}}$ of the body $\Omega_{L_{0,1}}$, $U^t_{L_{0,1}} = \dfrac{\int_\Omega \rho^n U_{\mathrm{col}} \cdot \chi_{L^n_{0,1}} dx}{\int_\Omega \rho^n \chi_{L^n_{0,1}} dx}$,

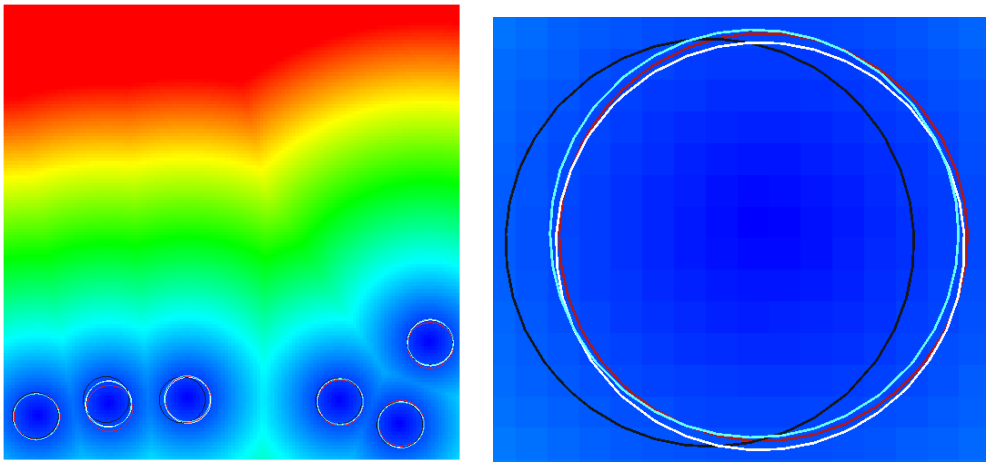


Figure 5.2: Flow field for six rigid disks falling under gravity (left) and a zoom of the location of the interface for different values of ρ_0 (right). The conjugate gradient method is in red, $\rho_0 = \rho_f$ in black, $\rho_0 = \rho_{L_{0,1}}$ in white and $\rho_0 = (\rho_{L_{0,1}} + \rho_f)/2$ in blue. The background color shows the level set amplitude.

3. Compute the angular velocity $W^{L_{0,1}}$ of the body $\Omega_{L_{0,1}}$

$$W^{L_{0,1}} = J^{-1}_{L_{0,1}} \int_\Omega \rho^n r^n \times U_{\text{col}} \cdot \chi_{L^n_{0,1}} dx \times r^n, \quad \text{where} \quad r^n = (x - x^g_{L^n_{0,1}})$$

4. Compute the rigid velocity $U_{L_{0,1}}$ of the body $\Omega_{L_{0,1}}$, $U_{L_{0,1}} = U^t_{L_{0,1}} + W^t_{L_{0,1}}$

5. Correct the velocity using an implicit treatment of the penalization term

$$\frac{U^{n+1} - U_{col}}{\Delta t} = \frac{1}{\lambda} \chi_{L_{0,1}} (U_{L_{0,1}} - U^{n+1}),$$

6. Transport the solid bodies with the fluid velocity or the rigid velocity $\phi^{n+1} = \phi^n - \Delta t U^{n+1} \cdot \nabla \phi^n$,

7. Redefine the distance function $\varphi_1^{n+1} = |\phi^{n+1}|$,

   Redefine L_0^{n+1},L_1^{n+1} using ϕ^{n+1}

   Perform the multi label fast marching method described in Chapter 3

8. Redefine the gravity centers as:

$$x_g(L_{0,1}) = \frac{\int_{\Omega_{L_{0,1}}} \rho X}{\int_{\Omega_{L_{0,1}}} \rho} \tag{5.24}$$

The implicit treatment of the penalization term allows to use a large penalization coefficient λ. Moreover with this algorithm, the incompressibility constraint is imposed before the rigidity constraint. The two constraints are better imposed as:

$$D(U) = 0 \Rightarrow \operatorname{div} U = 0$$

Algorithm 2

The second algorithm consists in reconstructing the N level set functions using 5.16. Then, the steps 6 and 7 of the previous algorithm are modified as follows:

6. Transport the solid bodies using (5.16).

7. Redefine the label maps and distance functions:

$$\forall x \in \Omega,\ \forall i \in \{1, ..., N+1\},\quad \begin{cases} L_0^{n+1}(x) &= i \text{ if } \phi_i^{n+1} \leq 0 \\ L_1^{n+1}(x) &= \underset{j \neq L_0^{n+1}(x)}{\arg\min}\ \phi_j^{n+1} \\ L_2^{n+1}(x) &= \underset{j \notin \{L_0^{n+1}(x), L_1^{n+1}(x)\}}{\arg\min}\ \phi_j^{n+1} \end{cases}$$

$$\forall x \in \Omega,\quad \begin{cases} \varphi_1^{n+1}(x) &= d(x, \Gamma_{L_1^{n+1}(x)}) \\ \varphi_2^{n+1}(x) &= d(x, \Gamma_{L_2^{n+1}(x)}) \end{cases}$$

Comparison of the two algorithms of evolutions

For circular rigid bodies, we compare the CPU time of both algorithms. In algorithm 1, the transport of the solid bodies with the help of one level set function is achieved using a WENO5 scheme whereas in algorithm 2it is performed by direct explicit transport of the center of the bodies. The redefinition of the label maps is performed either with a multi label fast marching algorithm on a thin band around the particles or using N signed distance functions that are updated directly thanks to the transport of the N gravity centers. The narrow-band sizes are $NB_{L_1} = 10h$ and $NB_{L_2} = 5h$ where h is the space discretization step.
To compare the CPU time of both algorithms, we fill the computational domain with particles next to each other so that the computation of L_2 is active as soon as there are two particles in the domain. Thus, the CPU time provided for the Multi Label fast marching is the worst CPU time one can obtain according to the number of particles.

Table 5.1 shows the CPU time of both algorithms, we can see that the CPU time of algorithm 2 is faster. Nevertheless, the difference is small and during numerical simulations the CPU time of the Algorithm 1 will be lower as the particles can be far from each other. Moreover, the Algorithm 2 is provided in the case of circular rigid disks, no interpolations of the initial level set functions are required. For general shape of bodies, interpolations of the N initial level set functions have to be achieved, the resulting CPU time will be higher. It seems most efficient to use the algorithm 1 for large number of particles.

Number of disks	Algorithm 1 CPU time	Algorithm 2 CPU time
10	0.1	0.05
50	0.5	0.6
100	1.0	0.7
200	2.1	1.4
400	4.0	2.8

Table 5.1: Computational time of both algorithms.

5.2 Sensitivity to grid resolution

A grid convergence study is carried out in dimension two and three.

In dimension two, the grid convergence is performed on four grid levels (G_1, G_2, G_3,G_4) which contain respectively : (128 $\times$ 128), (256 $\times$ 256), (512 $\times$ 512) and (1024 $\times$ 1024) cells on a uniform mesh. The regularization parameter ε is fixed to Δx^{G_1} where Δx^{G_1} denotes the mesh size corresponding to the coarsest grid (128 $\times$ 128).
Three test cases are investigated, the first one deals with the sedimentation of a 2D cylinder in an infinite canal. Then, we consider eight circular bodies falling under gravity and we present the grid convergence in the case of a constant intensity of the repulsive force (k is a constant) and with a repulsive force depending on the relative velocities. Finally, a qualitative grid convergence is presented for the sedimentation of 25 particles with a constant intensity of the short-range repulsive force. For all these simulations, the density of the fluid is set to 1, the dynamic viscosity $\mu = 0.01$, the gravity force $g = -980$ and the interfaces are regularized on thickness $\varepsilon = 1/128$, the rebound parameter is fixed to $\varepsilon_b = \varepsilon$.

The grid sensitivity in dimension three is achieved for a grid step h which varies from 0.015625 to 0.0039 and for the sedimentation of two hundred spherical rigid bodies.

5.2.1 Sedimentation of a 2D cylinder on a plane wall

We consider here the case of a 2D cylinder falling in an infinite canal and hitting a flat plane. The parameters used are the same as the ones used in [67]. The computational domain Ω is a square cavity of size $[0, 2] \times [0, 6]$, the density of the solid body ρ_s is set to 1.5. The particle of radii 0.125 is initially located at $(1, 4)$. The repulsive coefficient k is fixed to 100.
We show on Figure 5.3 the results obtained for two different resolutions $h = 1/256$ and $h = 1/512$, we represent the trajectory of the y coordinate of the gravity center (top) and the vertical velocity of the rigid body (bottom). We observe that the particle falls fastly until it approaches the wall and it hits the bottom, then it bounces on the wall and finally reachs a steady state.

For the two different grids, the results are almost similar. These results match almost perfectly the one obtained in [67] and [31] for which two resolutions $h = 1/256$ and $h = 1/384$ of this benchmark were presented. The only difference comes from the minimal distance between the wall and the particle which is smaller here and thus the rebound time is shifted compared to their simulations.

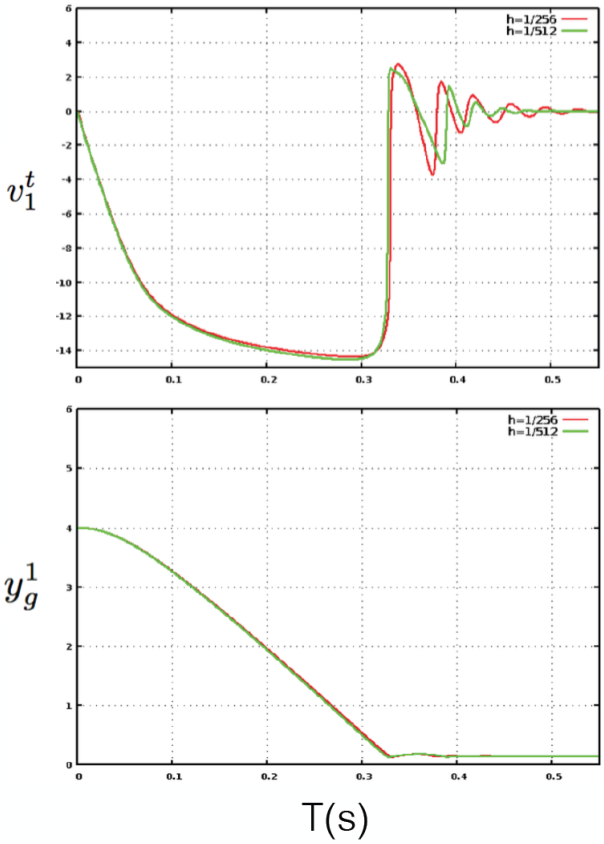


Figure 5.3: Simulation of a 2D cylinder falling on a flat plane.

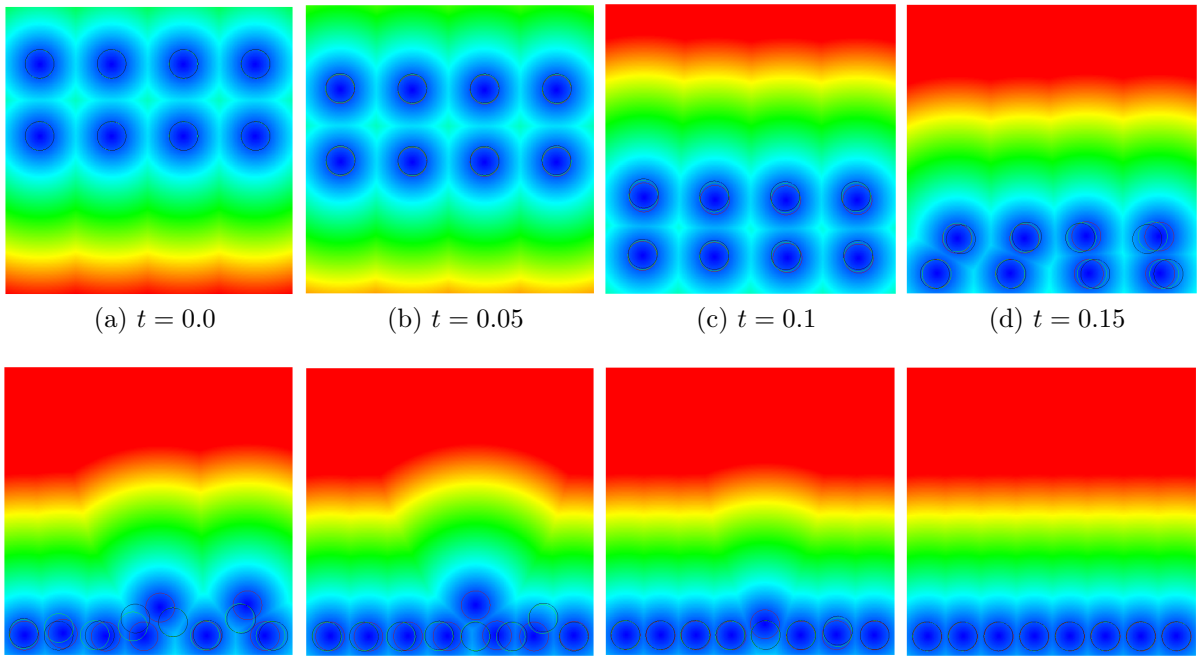
5.2.2 Collision model with constant repulsive intensity

We investigate here the grid sensitivity of our model in the case of multiple immersed particles. The computational domain is a square $\Omega = [0, 1]^2$. The density of the rigid particles is set to $\rho_s = 2$. The repulsive coefficient is fixed to $k = 100$ for all the bodies. The same coefficient is chosen for the repulsive forces exerted by the four wall $k_{\text{wall}} = 100$. The repulsive forces are applied on a ring around the interface of thickness $\varepsilon = \Delta x^{G_1}$. Two test cases are explored: the sedimentation of eight rigid disks of radii $r = 0.05$ and 25 circular rigid bodies having the same radius $R = 0.025 + \varepsilon$.

Results of the first benchmark are presented on Figure 5.4. Figures 5.5 and 5.6 show the time history of the trajectory and velocity of the first particle. Before the collision time $t \simeq 0.1$, a grid convergence is achieved as the three finer grids provide similar results. Then, we observe the well-known phenomena of drafting, kissing and tumbling [58]. Once collisions occur, one can not obtain similar results for the different resolutions, we should note however that compared to the coarsest grids, the two finer grids seems to produce results close.

The second simulation which deals with the sedimentation of 25 particles is represented on Figure 5.7, the white line corresponds to the numerical size of the rigid particles ($\phi = \varepsilon$). At initialisation, the spheres are released from rest and fall because of gravity. As time proceeds, the distance between spheres decreases until collision occurs that is kissing process. The kissing phenomenon persists and then we can observe a tumbling stage . The dynamic of the rigid bodies and the interaction between them is quite similar for the different resolutions. By $t = 0.48$, all simulations have reached static equilibrium which

represents different local minimum of the sedimentation of the 25 disks. We observe that the two fine simulations are very close to each other until time $t = 0.15$ and keep a symmetrical distribution. At the end of the simulation, a packing of the bodies is formed at the bottom of the computational domain and is composed of three layers. Each layer contains the same number of structures for the four different grids: eleven bodies on the first one, ten bodies on the second one and four on the last one. The distribution of the four bodies of the third layer is different for the various grids but is much closer on the two finest simulations. From these two test cases, one can conclude that the grid convergence is achieved for the grid G_{512}, the corresponding discretization space step is $h \simeq 1.95.10^{-3}$.



(a) $t = 0.0$ (b) $t = 0.05$ (c) $t = 0.1$ (d) $t = 0.15$

(e) $t = 0.20$ (f) $t = 0.25$ (g) $t = 0.30$ (h) $t = 0.35$

Figure 5.4: Study of the grid convergence with a test case of 8 rigid disks falling under gravity, the repulsive coefficient $k = 100$. The red contour corresponds to the grid G_{256}, the green contour to G_{512} and the black contour to the grid G_{1024}.

5.2.3 Collision model depending on the relative velocities

We investigate here the grid convergence in case in which the collision model depends on the relative velocities of the closest particles . We consider again the test case of eight disks of radius $R = 0.05$ falling under gravity force $g = -980$.
We denote by $D_{\min}(i, j)$ the minimal distance between the ith and jth particle computed by means of the distance function φ_2 and by $K(i, j)$ the intensity of the repulsive force. When the particles get close from each other,$D_{\min}(i, j) \leq 2\varepsilon$, the repulsive coefficient $k(i, j)$ is fixed until the distance between the particles is higher than a certain value, here we take $D_{\min}(i, j) \leq 2\varepsilon + 4\Delta x^{G_1}$. When the relative velocities are close to zero, the intensity of the force is fixed.

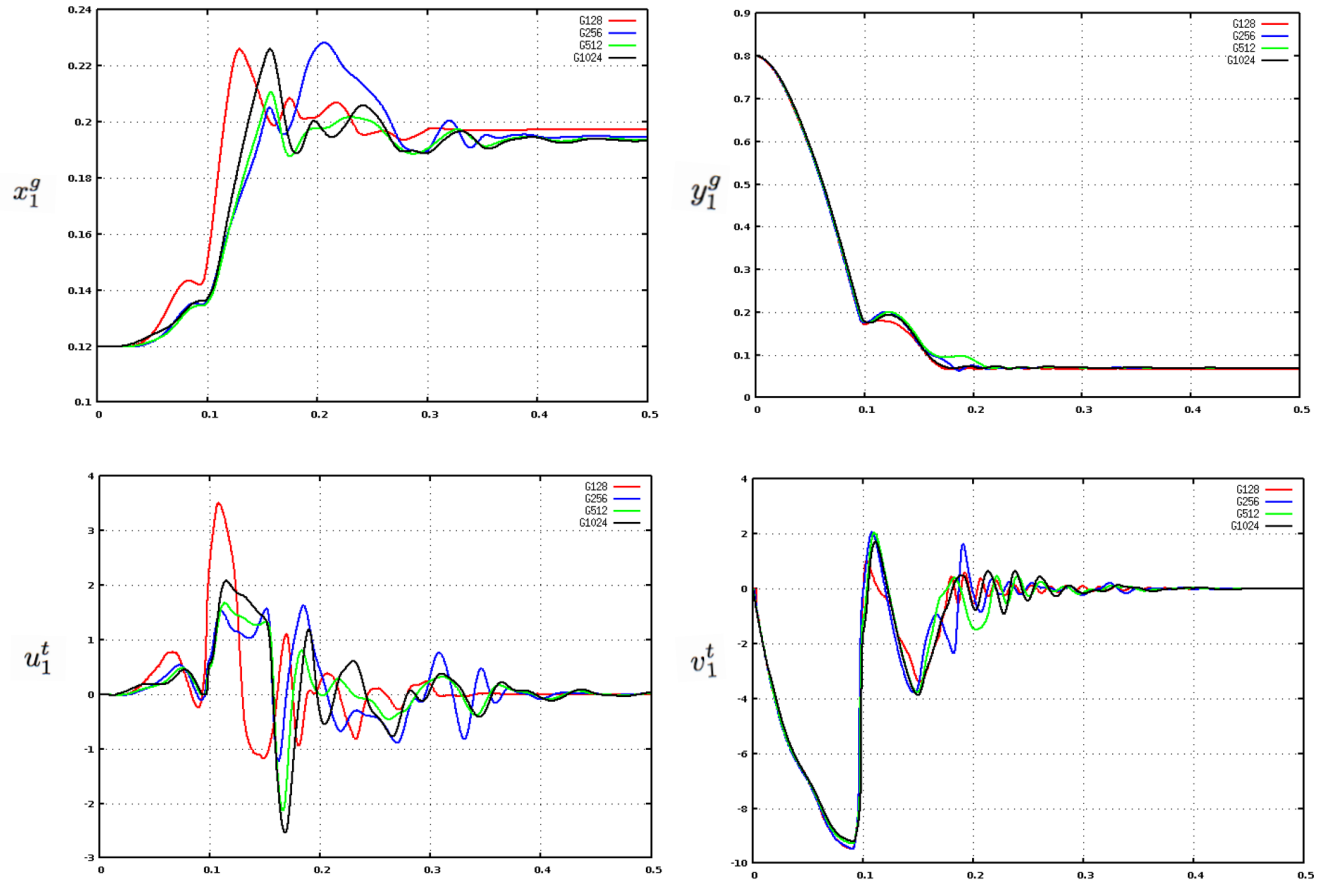


Figure 5.5: Study of the grid convergence with a test case of 8 rigid disks falling under gravity, the repulsive coefficient $k = 100$. Time history of the trajectory and velocity of the first particle.

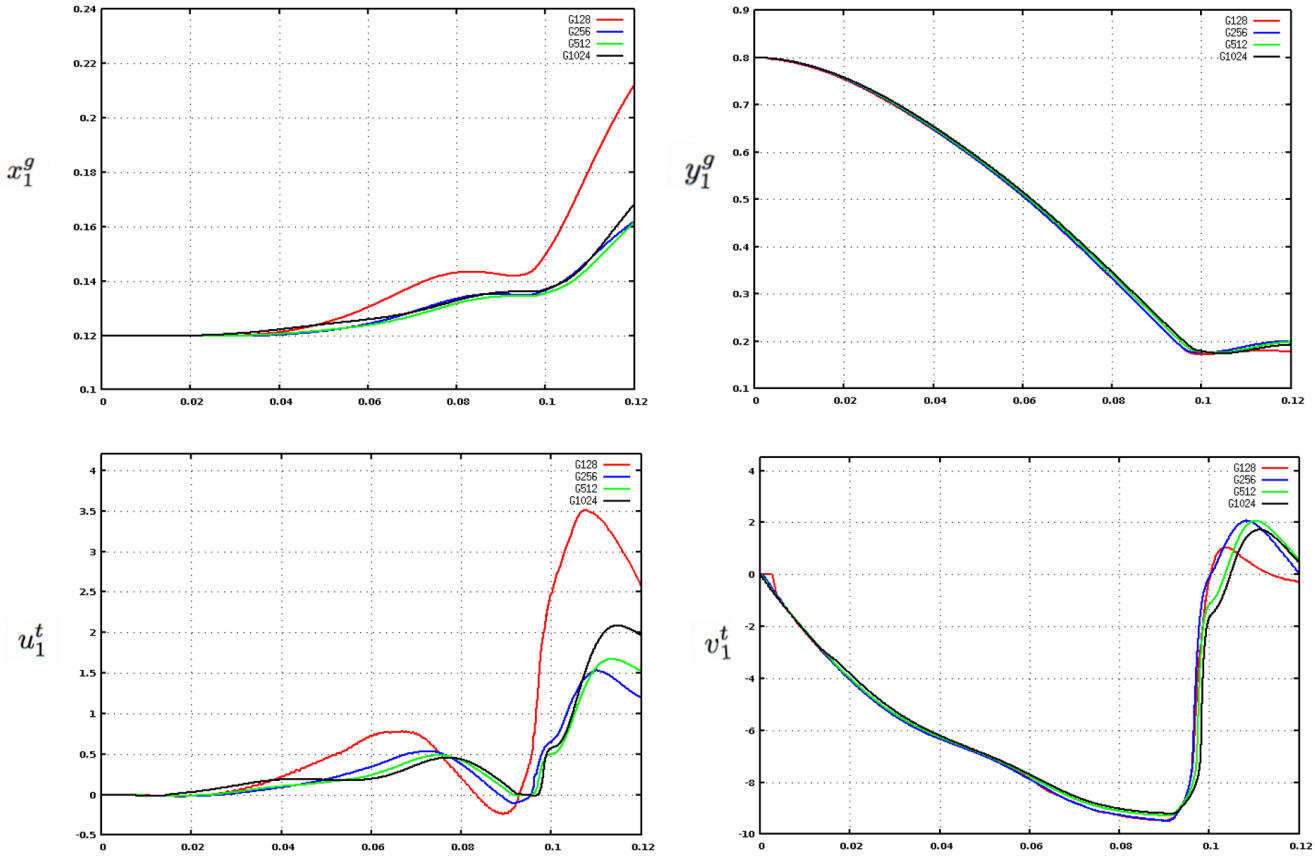


Figure 5.6: Study of the grid convergence with a test case of 8 rigid disks falling under gravity, the repulsive coefficient $k = 100$. Time history of the trajectory and velocity of the first particle just before the collision time $t = 0.1$.

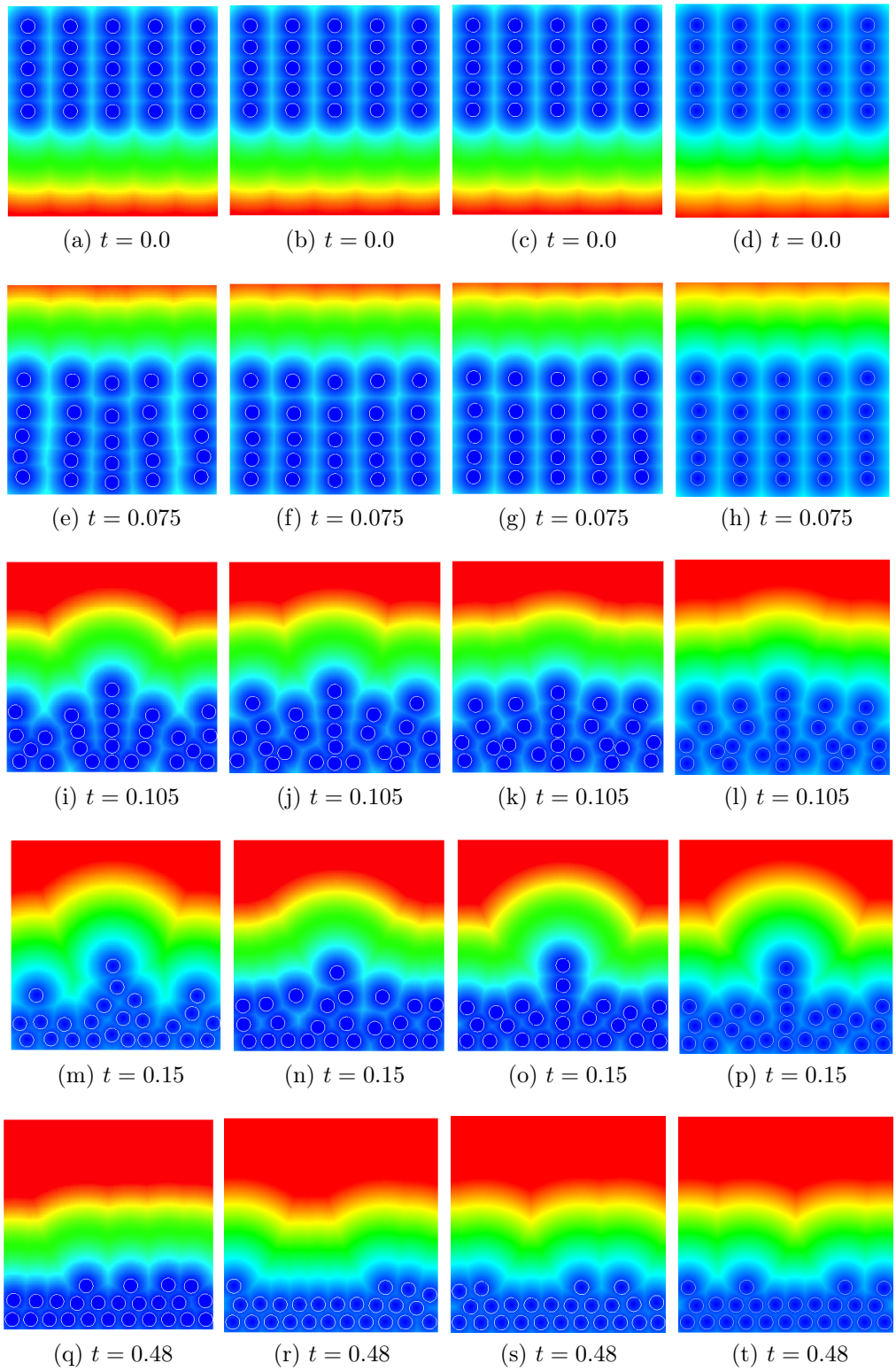


Figure 5.7: Study of the grid convergence with a test case of 25 rigid disks falling under gravity. From left to right, grid 128×128, grid 256×256, grid 512×512 and grid 1024×1024. The background color shows the level set amplitude.

In Figure 5.8, we represent the vertical trajectory of the first and fifth particle obtained for the three resolutions $G_{256}, G_{512}, G_{1024}$. The particle one is located in the first layer (from top to bottom) and is the first one starting from left, the $6th$ body is part of the second layer and corresponds to the first one starting from left. In Figure 5.9, we represent the minimal distance between these particles and the associated repulsive coefficient.
We can see, that the repulsive coefficient for the resolution $h = 1/256$ is higher than the others. Moreover, we observe several rebounds for this resolution, the particles move away and approach each other several times.
On the contrary, for G_{512} and G_{1024}, only one high rebound is observed and then the particles remain close from each other until the end of the simulation.
We also represent the results of the simulation on Figure 5.10. The grid convergence is clearly achieved for the resolution G_{512}.

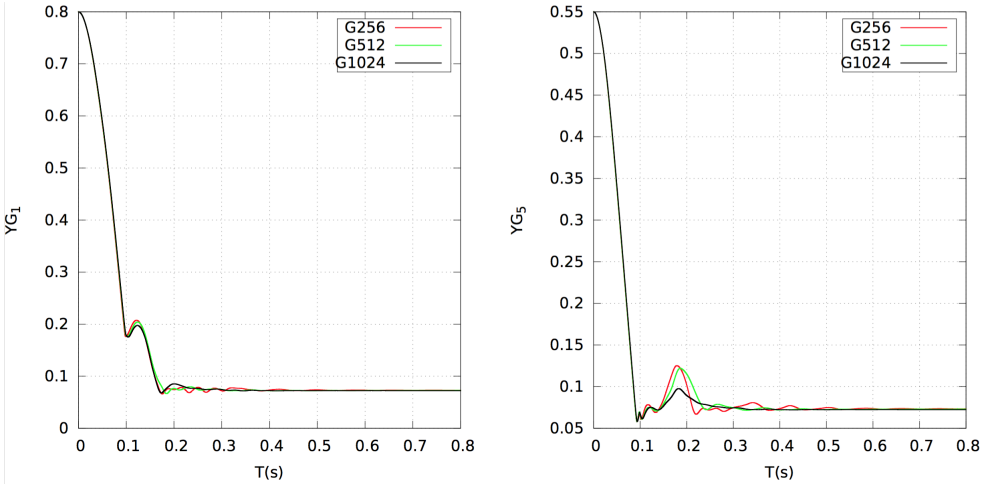


Figure 5.8: Trajectory of the gravity center (y coordinate) of the first and fifth particle.

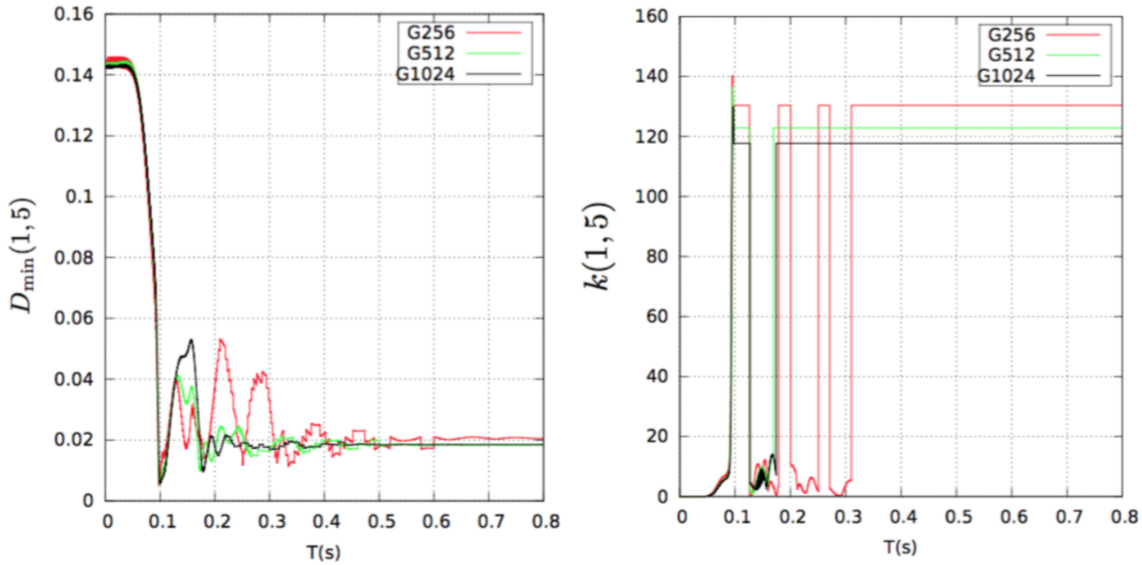
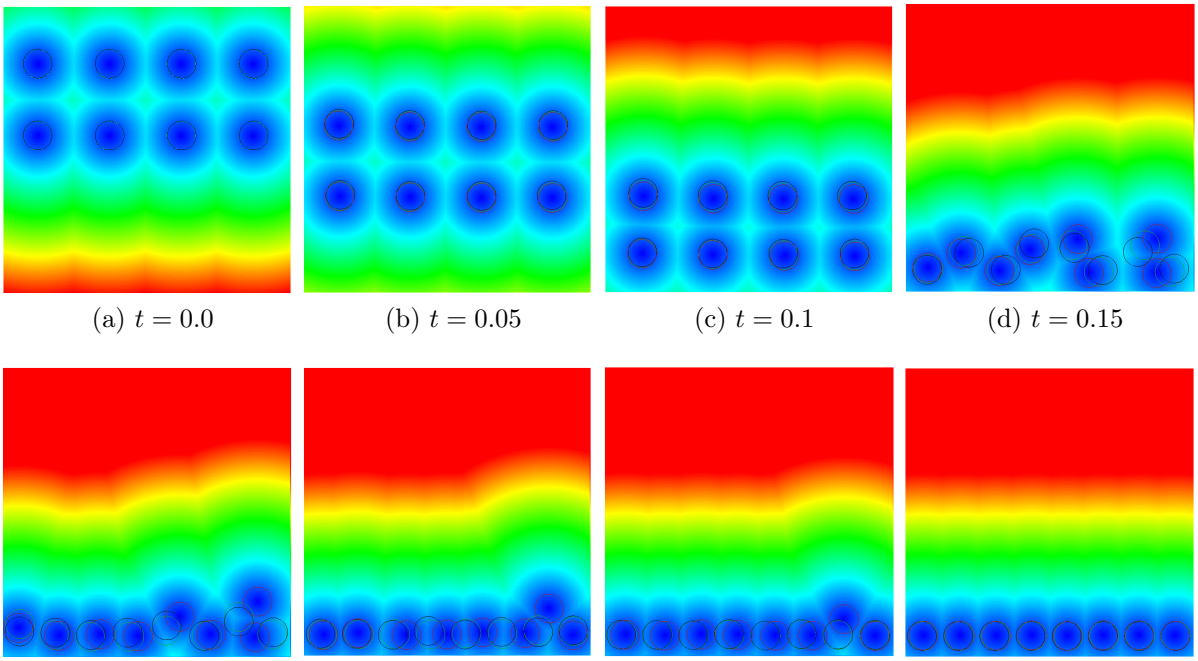


Figure 5.9: Minimal distance between the first and fifth particle (top) and the associated repulsive coefficient (bottom) which depends on the relative velocities.



(a) $t = 0.0$ (b) $t = 0.05$ (c) $t = 0.1$ (d) $t = 0.15$

(e) $t = 0.2$ (f) $t = 0.25$ (g) $t = 0.3$ (h) $t = 0.4$

Figure 5.10: Study of the grid convergence with a test case of 8 rigid disks falling under gravity, the intensity of the force is proportional to the relative velocity. The red contour corresponds to the grid G_{256}, the green contour to G_{512} and the black contour to the grid G_{1024}.

5.2.4 Grid convergence 3D

We consider a computational domain $\Omega = [0, 1]^3$ and we study the grid sensitivity using three different grids which contain respectively : (64^3), (128^3) and (256^3) cells on a uniform mesh. The half thickness of the interface is $\varepsilon = 2\Delta x$. The coefficient of gravity g is set to -980. The repulsive coefficients are: $k_x = -g/10$, $k_y = -g/10$, $k_z = -g/10$, $k_x^{\text{wall}} = -g/40$, $k_y^{\text{wall}} = -g/40$, $k_z^{\text{wall}} = -g$. The first simulation deals with the sedimentation of a sphere falling on a plane wall. At initialization the gravity center of the sphere is located $(0.5, 0.8, 0.5)$. Figure 5.11 shows the time history of the vertical trajectory of the gravity center and the vertical velocity performed with the three different grids, we can see that for the two finer grids provides similar results, the sphere hits the bottom wall at the same time, then the sphere rebounds on the wall, the obtained results for these two resolutions are close before the second rebound, due to instabilities we observe thereafter a delay between the two results.

Figures 5.12, 5.13 and 5.14 show the simulation of 200 rigid spheres of radius $R = 0.01$ falling under gravity. The results obtained with the finest grids are almost similar, the particles fall down symmetrically no interaction occurs between the particles before hitting the bottom wall. On the contrary, with the coarser grid, the particles interacts and falls unsymmetrically, this is due to the fact that the short range repulsive forces are active from the beginning as there are less discretization points in the interstitial gaps between the particles.
Thus a 3D qualitative grid convergence is achieved for a large number of particles.

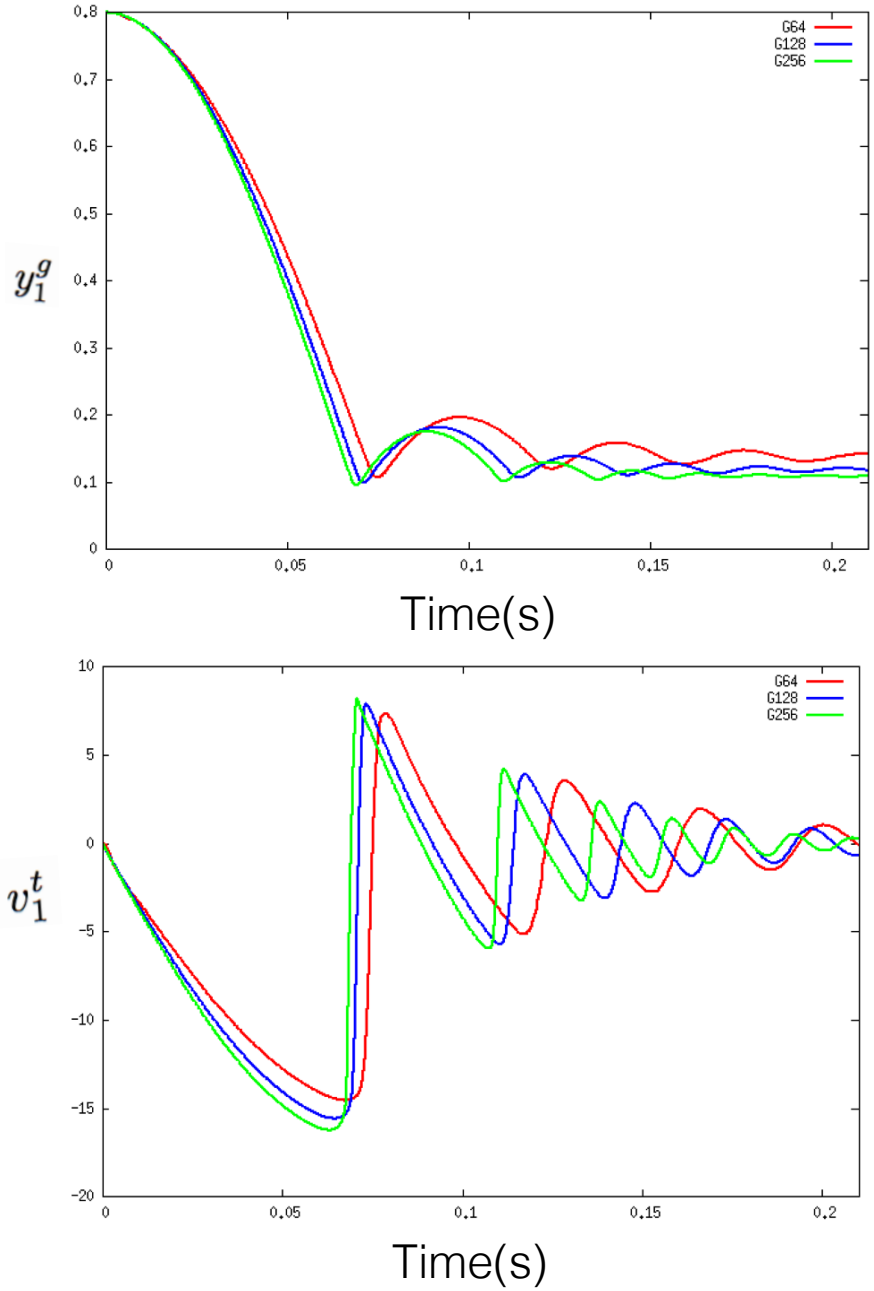
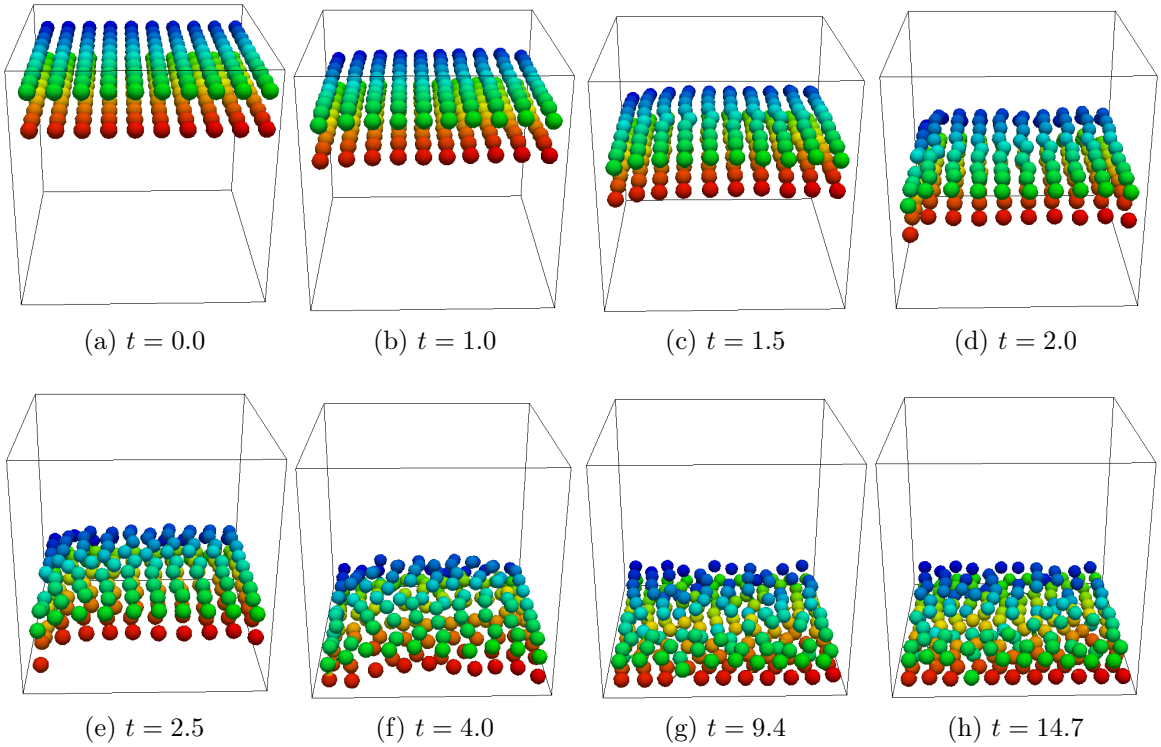


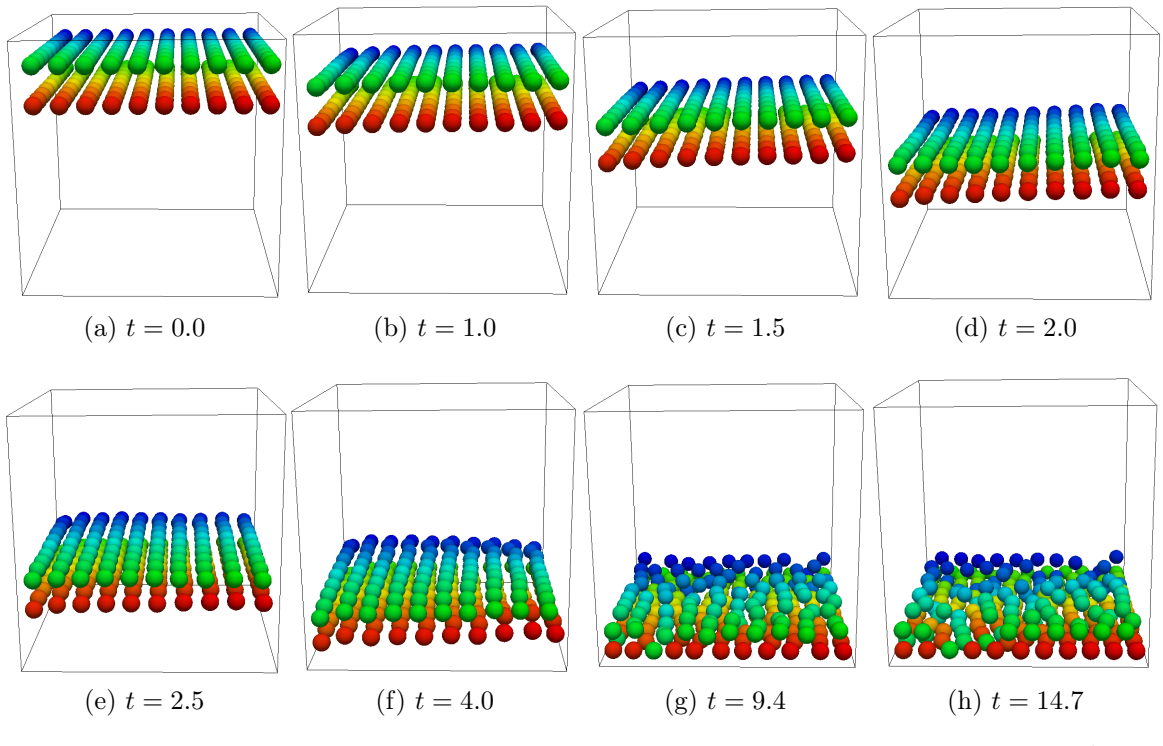
Figure 5.11: Vertical trajectory and velocity of a sphere falling on a plane wall obtained with three different resolutions.



(a) $t = 0.0$ (b) $t = 1.0$ (c) $t = 1.5$ (d) $t = 2.0$

(e) $t = 2.5$ (f) $t = 4.0$ (g) $t = 9.4$ (h) $t = 14.7$

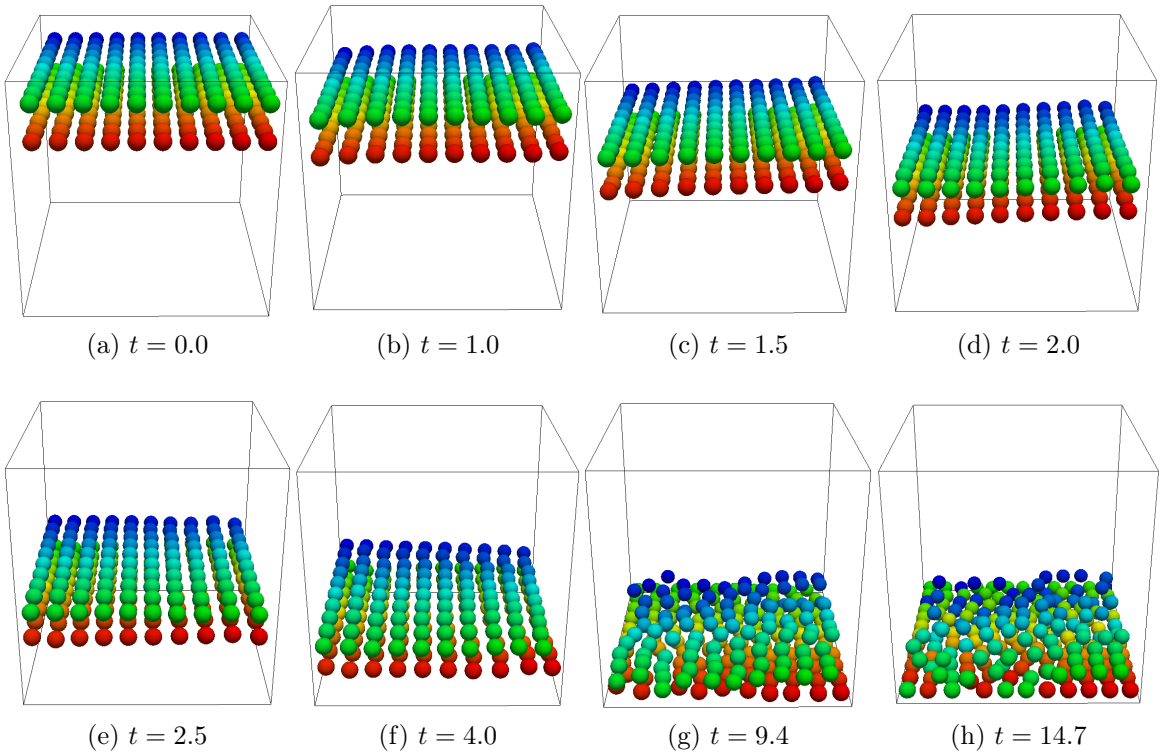
Figure 5.12: Simulation of 200 rigid spheres subject to gravity (grid resolution size 64^3). The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 200^{th} body and red for the fluid that is the 201^{th} object.



(a) $t = 0.0$ (b) $t = 1.0$ (c) $t = 1.5$ (d) $t = 2.0$

(e) $t = 2.5$ (f) $t = 4.0$ (g) $t = 9.4$ (h) $t = 14.7$

Figure 5.13: Simulation of 200 rigid spheres subject to gravity (grid resolution size 128^3). The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 200^{th} body and red for the fluid that is the 201^{th} object.



(a) $t = 0.0$ (b) $t = 1.0$ (c) $t = 1.5$ (d) $t = 2.0$

(e) $t = 2.5$ (f) $t = 4.0$ (g) $t = 9.4$ (h) $t = 14.7$

Figure 5.14: Simulation of 200 rigid spheres subject to gravity (grid resolution size 256^3). The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 200^{th} body and red for the fluid that is the 201^{th} object.

5.3 Comparison of the method with level set decomposition

We provide here a comparison of our penalization model (5.17) and the penalization model that uses a level set decomposition (5.11).

5.3.1 Computational time using the algorithm (2) for rigid disks

We give a comparison of the computational time in the case of N rigid disks. Instead of transporting one level set function and performing the multi label fast marching method, we transport the gravity centers of the N rigid structures and reconstruct their associated level set functions. Then, the label and distance functions are reinitialized by using their definition. It is most convenient to use this algorithm in order to compare the saving computational time which is induced by changing the penalization model (3.12) and our proposed penalization term. We average the computational time on the ten first iterations. The averaged CPU time of our algorithm (table 2) is compared to the method using N level set functions (table 1), according to the number of cells. As noticed before the collision model (3.12) computes N^2 repulsive forces which induced a high computational cost as shown in the second column of table 1 whereas in the present algorithm, the CPU time of the collision model is constant as it does not depend on the number of cells. The CPU time of the penalization model is larger in (5.11) because it depends on the number of cells. Indeed, N rigid velocities must be computed to get the right velocity of each cell. In the present algorithm, we must add the label redefinition that depends almost linearly

on the number of cells and so is quite cheap. For a low number of cells the CPU time of both methods is close but increasing this number from 2 to 400 cells induces a total CPU time 8000 larger for the model (5.11) whereas with the present model the CPU time is only 25 times larger. The proposed model is hence around 340 times faster for 400 cells.

Number of disks	Collision model (3.12) CPU time	Penalization model (5.11) CPU time	Total CPU time
2	0.02	0.06	0.2
5	0.17	0.16	0.48
10	0.72	0.35	1.24
25	4.87	0.88	6
50	19.25	1.75	21.5
100	80.8	3.9	85.3
400	1583.4	19.75	1605.3

Table 5.2: Averaged CPU time using the N level set decomposition

Number of disks	Model (3.16) CPU time	Model (5.17) CPU time	Label redefinition CPU time	Total CPU time
2	0.015	0.05	0.008	0.2
5	0.015	0.06	0.014	0.23
10	0.015	0.09	0.02	0.25
25	0.016	0.18	0.08	0.4
50	0.016	0.3	0.16	0.6
100	0.016	0.56	0.23	0.9
400	0.016	2.52	2.06	4.7

Table 5.3: Averaged CPU time using the algorithm of section 5.1.4

5.3.2 Numerical comparison of the two collision models

We provide here a comparison of the two collision model in the case of a constant intensity of the repulsive forces to validate the proposition 1 presented in Chapter 3. This proposition is also avalaible for $p = 2$, so that L^2 norm can be used. To highlight the differences between the two collision models we first focus on a a test case with three circular rigid bodies falling on each other. The simulations are performed on a grid of size (128×128) corresponding to a space step $\Delta x = 7.8125.10^{-2}$ in order to better see the difference between the two models. The bodies have the same radius $R = 0.1$ and the thickness of the interface is $\varepsilon = 2\Delta x$. The repulsive coefficients exerted by the wall are: $k_x = -g/10$, $k_y = -g/10$, $k_x^{\text{wall}} = -g/40$, $k_y^{\text{wall}} = -g$.

On Figure 5.16, black and white lines represents the interfaces of the three different objects. The black line stands for the collision model (3.12) and the white line for the collision model (3.16) corresponding also to the colors that represent the values of the level set function. We can see that the bodies have the same behaviour, as expected, because on the one hand the radius is large and on the other hand the forces applied on the bodies are very similar. So the difference is small. The Figure 5.17 shows the results obtained with 6 rigid bodies. here we can see again that the behaviour is very similar for

a large radius. The last test concerns the same configuration with six smaller disks with radius $R = 0.03$ leading to a higher difference in the dynamics even if the final state is the same. The difference between the two models is stronger when the number of body is larger or when the force coefficients are higher. The bottom plot in Figure 5.19 shows the vorticity inside the fluid domain, when the bodies reached the bottom they move to the right and induce a strong positive vortex that has a strong influence on the dynamics of the bodies in its turn. We observe that the vorticity increases when the bodies are close to each other as the repulsive force is higher. At time $t = 0.5$ the vorticity vanishes because the six rigid bodies have reached the static equilibrium.

We represent on Figure 5.15, the estimation of the error $\frac{\|F_{\text{global}} - F_{\text{label}}\|_{L^2(\Omega)}}{\|F_{\text{label}}\|_{L^2(\Omega)}}$ for different number of particles. Then, we take the case of three particles and we vary their radius, the results are presented in Table 5.4. These results validate numerically the Proposition 1 that is the error estimation depends on the number and the size of the particles. The larger are the particles the smaller will be the difference between the two collision models.

Radius of disks	Error estimation
1.10^{-2}	$6.8.10^{-2}$
$2.5.10^{-2}$	1.10^{-5}
5.10^{-2}	2.10^{-9}
1.10^{-1}	6.10^{-16}

Table 5.4: Error estimation $\frac{\|F_{\text{global}} - F_{\text{label}}\|_{L^2(\Omega)}}{\|F_{\text{label}}\|_{L^2(\Omega)}}$ for three particles according to their radius.

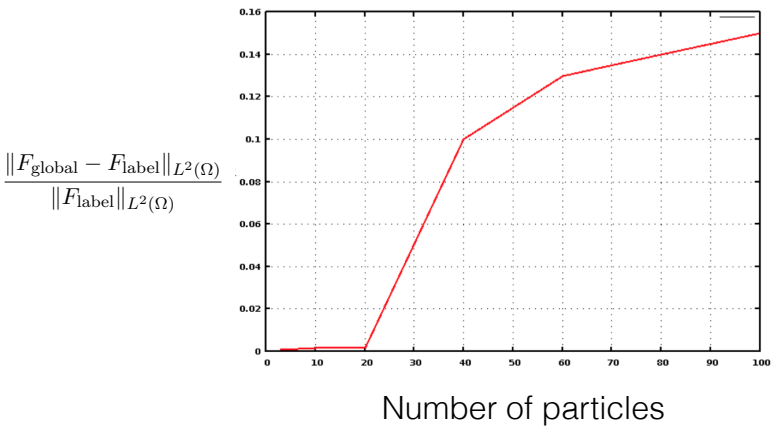
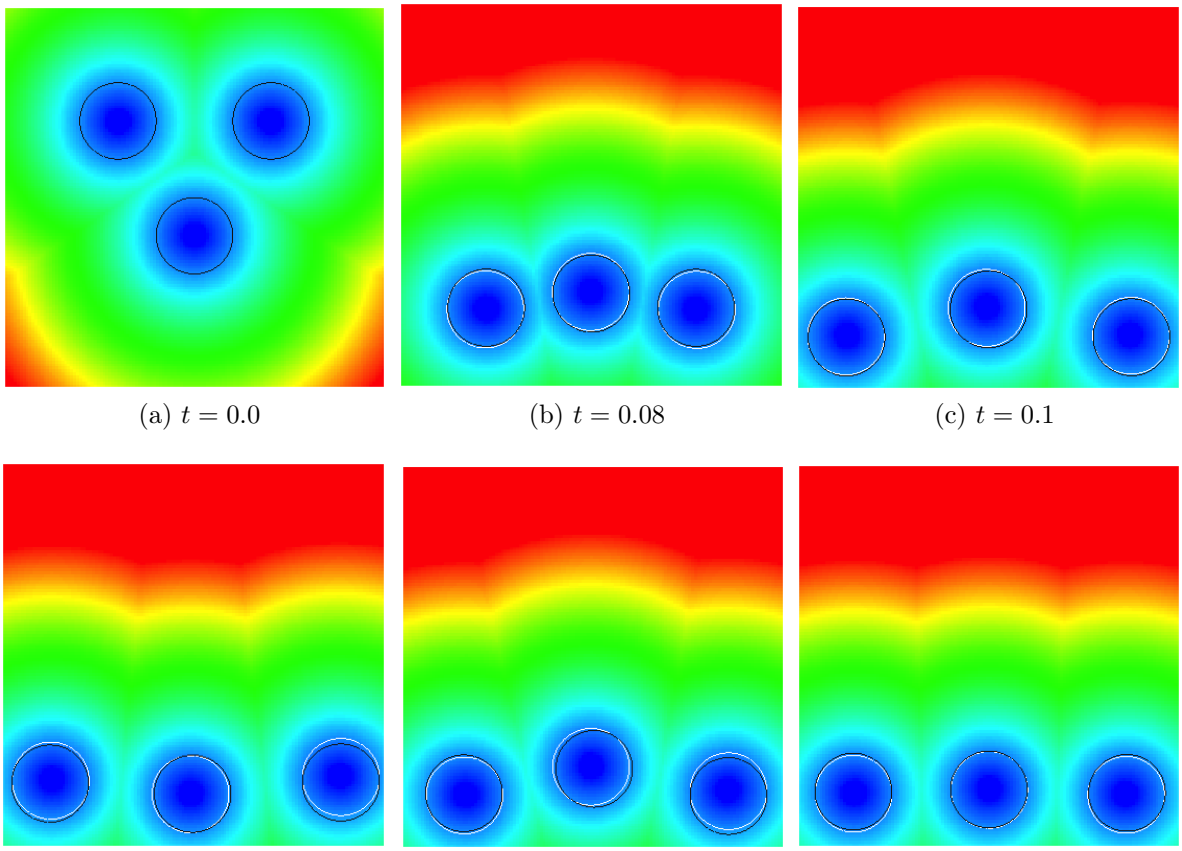


Figure 5.15: Error estimation $\frac{\|F_{\text{global}} - F_{\text{label}}\|_{L^2(\Omega)}}{\|F_{\text{label}}\|_{L^2(\Omega)}}$ according to the number of particles N.

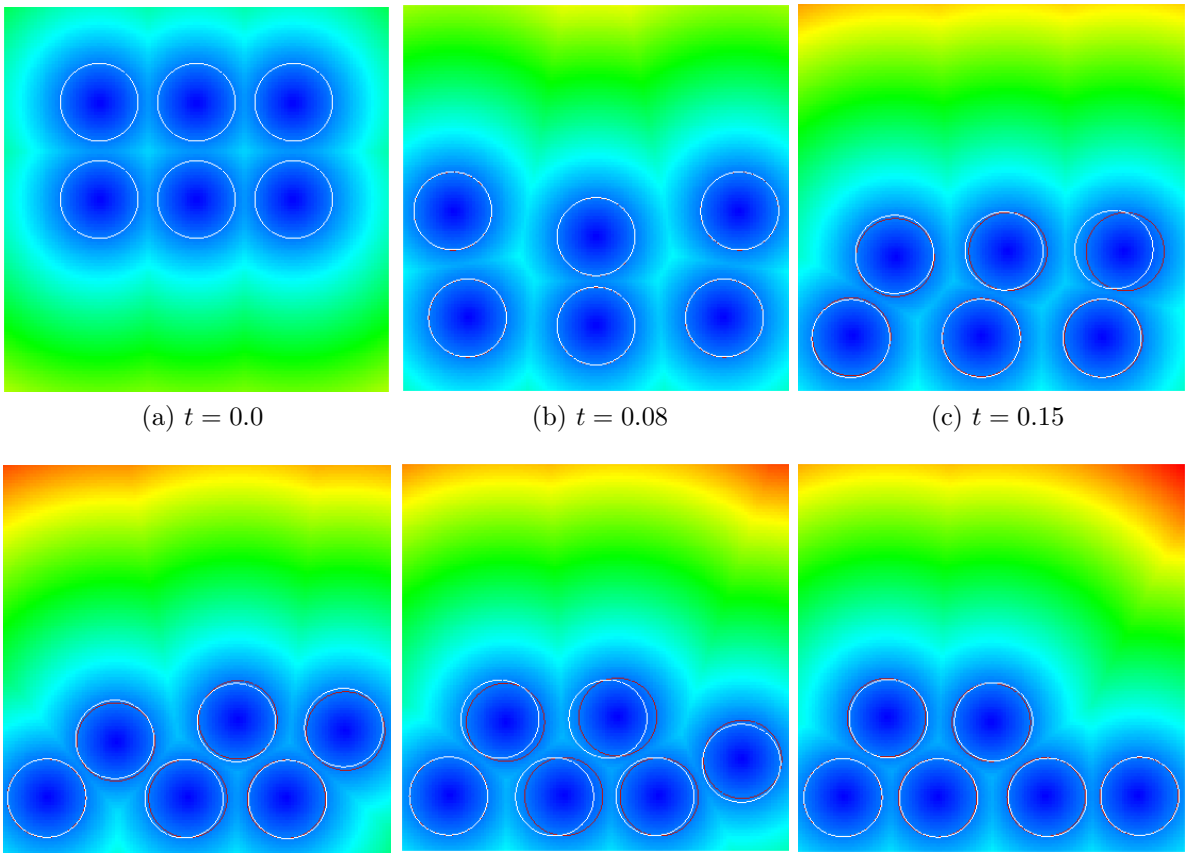
A scheme for inelastic collisions is implemented imposing a minimal distance between bodies



(a) $t = 0.0$ (b) $t = 0.08$ (c) $t = 0.1$

(d) $t = 0.12$ (e) $t = 0.15$ (f) $t = 0.23$

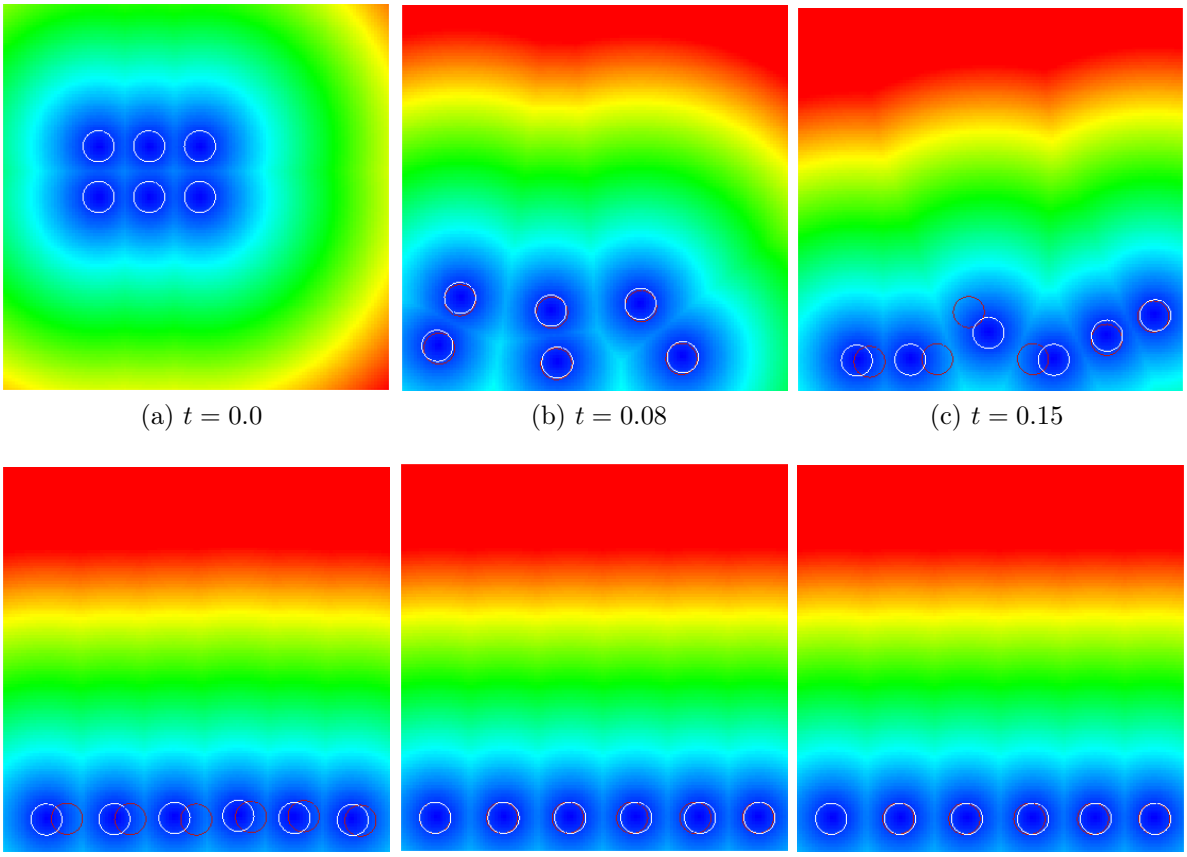
Figure 5.16: Comparison of the two collision models for three disks of radius $R = 0.1$. The background colors show the level set amplitude.



(a) $t = 0.0$ (b) $t = 0.08$ (c) $t = 0.15$

(d) $t = 0.25$ (e) $t = 0.38$ (f) $t = 0.50$

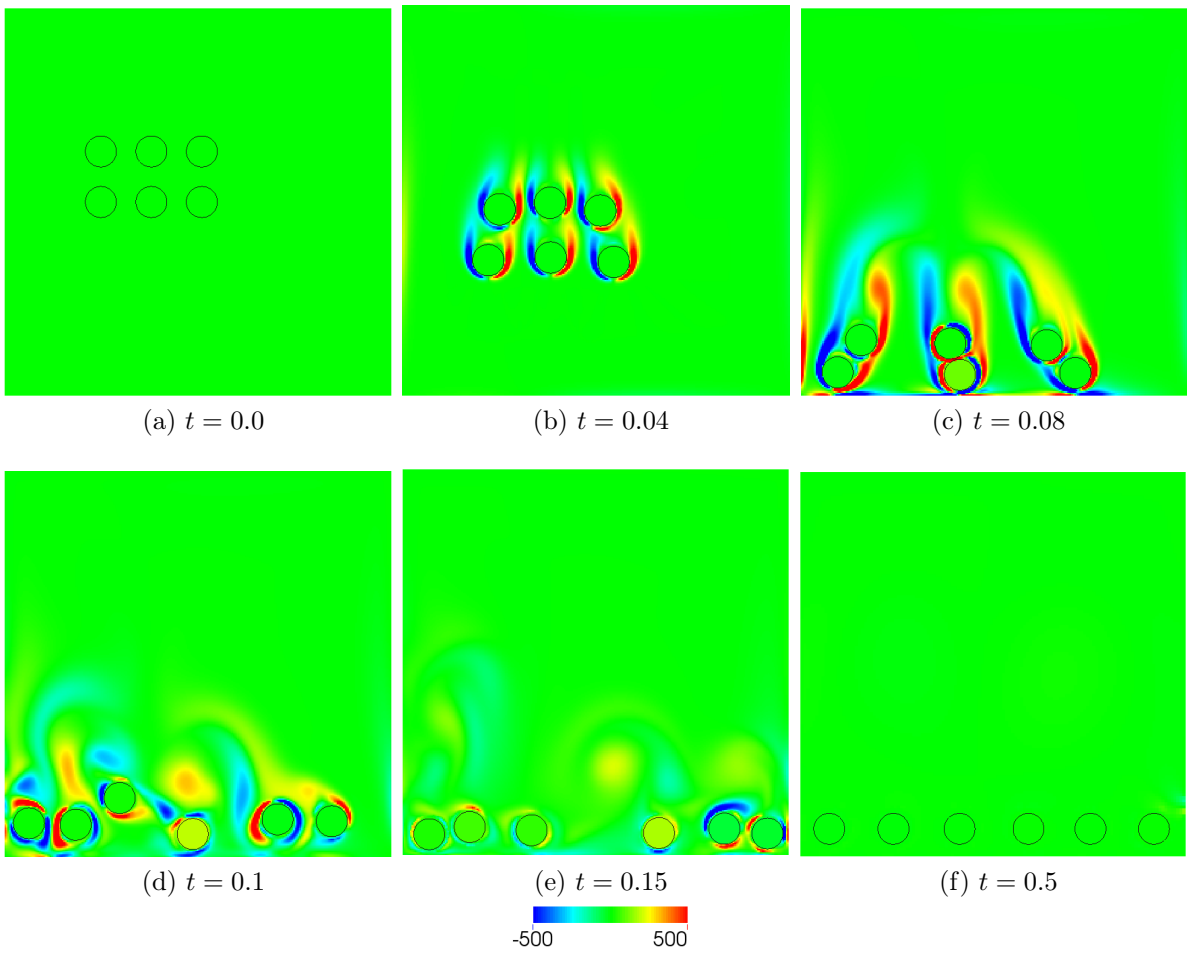
Figure 5.17: Comparison of the two collision models for six disks of radius $R = 0.1$.The background colors show the level set amplitude.



(a) $t = 0.0$ (b) $t = 0.08$ (c) $t = 0.15$

(d) $t = 0.25$ (e) $t = 0.38$ (f) $t = 0.5$

Figure 5.18: Comparison of the two collision models for six disks of radius $R = 0.03$. The background colors show the level set amplitude.



(a) $t = 0.0$ (b) $t = 0.04$ (c) $t = 0.08$

(d) $t = 0.1$ (e) $t = 0.15$ (f) $t = 0.5$

Figure 5.19: Six disks falling under gravity, colors represents the values of the vorticity field.

5.4 Comparison of the collision model with the algorithm to handle contacts introduced in [105]

In this part, we compare qualitatively our model to an existing method proposed in [105]. We first describe briefly this method, then a numerical comparison on two test cases are presented.

5.4.1 Description of the model

In that model, the solid bodies are taking into account by penalizing the strain tensor to enforce the rigid body motion (see [88]).
A scheme for inelastic collisions is implemented imposing a minimal distance between bodies and therefore avoiding contacts between the particles and the four walls (see [124]). Only spherical particles are considered so that the projected velocities are the translational velocity of the particles. As the translational velocities are in the all computational domain, only a loop on the number of particles is required to update them.

The principle of the contact algorithm is to project the velocities onto a set of admissible velocities corresponding to the set of velocities for which the particles and the particles and the walls are not in contact.
The first step is to compute the signed distance functions between the particles and their gradient. Denoting by r_i and r_j the radius of the particles Ω_i and Ω_j and x_i and x_j their gravity centers, at each time step the signed distance function between Ω_i and Ω_j is defined by:

$$D_{i,j}(x^n) = |x_i^n - x_j^n| - r_i - r_j$$

Then, the gradient of this distance is given by:

$$\nabla D_{i,j}(x^n) = (..., 0, -e_{i,j}^n, ..., 0, e_{i,j}^n, 0, ..)$$

where the normal $e_{i,j}^n$ is defined by:

$$e_{i,j}^n = \frac{x_j^n - x_i^n}{|x_j^n - x_i^n|}$$

The signed distance functions are computed between pair-wise particles, so for N particles $\frac{N(N-1)}{2}$ distances and gradients are computed. The set of admissible velocities is defined as:

$$G(x^n) = \{\forall i < j, U \in R^{2n}, D_{i,j}(x^n) + \Delta t U \cdot \nabla D_{i,j}(x^n) \geq 0\}$$

If the velocities are projected onto this set, at the next time step, no overlapping can occur. Indeed, denoting by U_i and U_j the velocities of the particles Ω_i and Ω_j, the signed distance function $D_{i,j}$ is solution of the following transport equation:

$$\partial_t D_{i,j} + U \cdot \nabla Di, j = 0$$

Thus, if the contraint $G(x^n)$ is checked we have:

$$D_{i,j}^{n+1} \geq 0$$

In practice, it is better to define a minimal distance between the particles, which avoid the particles from being in contact, the constraint is changed as:

$$D_{i,j}(x^n) + \Delta t U \cdot G_{i,j}(x^n) \geq D_{\min}$$

giving:

$$D_{i,j}^{n+1} \geq D_{\min}$$

In order to project the velocities a priori U_*^n onto the set $G(x^n)$ a vector of lagrange multipliers is introduced. We denote by $\lambda \in R_+^{\frac{N(N-1)}{2}}$ the lagrange multipliers.

Then, the functional to be minimized is:

$$J(U, \Lambda) = \frac{1}{2}|U - U_*^n|^2 - \sum_{1 \leq i,j \leq N} \lambda_{i,j}(D_{i,j}(x^n) + \Delta t U \cdot \nabla D_{i,j}(x^n)) \tag{5.25}$$

Then, at each time step the saddle path problem resolved by an Uzawa algorithm is:

$$\text{Find } (U^n, \lambda^n) \in R^{2N} \times R_+^{\frac{N(N-1)}{2}} \text{ such as:}$$

$$\forall (U, \Lambda) \in R^{2N} \times R_+^{\frac{N(N-1)}{2}},\ J(U^n, \Lambda) \leq J(U^n, \lambda^n) \leq J(U, \lambda^n)$$

If there is no contact between the particles Ω_i and Ω_j, the lagrange multiplier $\lambda_{i,j} = 0$, and thus the associated velocity is not projected.
Finally, the new projected velocities U_i^n are used to transport the gravity centers as:

$$\forall i \in \{1, ..., N\},\ x_i^{n+1} = x_i^n + \Delta t U_i^n$$

5.4.2 Numerical comparison

Let us draw a comparison between the contact algorithm presented above and the proposed collision model in the case of a constant intensity of the force. The fluid flow is governed by the Navier Stokes equations with a dynamic viscosity $\mu = 0.01$ and density $\rho_f = 1$, the particles density is fixed to $\rho_s = 2$.
We first present a test case for which the contact algorithm is inserted in our code so that the only error comes from the procedure to deal with numerical contacts.

The test case consists in the sedimentation of eight disks. The gravity force is $g = -980$ and the radius are $R = 0.1$. The repulsive coefficient for the walls and the particle/particle contacts are fixed to 100. The computational domain $\Omega[0, 1] \times [0, 1]$ is discretized with a mesh grid which contains 512×512 nodes and the thickness of interface $\varepsilon = 2h$ where $h = 1/512$. For the contact algorithm of A.Lefebvre and B.Maury, we set the minimal distance $D_{min} = 2\varepsilon$.

The obtained simulations are presented in Figure 5.20 and 5.21. We observe that the dynamic of the rigid bodies is completely different with these two procedures.
In Figure 5.21, the particles fall until collisions occurs at time $t = 0.14$, then the top particles roll on the bottom particles to finally hit the bottom and rebound at time $t = 0.19$. At $t = 0.24$, all the particles are distributed on the bottom but keep moving until $t = 0.4$ where the particles have reached the steady state and are distributed symmetrically in

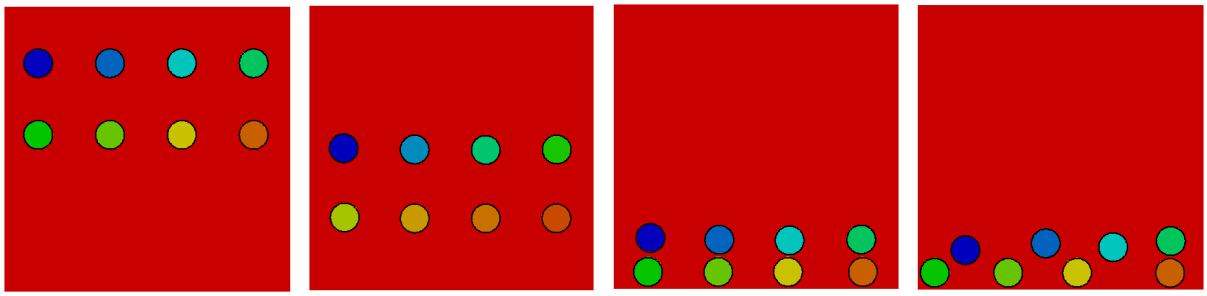
the computational domain. The kissing process is very fast as it only lasted around 0.5 seconds.
The results represented in Figure 5.20 are different from $t = 0.14$ when the particles collide. We observe the same kissing phenomenon, at $t = 0.17$ the top particles roll on the bottom ones except for the green particle located at right which stands on the orange rigid body. Then, instead of falling and stay close to the ground, we can see at $t = 0.2$ that the light blue particle rebounds on the wall until it hits the green particle located at right. There are more interactions between the particles, at $t = 0.25$ the blue, light blue, green and orange particles form a packing and move together to finally end up on the bottom wall. Then, the particles keep moving until the steady state is reached at $t = 0.6$. Consequently, the two simulations provides two completely different behavior of the particles. On one side, the particles just fall, rebound, kiss and hit the bottom whereas with the contact algorithm the particles interact much more with each other in particular we observe a collective motion of some particles. The final steady state is reached later $t = 0.6$ with the contact algorithm and the distributions of the particles are not the same at all with the two procedures. With the contact procedure 5.20 the dynamic of the particles seem physically unrealistic.
Concerning the CPU time of the contact algorithm and our model it is not possible to make a comparison as the behavior is different. We should note however that the computation of the signed distance functions is very fast as it only uses the gravity centers, the projection of the velocities is almost always active from $t = 0.17$ as the particles are close, in particular from $t = 0.25$ to $t = 0.6$ it is active between the closest bodies inducing a higher computational cost than just computing a short-range repulsive force, in addition it may have an influence on the resulting late steady state.

We now investigate the sedimentation of 100 rigid particles of radius $R = 0.01$ subject to the gravity force $g = -70$. The corresponding repulsive coefficient are:

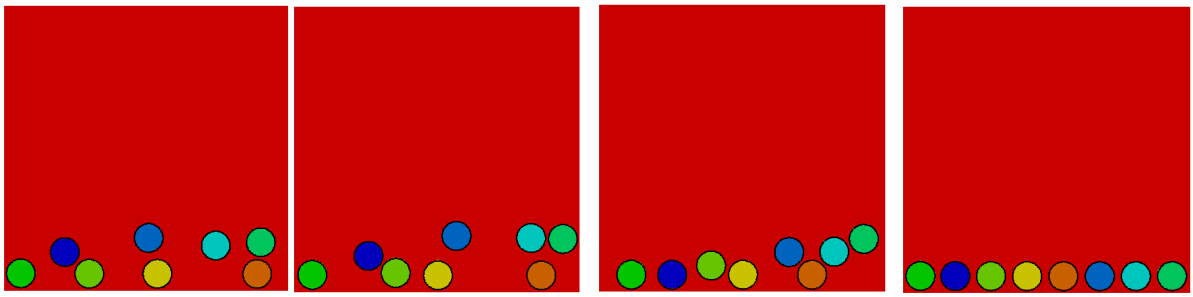
$$k_x = -g/7, \quad k_y = -g/7, \quad k_x^{\text{wall}} = -g/28, \quad k_y^{\text{wall}} = -g.$$

The Figure 5.22 shows the results obtained with the FreeFem ++code implemented by B.Maury and A.lefebvre (see [105]) on a mesh with x elements. Using our model, the simulations are performed on a grid of size (512×512), the half interface thickness ε is set to the mesh size Δx and the results are shown in Figure 5.23.
Despite a symmetrical configuration the results in Figure 5.22 are unsymmetric from the beginning whereas our results stay symmetric until time $t = 1.44$. The only common point between these two simulations is the time simulation corresponding to the static equilibrium which is reached at the same time $t = 4.8$. Otherwise, with our model the particles falls symmetrically, at $t = 1.44$ some particles have already hit the bottom, many drafting, kissing and tumbling process take place at several times. At the final steady state, a packing of the bodies is formed at the bottom composed of different layers where the particles have tried to fulfill each layer.
On the contrary, on Figure 5.22, we observe a convection motion, the particles are carried by the fluid and turns collectively in the domain, thus the particles fall more slowly. At the steady state, the packing of the bodies is not uniform different layers are formed but are not fulfilled by particles. We observe clusters of particles.
It is difficult to compare numerically the obtained simulation for the two procedures as the dynamic of bodies is not of the same type, the gravity force seems to have more influence in our case whereas for the other procedure the fluid and the interaction between particles

have the most effect. We should note however, that the proposed method does not allow to reproduced clusters of particles contrary to the contact algorithm proposed by B.Maury.

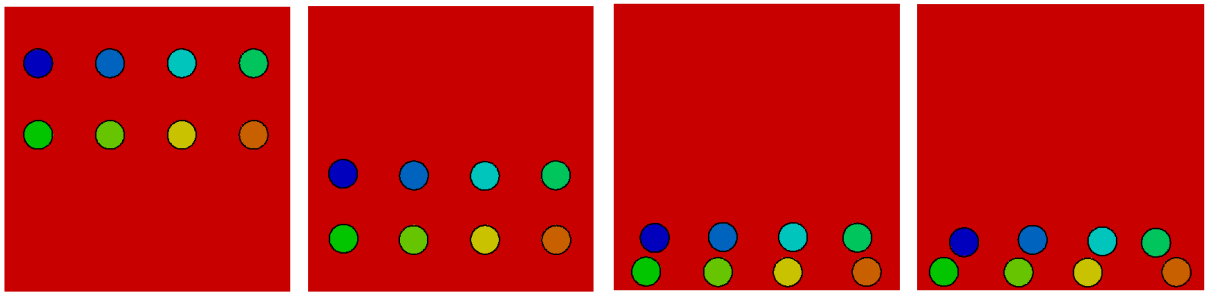


(a) $t = 0$ (b) $t = 0.10$ (c) $t = 0.14$ (d) $t = 0.17$

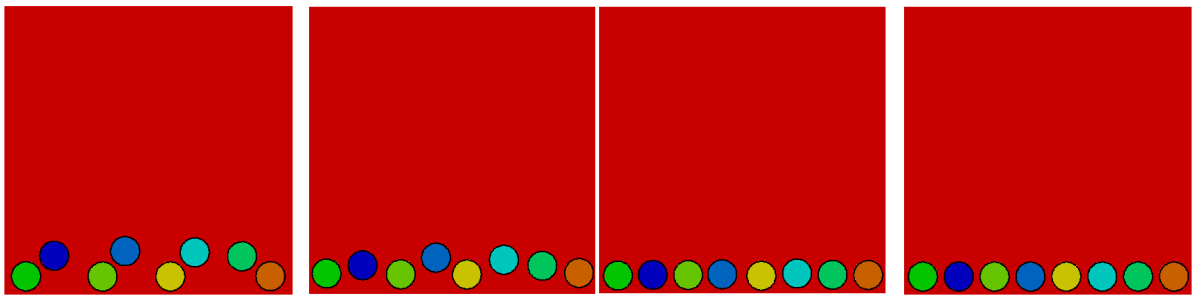


(e) $t = 0.18$ (f) $t = 0.20$ (g) $t = 0.24$ (h) $t = 0.60$

Figure 5.20: Simulation of 8 rigid particles submitted to gravity obtained with the contact algorithm [105].

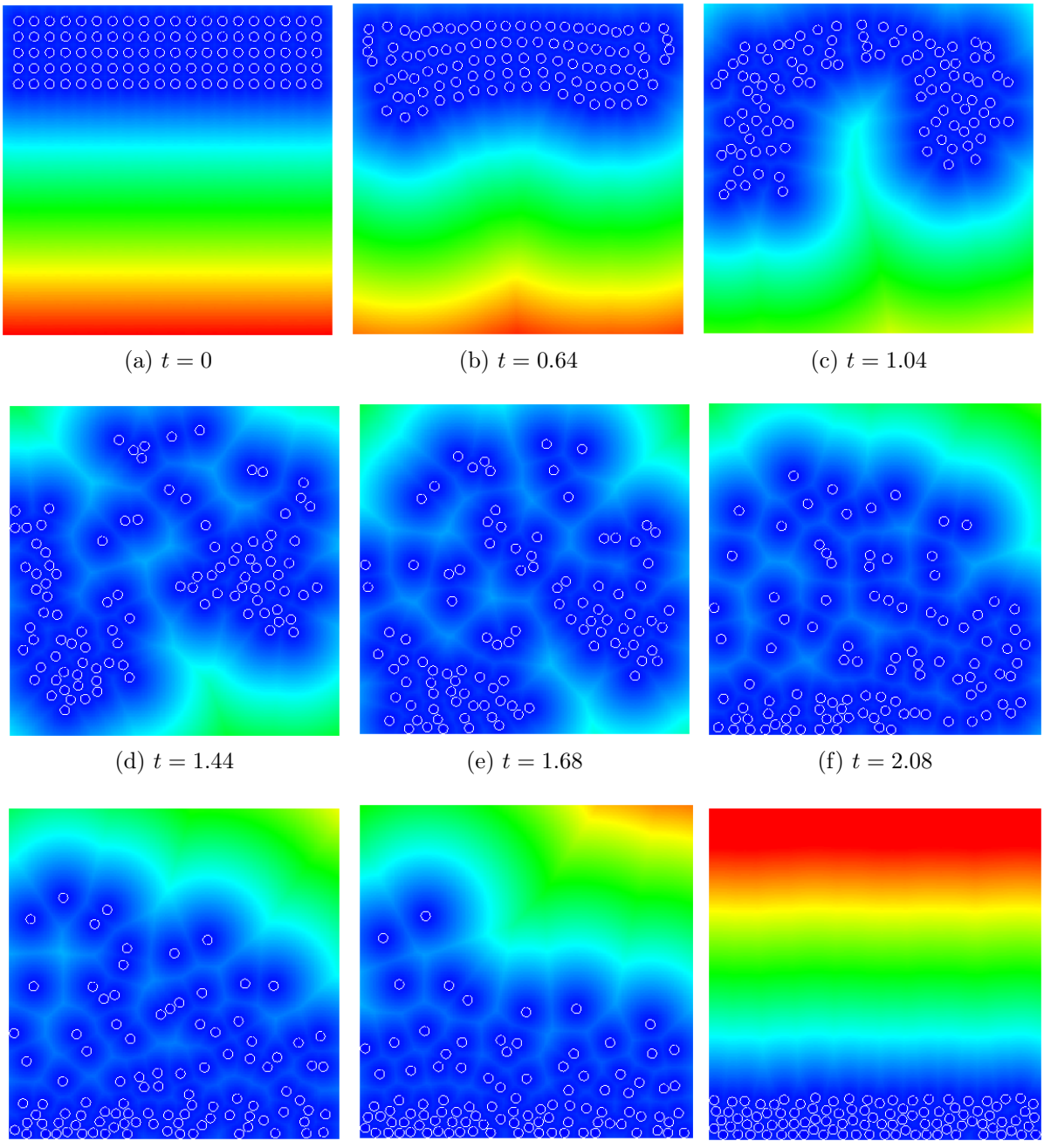


(a) $t = 0$ (b) $t = 0.10$ (c) $t = 0.14$ (d) $t = 0.17$



(e) $t = 0.18$ (f) $t = 0.20$ (g) $t = 0.24$ (h) $t = 0.40$

Figure 5.21: Simulation of 8 rigid particles submitted to gravity obtained with the proposed collision model.

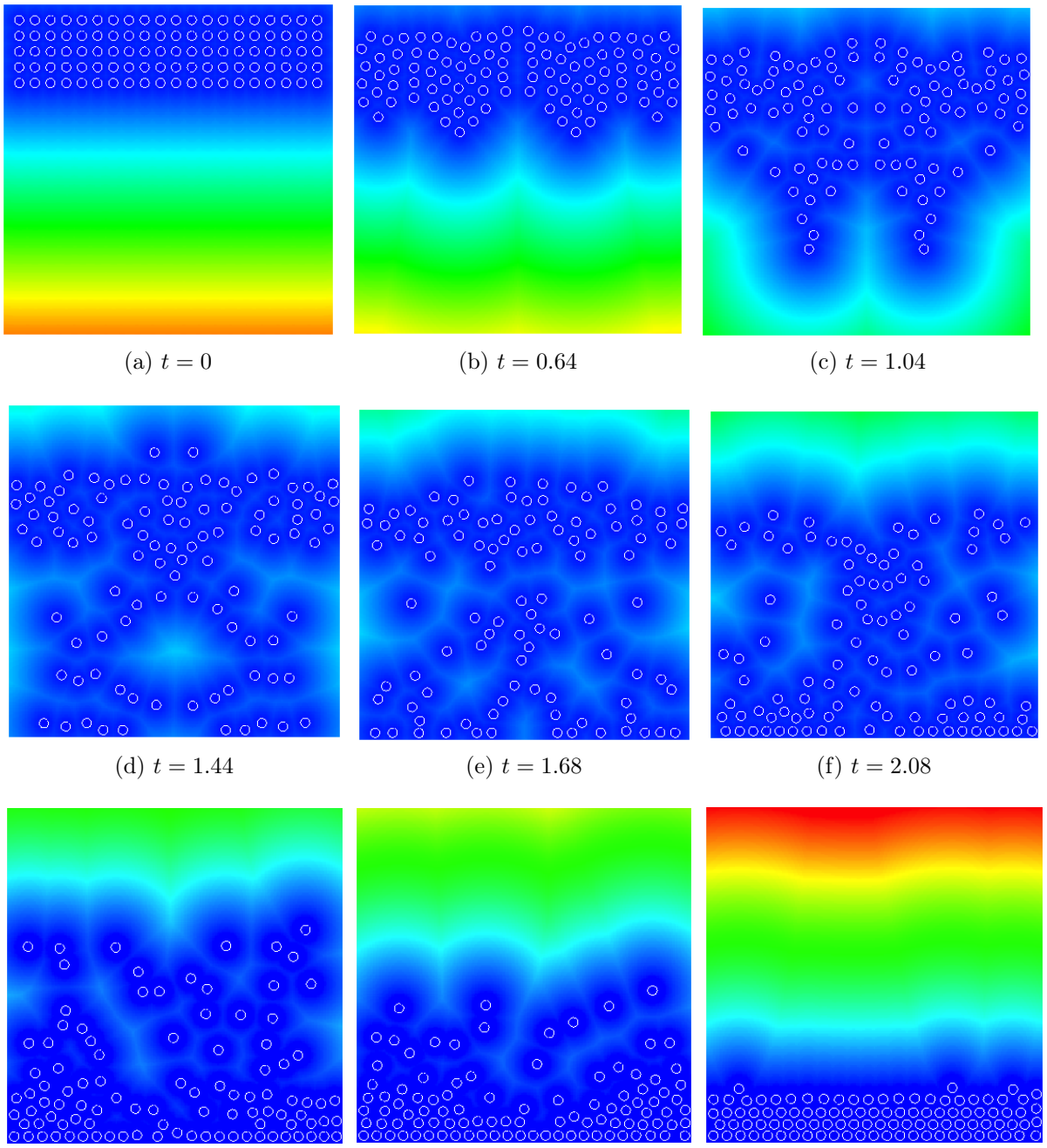


(a) $t = 0$ (b) $t = 0.64$ (c) $t = 1.04$

(d) $t = 1.44$ (e) $t = 1.68$ (f) $t = 2.08$

(g) $t = 2.4$ (h) $t = 2.88$ (i) $t = 4.8$

Figure 5.22: Simulation of 100 rigid particles submitted to gravity obtained with the FreeFem code [105]



(a) $t = 0$ (b) $t = 0.64$ (c) $t = 1.04$

(d) $t = 1.44$ (e) $t = 1.68$ (f) $t = 2.08$

(g) $t = 2.4$ (h) $t = 2.88$ (i) $t = 4.8$

Figure 5.23: Simulation of 100 rigid particles submitted to gravity obtained with our model. The background colors show the level set amplitude.

5.5 Comparison of the repulsive force with or without constant intensity

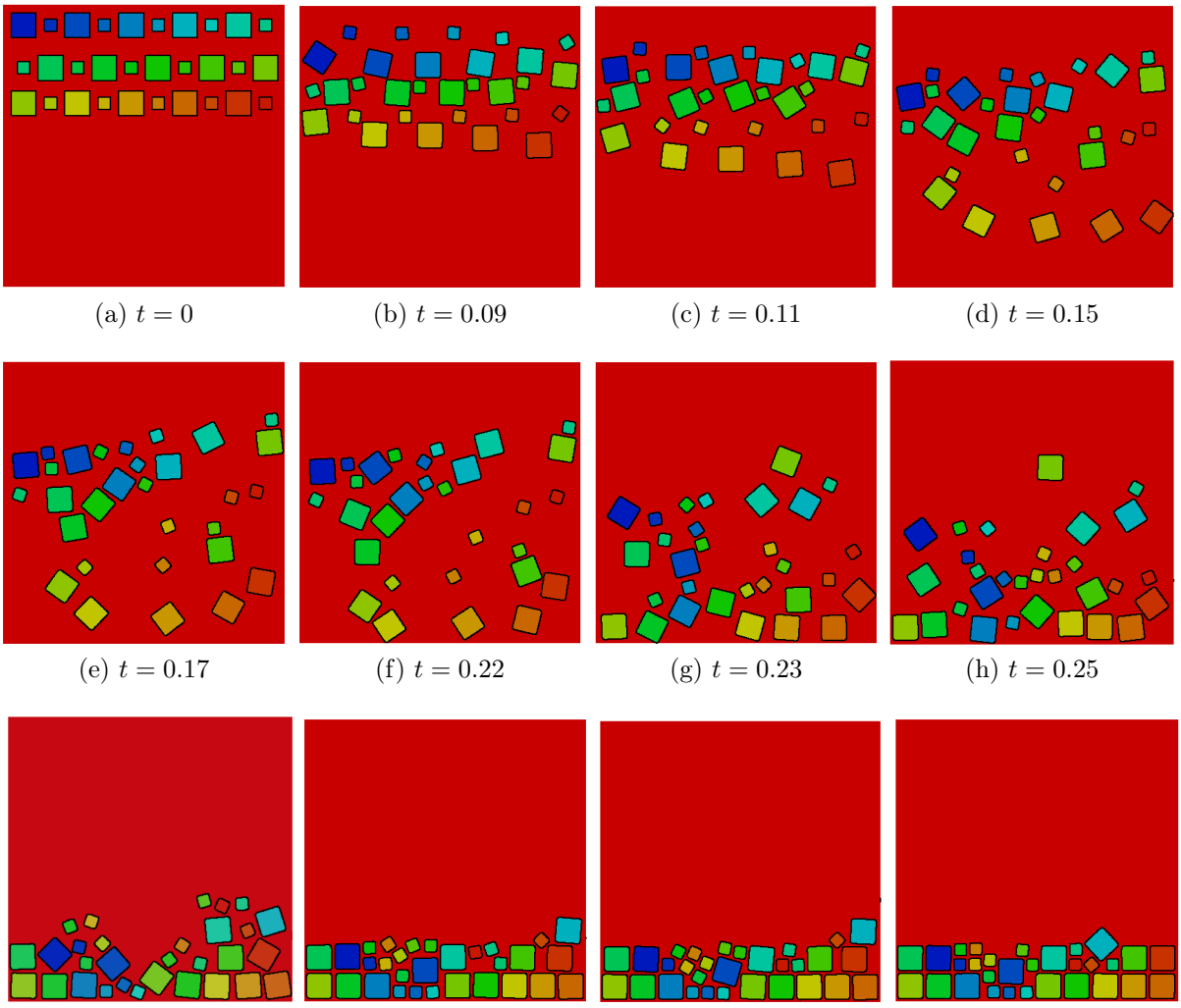
In this section is shown the dynamic of rigid bodies of various size, the numerical contacts are either handled by using the short-range repulsive force with a constant intensity or according to the relative velocity between closest bodies.

The simulations deal with the sedimentation of 30 rigid bodies for the case of disks the radii are $R = 0.05$ and $R = 0.025$ and we also considered rigid squares each side is equal to $\sqrt{\pi}R$.
The simulations are performed on a grid of size (512×512) and the half thickness of the interface is $\varepsilon = 1.5\Delta x$. The coefficient of gravity g is set to -980. The repulsive coefficients are fixed to 100 in the constant intensity force case.

Figures 5.25 and 5.24 show the sedimentation of 30 rigid squares of different sizes. In the presence of rigid particles, the hydrodynamical instabilities are more prominent, as initially the particles are close, the symmetry loss is almost immediate. Once, the symmetry lost, we can not observe the same dynamics for these two simulations, as due to instabilities the flow exhibits a very complex behavior, and it seems obvious that the trajectories of rigid bodies can not be the same for the two simulations. This is even more visible on Figures 5.27,5.26 where the vorticity field is represented at different time. At first the vorticity is created by the falling of the bodies whereas, later, the vortices convect the bodies, the bodies are then driven by the fluid flow.

Nevertheless, we can not that globally the behavior of the rigid bodies is in a sense similar: we observe that until $t = 0.17$ for both collision model the obtained results are similar, the particles are carried by the fluid flow and turns in the domain convected by the created vortices. At $t = 0.25$ the minimal distance between the rigid bodies is smaller for the simulation represented in Figure 5.24 than the particles in Figure 5.25, there are more interactions between the particles. At $t = 3.18$, the steady state has been reached for both simulations, the configuration of bodies are different. With the collision model depending on the relative velocities, the particles have fulfilled the bottom of the computational domain whereas with the constant model we observe three layers of particles not completely filled.

The same simulation for rigid disks having the same area has been carried out. The results are presented on Figure 5.28. We only colored six particles to better track their motions, we can see that at time $t = 0.03$ the models provide exactly the same results, then at time $t = 0.08$, we observe a slight delay between the two simulations for the orange and light blue particles. This slight delay is sufficient to change all the particles trajectories. However, a collective motion of the big green particle and the smaller red disks is observed for both simulations. Moreover, globally the motion of the bodies is similar with the two models, the bodies are carried by the fluid, we observe the same process of kissing, tumbling of particles. At the end of simulations, the final configuration is different, here again with the model depending on the relative velocities, the particles have completely fulfilled the domain. The model using the relative velocities allow the particle to be close, meaning that the repulsive constant coefficient may not be well calibrated.

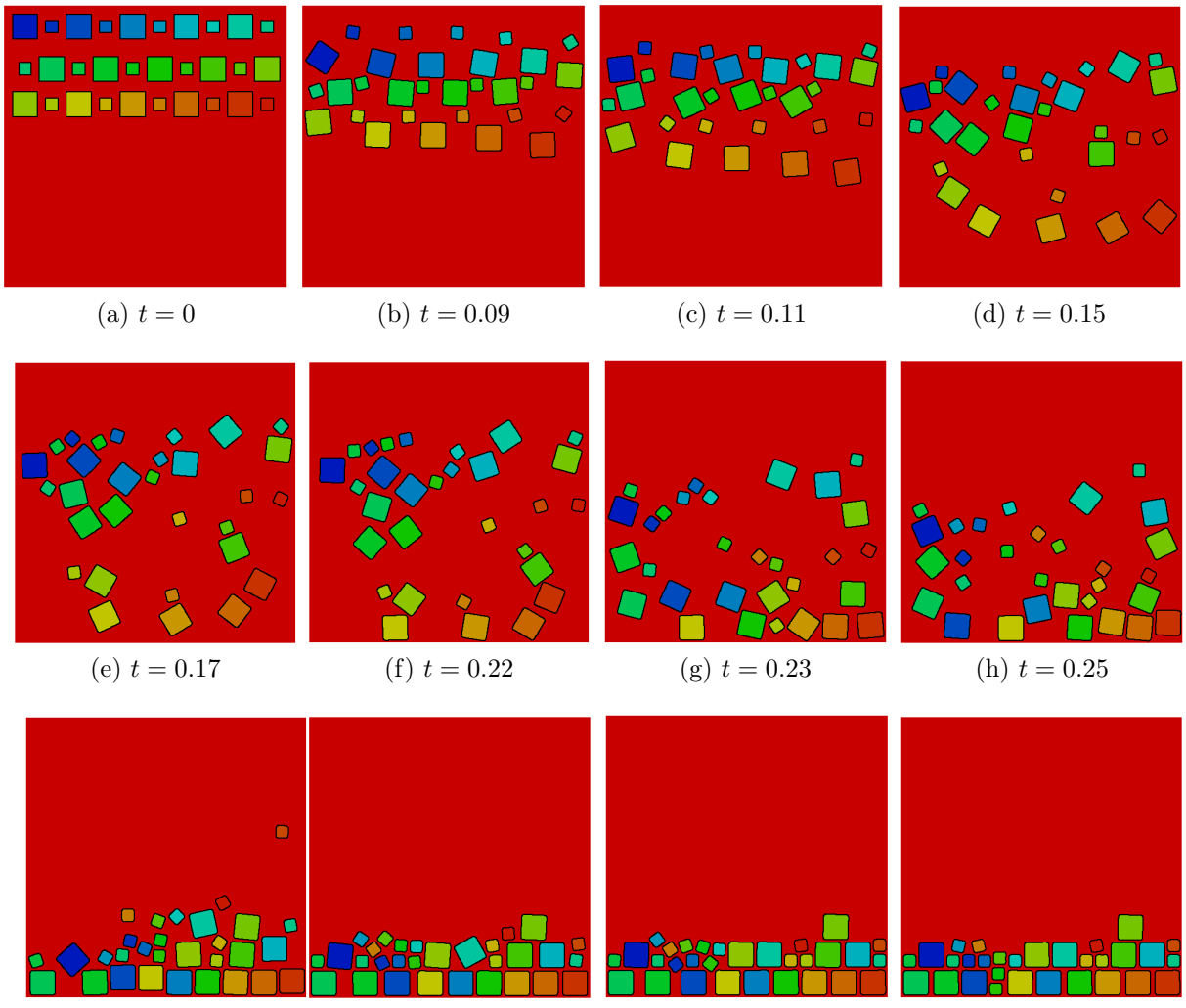


(a) $t = 0$ (b) $t = 0.09$ (c) $t = 0.11$ (d) $t = 0.15$

(e) $t = 0.17$ (f) $t = 0.22$ (g) $t = 0.23$ (h) $t = 0.25$

(i) $t = 0.35$ (j) $t = 0.5$ (k) $t = 1.$ (l) $t = 3.18$

Figure 5.24: Simulation of 30 square rigid bodies of different size falling under gravity. The intensity of the force depends on the relative velocities of the closest particles.

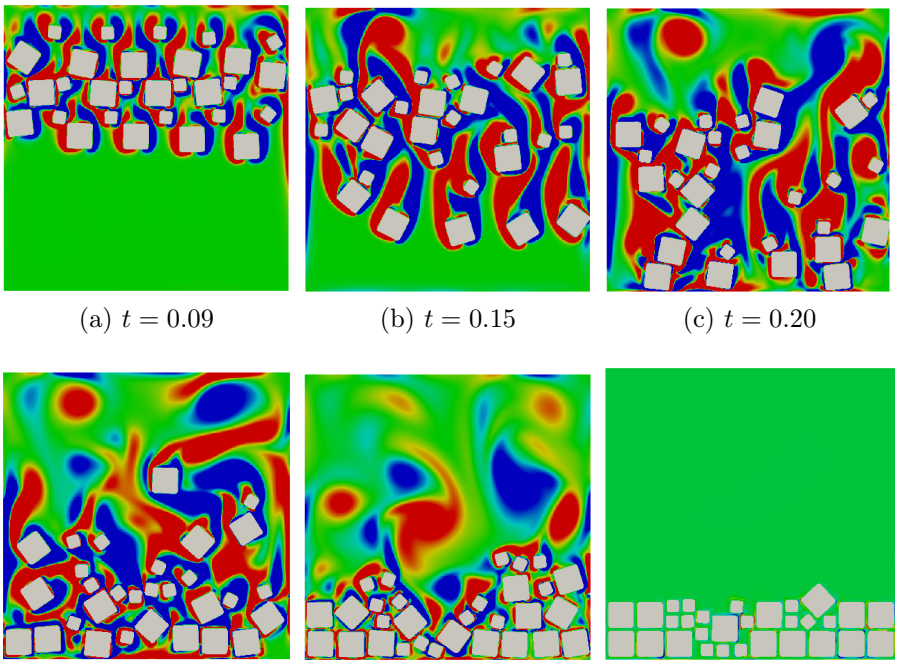


(a) $t = 0$ (b) $t = 0.09$ (c) $t = 0.11$ (d) $t = 0.15$

(e) $t = 0.17$ (f) $t = 0.22$ (g) $t = 0.23$ (h) $t = 0.25$

(i) $t = 0.35$ (j) $t = 0.5$ (k) $t = 1.$ (l) $t = 3.18$

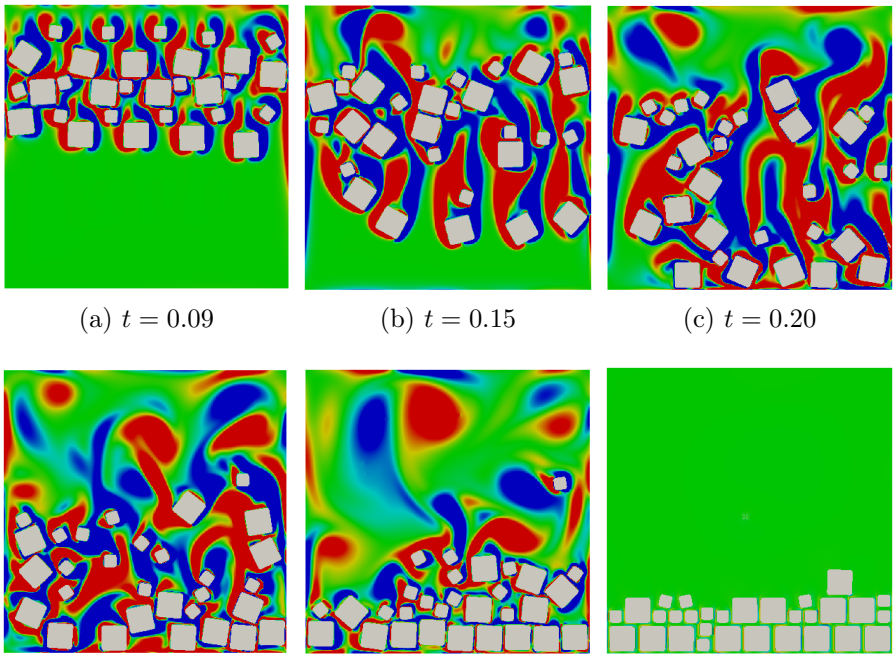
Figure 5.25: Simulation of 30 square rigid bodies of different size falling under gravity. The intensity of the force is constant $k = 100$.



(a) $t = 0.09$ (b) $t = 0.15$ (c) $t = 0.20$

(d) $t = 0.25$ (e) $t = 0.35$ (f) $t = 3.18$

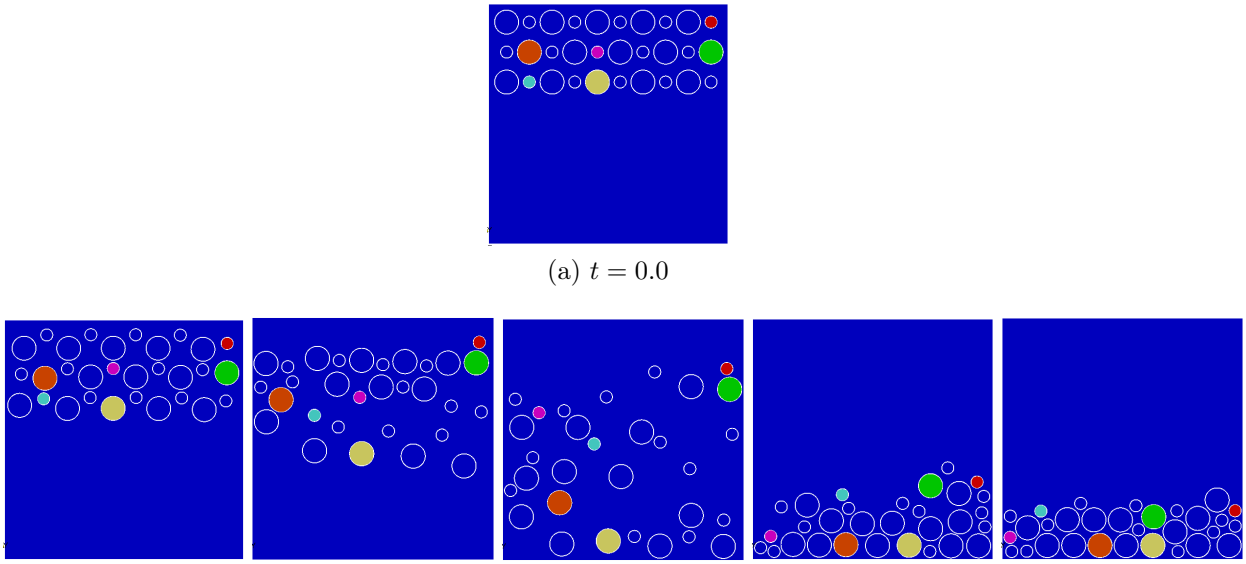
Figure 5.26: Simulation of 30 square rigid bodies of different size falling under gravity. The intensity of the force depends on the relative velocities of the closest particles. The background colors represents the vorticity field intensity.



(a) $t = 0.09$ (b) $t = 0.15$ (c) $t = 0.20$

(d) $t = 0.25$ (e) $t = 0.35$ (f) $t = 3.18$

Figure 5.27: Simulation of 30 square rigid bodies of different size falling under gravity. The intensity of the force is constant. The background colors represents the vorticity field intensity.



(a) $t = 0.0$

(b) $t = 0.03$ (c) $t = 0.08$ (d) $t = 0.16$ (e) $t = 0.3$ (f) $t = 0.6$

(g) $t = 0.03$ (h) $t = 0.08$ (i) $t = 0.16$ (j) $t = 0.3$ (k) $t = 0.6$

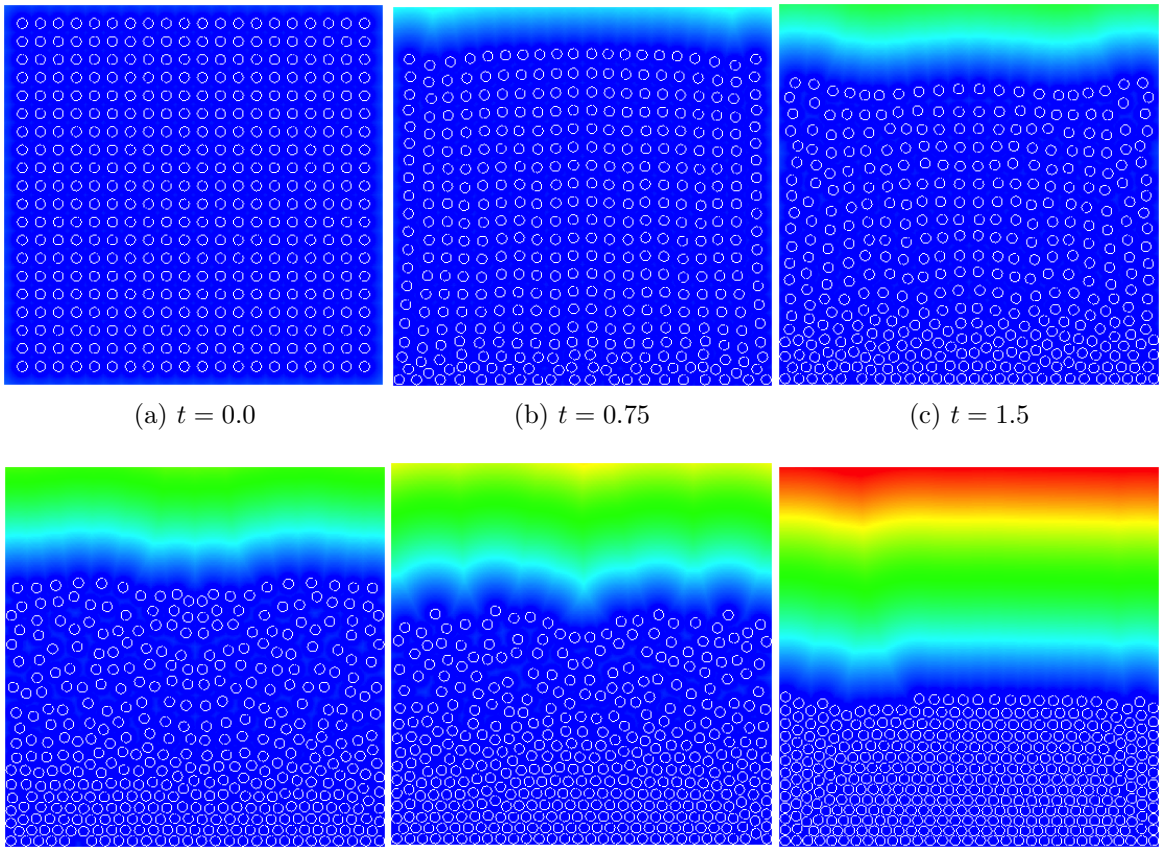
Figure 5.28: Simulations of 30 rigid bodies of different radii ($R = 0.05$ or $R = 0.025$) falling under gravity. The top simulation has been achieved using a constant intensity of the force whereas for the bottom one it depends on relative velocities.

5.6 Dense suspensions of rigid bodies in 2D and 3D

In this part, we present some results of dense suspensions of rigid bodies evolving in a fluid which was performed using our numerical model. The first simulations deal with the sedimentation of 400 rigid bodies of radius $R = 0.01$ in the two dimensional case. The simulations are performed on a grid of size (512×512) and the half thickness of the interface is $\varepsilon = 1.5\Delta x$. The white line shows the real numerical size of the particles corresponding to the isoline $\phi = \varepsilon$. The coefficient of gravity g is set to -980, the associated repulsive coefficients are $-g/10$. The 400 bodies fall down symmetrically to reach a dense repartition at the bottom as can be seen in the Figure 5.29.

The second simulation addresses the 3D case. Figures 5.30 and 5.31 show the simulation of 500 rigid spheres of radius $R = 0.01$ falling under gravity for two different grids of size 64^3 and 128^3. The half thickness of the interface is $\varepsilon = 2\Delta x$. The coefficient of gravity g is set to -980. At initial step, there are five slices of 100 bodies at a distance $d = 0.1$ (distance of two closest bodies' centers). Consequently, on the coarser mesh there is only one full mesh cell between the two numerical slices. Indeed, the numerical radius is $R + \varepsilon \approx 0.041$ and so the repulsive forces are active, whereas on the finer mesh there are on average 6 mesh cells between the two numerical slices as $R + \varepsilon \approx 0.026$. In that case the repulsive forces are negligible. The interactions between bodies occur at once on the coarse mesh while they start after $t = 1.5$ on the fine mesh. As a consequence, the equilibrium state is reached much faster on the fine grid, around $t = 2.5$ instead of $t = 9.4$ for the coarse resolution.

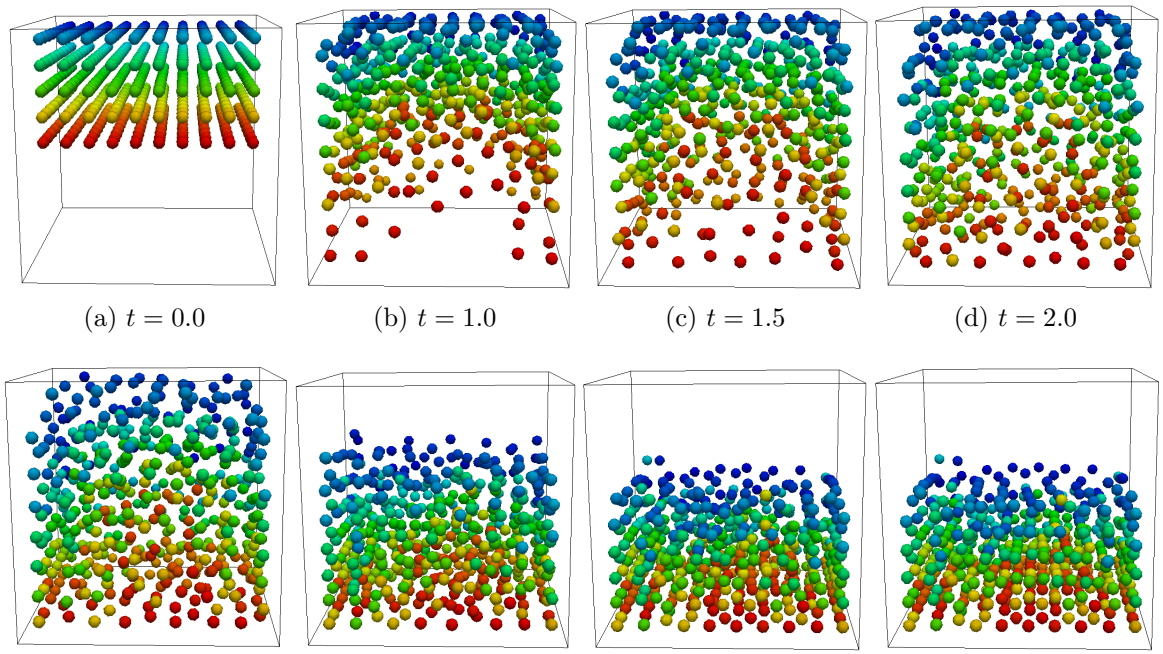
The last simulation deals with the dynamics of 90 rigid disks suspended in a shear flow. The intensity of the short-range repulsive force is proportional to the relative velocities between the closest bodies. The computational domain Ω is a rectangular domain of size $[0, 2] \times [0, 1]$, the grid resolution is (512×256), the regularization parameter $\varepsilon = 1.5\Delta_x \simeq 5.9.10^{-3}$. Figure shows the obtained results at different time, the disk are colored according to the values of the label map L_0 and the background color corresponds the magnitude of the velocity. Until time $t = 0.25$, the rigid bodies moves horizontally through the canal. Then, we observe a convective motion, the solids are carried by the fluid and rotates in the domain. Thanks to the collision model, the numerical contacts between particles are avoided. This test case confirms the ability of the proposed collision model to handle collisions only by using the relative velocities of the closest cells.



(a) $t = 0.0$ (b) $t = 0.75$ (c) $t = 1.5$

(d) $t = 2.25$ (e) $t = 3.0$ (f) $t = 6.9$

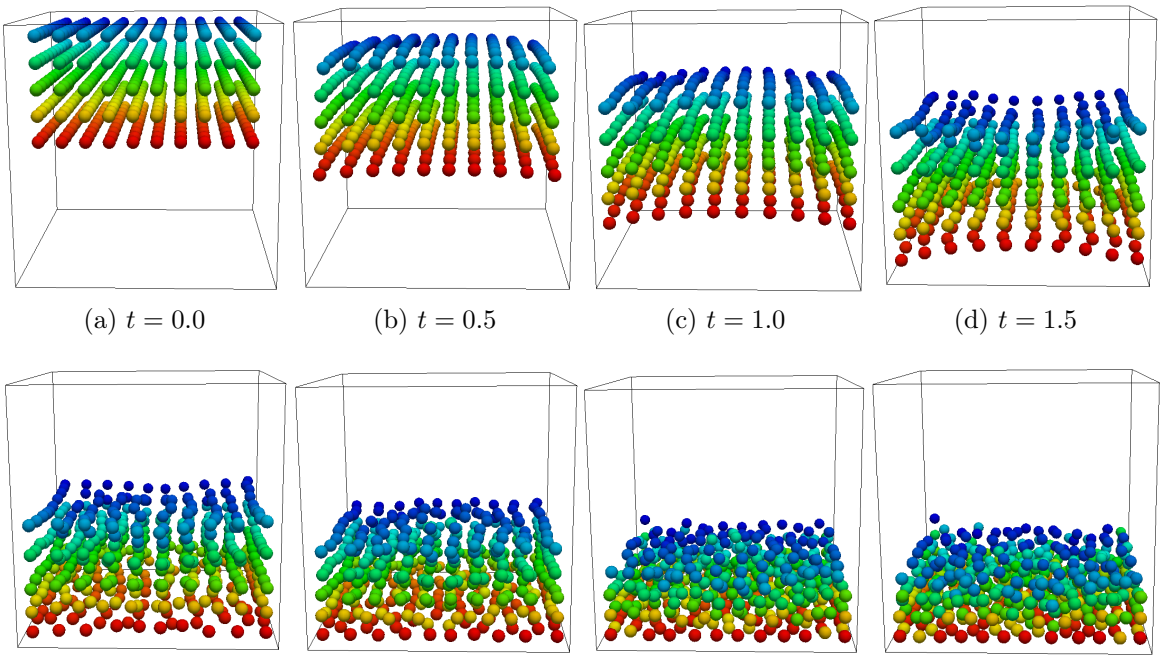
Figure 5.29: Simulation of 400 rigid disks submitted to gravity (the white line corresponds to the level line $\phi = \varepsilon$). The background colors show the level set amplitude.



(a) $t = 0.0$ (b) $t = 1.0$ (c) $t = 1.5$ (d) $t = 2.0$

(e) $t = 2.5$ (f) $t = 4.0$ (g) $t = 9.4$ (h) $t = 14.7$

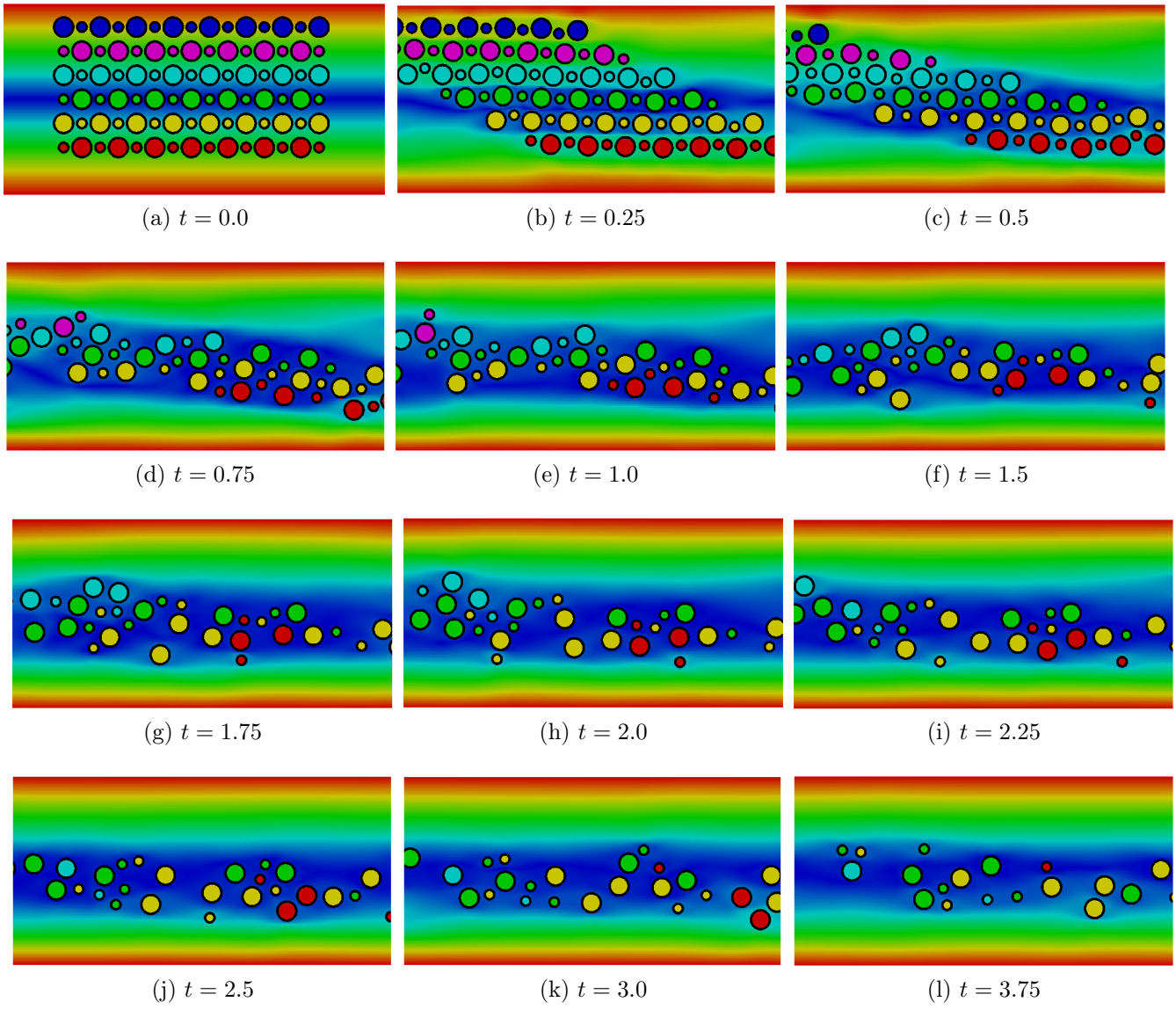
Figure 5.30: Simulation of 500 rigid spheres subject to gravity (grid resolution size 64^3). The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 500^{th} body and red for the fluid that is the 501^{th} object.



(a) $t = 0.0$ (b) $t = 0.5$ (c) $t = 1.0$ (d) $t = 1.5$

(e) $t = 1.8$ (f) $t = 2.0$ (g) $t = 2.5$ (h) $t = 3.0$

Figure 5.31: Simulation of 500 rigid spheres subject to gravity (grid resolution size 128^3). The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 500^{th} body and red for the fluid that is the 501^{th} object.



(a) $t = 0.0$ (b) $t = 0.25$ (c) $t = 0.5$

(d) $t = 0.75$ (e) $t = 1.0$ (f) $t = 1.5$

(g) $t = 1.75$ (h) $t = 2.0$ (i) $t = 2.25$

(j) $t = 2.5$ (k) $t = 3.0$ (l) $t = 3.75$

Figure 5.32: Rigid spheres suspended in a shear flow. The rigid disks are colored with the values of the label map L_0 and the background colors indicates the magnitude of the velocity.

Conclusion

The application performed in this chapter illustrated the capability and efficiency of the present method to handle numerical contacts between structures at low resolution. A penalization model formulated according to the label maps and distance functions is proposed. This model is compared to the penalization model used in [31] in a velocity and pressure formulation. The numerical results confirmed the proposition 1 presented in Chapter 3, the error estimation between the two collision models depend on the size and the number of particles. Moreover, compared to this model which is totally dependent on the number of bodies, the proposed penalization model substancially reduces the CPU time. We also made a comparison between the influence of the proposed collision model and the contact algorithm proposed by B.Maury. During numerical calculations, our model allows to maintain symmetry for a longer time than with the contact algorithm. However, contrary to this contact algorithm, the proposed method does not allow to reproduce particle clustering. A comparison of simulations performed either with a collision model dependent on relative velocities or having a prescribed intensity is carried out. Due to instabilities, the dynamic of the particles is different, we observed however a similar global motion for both models. Moreover, the model depending on the relative velocities seems to provide a more realistic final configuration of bodies as all the bottom domain is fulfilled by the particles.

Chapter 6

Numerical simulation of the dynamics of a suspension of vesicles

Contents

The purpose of this chapter is to validate the proposed model which includes a short-range repulsive force in the case of deformable bodies. An application to immersed vesicles suspensions is proposed. We first introduce the properties of blood flow, red blood cells and vesicles as well as an overview of existing numerical methods to simulate vesicles suspensions. Then, the Eulerian fluid/elastic membrane coupling model introduced in [32] is described. Finally, we describe the model used for the simulations, and a validation of the model in the case of multiple vesicles under various flow are presented.

6.1 Vesicles simulations

6.1.1 Context

The mechanism of blood flow in micro capillaries is still not fully understood because of its complex behaviour. Indeed, its main components are a fluid called plasma and different cells, each of them having different mechanical properties. More precisely, blood is composed of:

1. Plasma: Blood contains 54.3% of plasma which consists of 90% of water, 8% of proteins and 2% of organic and inorganic salts.

2. Red blood cells: These cells are the main cellular components of blood. Indeed, 45% of red blood cells are present in blood, their mission is to transport oxygen via hemoglobin components into the whole body.

3. White blood cells and cellular platelets represents 2% of the composition of blood. The white blood cells play a role in the immune system, while the cellular platelets control the blood cotting.

The average density of blood is close to the density of water, it is around $1060kg/m^3$. From experiments, to theory and numerical simulations, large researchs have been devoted to the study of the behavior of blood flow in blood vessels.
In 1931, Fahraeus and Lindqvist observed a dependence of blood viscosity on the size of capillaries, it has been shown that the blood viscosity decreases when blood flows through capillaries of decreasing diameters. This phenomenon also known as the Fahraeus-Lindqvist effect has been confirmed and explained later in [10; 102; 185]. It has been observed that the red blood cells moving in small capillaries concentrate at the center of the capillaries, thus along the capillary walls only the plasma is present. The effect of a lubricating fluid layer along the capillary walls leads to a decrease of the global resistance of the flow and thus of the viscosity.
Consequently, the viscosity, the velocity of the blood flow in a vessel is highly influenced by the mechanical properties of the cells contained in plasma, the interaction between cells and the interaction of cells with the vessel walls. The red blood cells constitute 99% of the cells present in blood. Thus, the dynamic behaviour of blood flow is largely affected by the properties of red blood cells. In particular, the non Newtonian behavior of blood flow is mainly due to aggregation of RBCs.

Red blood cells

Contrary to other living cells which have nucleus and organelles, red blood cells have the particularity to only contain a fluid called the hemoglobin and a cytoskeleton. This particularity made them highly deformable allowing them to deform and pass through micro capillaries. Their shape is a biconcave-disk of diameter close to $7.8\mu m$ with a thickness of $2.5\mu m$.

The membrane of RBC consists of a bi-layer of phospholipids with one layer composed of hydrophilic head groups and the other of hydrophobic head groups and an elastic cytoskeleton is fixed on the membrane. The physical properties of the RBC are: a fluid membrane, a fixed surface area and a fixed cytoplasm volume [155]. Moreover, its shape is controlled by a bending energy and the deformation of the cytoskeleton induces an elastic force.

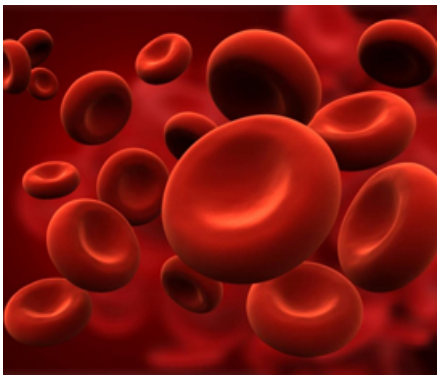


Figure 6.1: Shape of red blood cells.

Vesicles

In aqueous solutions, lipids molecules can form bilayer preventing contacts between hydrophobic groups of molecules and water. When these lipids bilayers formed a closed membrane, they are called vesicles.
Like RBC, the membrane of a vesicle is composed of phospholipids bilayer formed by one layer of hydrophilic head groups and a second layer composed of hydrophobic head groups like illustrated in Figure 6.2. The bilayer size is around $4nm$ thick whereas the size of the vesicle is around $10\mu m$.
The physical properties of their membrane are similar to the membrane of red blood cells: the membrane is fluid, inextensible, opposes the bending and is impermeable so that the volume of fluid enclosed in the vesicle is conserved. The only difference comes from the elasticity due to the cytoskeleton fixed on the RBC membrane. Nevertheless, this is a very desirable model to understand the mechanical behaviour of living cells, in particular in dimension two as due to the membrane inextensibility the elastic force coming from the cytoskeleton is negligible.

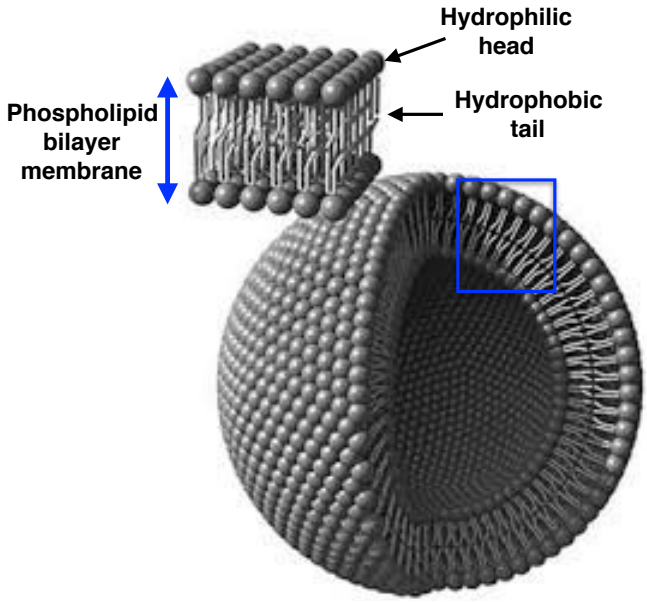


Figure 6.2: Phospholipidic vesicle.

6.1.2 Vesicle simulation

The physical properties of vesicles involves to deal with two fluids and at the same time impose the inextensibility constraint of the membrane, the resistance of the membrane to bending and the conservation of the internal fluid volume.
The fluid enclosed by the vesicle and the external fluid can have different material properties, such as a different viscosity and density. The two fluids are governed by the incompressible Navier-Stokes equations. In micro-capillaries, as the Reynolds number is small (around 10^{-4}), the incompressible Stokes equations are often used.

The dynamic of vesicles have been studied both experimentally [1; 22; 40; 93; 114; 164] and theoretically [97]. Later, the deformation of the membrane has been taken into ac-

count experimentally in [40]. Since then, experiments [1; 22; 40; 93; 114; 164; 183] have become more and more quantitative. A transition from tumbling to tank-tread like motion has been observed depending on the increase of the viscosity ratio between the external fluid flow and the fluid enclosed by the vesicle membrane[92; 94].

Numerous mathematical models and computational techniques have been used to simulate vesicles in fluid flow. Most of vesicle simulations are derived from molecular dynamics models [119] and continuum models. The continuum models are based on the Helfrich energy [21; 76] provided by:

$$E_b = \frac{\alpha}{2} \int_{\Gamma_1} k^2 \tag{6.1}$$

where k is the mean curvature of the surface and α is the bending modulus of the vesicle and Γ_1 is the vesicle boundary.

Among the numerous numerical methods derived from the continuum approach, the Boundary Integral methods, the Immersed boundary method and Eulerian approachs such as level set and phase field approach are the most popular.

Boundary integral method

The Boundary Integral Method is one of the most popular method for vesicle simulation. This technique has been first used to study the behavior of capsules and drops [110; 144; 145; 148; 197]. In [146], Posrikidis applied this approach to simulate an inextensible vesicle suspended in a viscous fluid.
This technique consists in coupling the green tensor of the Stokes equations and the elastic forces on the membrane. The inextensibility of the membrane is imposed either by means of the tension force with a high value energy or by using a Lagrangian multiplier. The bending force is derived from the Helfrich energy. A common expression of the elastic energy of the membrane which takes into account bending and membrane tension is:

$$E(k, \gamma) = \int_{\Gamma_1} \frac{\alpha k^2}{2} + \gamma \tag{6.2}$$

where γ is the membrane tension, α its bending modulus and k its mean curvature. To evolve the vesicle, the velocity has to be computed on each point of the interface, which itself depends on an integral of all the other points of the interface. Indeed, the velocity at a point x depends on the integral of the green tensor of the Stokes equations G on the vesicle boundary, that is :

$$U(x) = U_\infty(x) + \int_{\Gamma_1} G(s - x) f_e ds$$

where f_e is the total force derived from the elastic energy E and U_∞ is the imposed velocity. Thus, the complexity of this algorithm is M^2 where M is the number of marker points on the membrane. However, thanks to the integral equation formulation, this method is still efficient as only the membrane is discretized. To lower the complexity, the Fast Multipole Method can be used and allows to reduce the computational complexity to $M \log(M)$.

In [178; 179], this method is used for the simulation of a dense suspension of vesicles, the associate computational complexity is $NM \log(M)$ where N represents the number of

vesicles. Numerous 2D simulations are explored including the simulation of 256 vesicles suspended in a Poiseuille flow, the algorithm is very efficient, as for 64 points on each vesicle membrane, the computational cost represents only 4s per time step. Different numerical experiments of a suspension of vesicles in shear flow are also explored.
An extension to the three dimensional case has been achieved in [179; 195; 196].

An other extension of the method has been introduced in [91; 147]. Efficient and stable algorithms for simulating accurately two-dimensional vesicle suspensions at low resolution are proposed. These algorithms include adaptive time stepping, reparametrization of the membrane, a correction of vesicles' area and lenght is used to enforce the inextensibility of the membrane and volume conservation. Moreover, to handle numerical contacts between vesicles, a collision detection algorithm is performed with spectral accuracy and a repulsion force is implemented to keep a minimal distance between vesicles. The repulsive force is inspired from collision model from contact mechanics. The model is in a polynomial form and is developed for converging vesicles with low velocities. The parameters of the repulsive force are calibrated by comparing its effect on a low resolution compared to high resolution simulation for which the lubrication forces are correctly captured. Different bench are presented and show the efficiency of the method to deal with multiple vesicles.

The Boundary integral approach has been extensively used for the simulation of vesicles like in [95] where two and three dimension numerical experiments of vesicles suspended in Poiseuille flow are presented. In [62], the study of a vesicle suspensed in a curve flow revealed that the vesicle tends to migrate to the regions of high shear rate. In [63], the behavior of a vesicle suspended in a shear flow is studied. The main drawback of this method is its limitation to Stokes fluid flow.

Immersed Boundary method

The immersed boundary method uses an Eulerian description for the velocity and the incompressibility of the system and a Lagrangian one for the membrane configuration including the location of the membrane and the stretching. The immersed boundary model for vesicles is based on an energy which depends on the mean curvature H, the bending modulus α, the coefficient of membrane tension γ and the pressure θ between the inner and outer fluids. This energy is expressed as:

$$E = \int_{\Gamma_1} (2\alpha H^2 + \gamma) + \theta \int_{\Gamma_1} dx \tag{6.3}$$

The interactions between the fluids and the membrane is taken into account by the Willmore function:

$$W = \Delta_{\Gamma_1} H + 2H(H^2 - \alpha)$$

where Δ_{Γ_1} is the surface laplacian at the interface Γ_1. The total force exerted on the membrane is then given by:

$$F = W + 2\gamma H n + \nabla_{\Gamma_1} \gamma + \theta n$$

where n is the normal to the interface pointing outside and ∇_{Γ_1} the surface gradient. The

immersed boundary model is then:

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U) - \mu\Delta U + \nabla p = f & \text{in } \Omega \\ f(x,t) = \int_0^L F(s,t)\delta(x - X(s,t))ds & \text{in } \Omega \\ \dfrac{\partial X(s,t)}{\partial t} = u(X(s,t),t) = \int_\Omega u(x,t)\delta(x - X(s,t))ds & \text{in } \Omega \end{cases} \tag{6.4}$$

The first application to vesicle of the immersed boundary approach has been performed in [194] to study the behavior of vesicles in a shear flow and a Poiseuille flow. A Lattice Boltzmman method is employed to solve the flow field over the Eulerian grid, results obtained are in a good agreement with the experiments, in particular the Fahraeus Lindviquist effect is reproduced.
In [101], an IBM method is developed, the inextensibility constraint is imposed by means of a penalty immersed boundary method (pIB). The principle of the pIB technique is to use two Lagrangian immersed boundaries to represent the vesicle membrane. One is massless and interacts directly with the fluid flow. The second boundary is massive and its motion is based on the vesicle dynamics including the inextensibility constraint. The connection between the two boundaries is achieved by applying penalty forces on both boundaries. Numerical simulations of 56 vesicles suspended in a shear flow are performed, the computational time is almost independent on the number of vesicles, the computational complexity is similar to the Boundary Integral Method that is $NM\log(M)$ where N is the number of vesicles. To avoid numerical contacts between vesicles fine grids are used. The addition of a vesicle in a domain almost filled with vesicles requires to refine the mesh grid.
In [59], the deformation and motion of red blood cells and vesicles passing through a micro-channel is investigated.

Phase field approach

The first application of the phase field approach for vesicle and membrane has been introduced in [7–9]. The description of the membrane is achieved by means of a scalar function which assumes a constant value in each region (for instance -1 in the fluid and 1 in the structure) and varies continuously from these two values across a thin boundary layer of thickness ε. This function namely the phase field, derived from a physical model, provides an energy of the form:

$$E_b(\phi) = \frac{\alpha}{2}\int_{\Omega_i} |f(\phi)|^2 \tag{6.5}$$

where :

$$f(\phi) = -\varepsilon\Delta\phi + \frac{1}{\varepsilon}(\phi^2 - 1)\phi$$

and ε is the thickness of the transition region between -1 and 1. Up to a constant multiple, this energy tends to the bending energy 6.1 when ε tends to zero. The difference between the inner fluid volume and the outside volume fluid is provided by:

$$A(\phi) = \int_\Omega \phi(x)dx \tag{6.6}$$

and an approximation of the area is shown to be:

$$B(\phi) = \int_\Omega \frac{\varepsilon}{2}|\nabla\phi|^2 + \frac{1}{4\varepsilon}(\phi^2-1)^2 dx \tag{6.7}$$

Thus, the problem reduces to find a phase field ϕ which minimizes the energy 6.5 with the constraints of fixed values α for A and β for B. Using a penalty method with penalty coefficients M_1 and M_2, the problem becomes a minimization of :

$$E(\phi) = E_b(\phi) + \frac{M_1}{2}(A(\phi)-\gamma)^2 + \frac{M_2}{2}(B(\phi)-\beta)^2 \tag{6.8}$$

The gradient of this energy is then added to multi-fluid flows Navier-Stokes equations. An extension to dimension three is proposed in [7].

Level set approach

In [32], Cottet et Maitre proposed an application of the level set method to vesicles simulation. They demonstrate that the stretching of the membrane can be explicitly computed using the level set field . An elastic force is derived to impose the surface incompressibility. This force is explicited in the Navier-Stokes equations, which is coupled to the advection equation of the level set function which captures the vesicle interface. In [129], T. Milcent completes this model by adding a curvature force to model to the resistance of the membrane to bending. The final model consists in the multi-fluid flow Navier Stokes equations supplemented by two forces which depends on the level set function, and combined to a transport equation of the level set function ϕ by the fluid velocity field. The validation of the model in dimension two and three has been achieved through the simulation of a vesicle suspended in a shear flow.

An other level strategy has been proposed in [153]. Both the level set function and its gradient are advected by the fluid velocity and a four-step projection method is developed to enforce the volume conservation and the membrane inextensibility. Numerical experiments of dense suspension of vesicles under various flow conditions are explored. The similarities of the phase field approach and the level set method for the simulation of vesicle have been investigated in [116] where the two models are compared.

6.2 Vesicle model in a level set framework

This section is devoted to the description of the model used for the numerical simulation of multiple vesicles evolving in a viscous fluid flow. In a first place, we recall the Eulerian model introduced in [32; 33] in the case of one immersed vesicle. To model the inextensibility of the membrane and its opposition to bending, two forces are exerted on the membrane and depend on a level set function which captures the vesicle boundary. Then, the combination of this model and the proposed capturing method is presented as well as its numerical resolution. The following notations are used:

- n: the normal
- H: the mean curvature
- α: the bending modulus

- λ: the rigidity of the membrane
- ∇_s: the surfacic gradient
- Δ_s: the surfacic laplacian

6.2.1 Case of one immersed vesicle

The computational domain $\Omega \in \mathbb{R}^d, d = 2$ is decomposed as $\Omega = \Omega_{N_f} \cup \Omega_1$ where Ω_{N_f} is the fluid subdomain and Ω_1 is the region occupied by the vesicle and Γ_1 denotes its boundary. The vesicle boundary is captured by a level set function ϕ_1. The fluid enclosed by the vesicle and the outer fluid are governed by the incompressible Navier-Stokes equations.

Elastic force

A strong elastic force F_e is introduced in order to impose the inextensibility of the membrane. This force derives from an elastic energy which depends on the variation of surface area of the membrane and on a parameter that provides the intensity of the force required in order to stretch the membrane.
In the level set context, it has been shown in [32; 33; 129] that in the case of incompressible fluid, the level set function records the stretching. Indeed, at each time t, the variation of the surface area is proportional to $\frac{|\nabla\phi_1(\cdot,t)|}{|\nabla\phi_1(\cdot,0)|}$. A part of the membrane elasticity is defined from the local area variation thus, assuming that the initial level-set function is a signed distance function (i.e. with modulus of gradient equal to one), one can consider an elastic energy E_e depending on $|\nabla\phi_1|$ defined as:

$$E_e = \int_\Omega E(|\nabla\phi_1|)\frac{1}{\varepsilon}\zeta(\frac{\phi_1}{\varepsilon})dx \tag{6.9}$$

The function E is a constitutive law for the membrane such as $E(1) = 0$ meaning that there is no initial stretching. A model usually used for vesicles is :

$$E_e(r) = \lambda(r-1)^2 \tag{6.10}$$

with $\lambda >> 1$. By differentiating this energy and using the advection equation of the level set function, it has been proved in [32] that the elastic force is:

$$F_e = \left\{\nabla(E'(|\nabla\phi_1|)) - \text{div}\left(E'(|\nabla\phi_1|)\frac{\nabla\phi_1}{|\nabla\phi_1}\right)\frac{\nabla\phi_1}{|\nabla\phi_1|}\right\}|\nabla\phi_1|\frac{1}{\varepsilon}\zeta\left(\frac{\phi_1}{\varepsilon}\right) \tag{6.11}$$

Denoting by $e_1 = |\nabla\phi_1|$ the stretching and using a renormalization procedure, this force can be reformulated as:

$$F_e = \left(\nabla(E'(e_1)) - \nabla\cdot(E'(e_1)n(\phi_1))\,n(\phi_1)\right)\frac{1}{\varepsilon}\zeta\left(\frac{\phi_1}{\varepsilon}\right) \tag{6.12}$$

An advantage of this reformulation is that it reduces the degree of derivation of ϕ in the elastic force by one.

Bending force

Using the level set function, the bending energy [21; 76] is expressed as:

$$\mathcal{E}_c = \int_\Omega G(H(\phi_1))\frac{1}{\varepsilon}\zeta(\frac{\phi_1}{\varepsilon})|\nabla\phi_1|dx \tag{6.13}$$

where G is provided by:

$$G(r) = \frac{\alpha}{2}r^2$$

The expression of the curvature force F_c can be deduced by differentiation [115]:

$$F_c = \nabla\cdot\left(-G(H(\phi_1))\frac{\nabla\phi_1}{|\nabla\phi_1|} + \frac{1}{|\nabla\phi_1|}\mathbb{P}_{|\nabla\phi_1|^\perp}\left(\nabla\left(|\nabla\phi_1|G'(H(\phi_1))\right)\right)|\nabla\phi_1|\zeta\left(\frac{\phi_1}{\varepsilon}\right)\right). \tag{6.14}$$

where

$$\mathbb{P}_{\nabla\phi_1^\perp} = \mathbb{I}_d - n(\phi_1)\otimes n(\phi_1)$$

is the orthogonal projector operator. As the averaged curvature H is given by:

$$H(\phi_1) = \nabla\cdot(\frac{\nabla\phi_1}{|\nabla\phi_1|})$$

the fourth order derivatives of the level set function are required to compute the bending force.

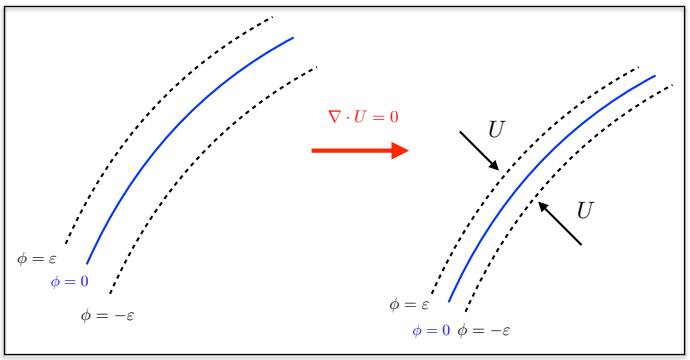


Figure 6.3: Stretching $|\nabla\phi|$.

Transport of the level set function and the stretching

To evolve the vesicle, the level set function is transported by the fluid velocity U:

$$\begin{cases}\partial_t\phi_1 + U\cdot\nabla\phi_1 &= 0\\ \phi_1(\cdot,0) &= \phi_1^0\end{cases} \tag{6.15}$$

As noticed before, the stretching of the membrane is recorded in the term $|\nabla\phi_1|$. Theoretically, the elastic force, due to the high coefficient λ maintains the value $|\nabla\phi_1|$ close to one. However, during numerical calculations the gradient of the level set function can take high values far from the membrane location resulting in an accumulation of errors for the transport of the level set function. In this scenario, it is more convenient to solve

an equation on the stretching $e_1 = |\nabla\phi_1|$ using the transport equation on the level set function ϕ_1:

$$\begin{cases} \partial_t e_1 + U \cdot \nabla e_1 & = -e_1 \left(n(\phi_1) \otimes n(\phi_1)\right) : D(U))) \\ e_1(\cdot, 0) & = |\nabla \phi_1^0| \end{cases} \tag{6.16}$$

As the stretching $|\nabla\phi_1|$ can take large values, a procedure proposed by V. Doyeux [45] is to reinitalize the stretching far enough from the interface, and to keep the right values of the stretching e in a vicinity of the membrane to compute correctly the elastic force. Denoting by e_b and e_a the values of the stretching before and after reinitialization, the value of the stretching e is then provided by:

$$e = e_b \left(H_\varepsilon(\phi_1 + \varepsilon_2) - (H_\varepsilon(\phi_1 - \varepsilon_2)) + e_a\right.$$

The parameter ε_2 is fixed to a higher value than the interface thickness ε. Using this procedure, the information on the stretching on the membrane is recorded, and as the stretching e is close to one, this procedure does not induces larges discontinuities on e.

Complete model

The fluid-structure interaction system hence reduces to the incompressible Navier-Stokes equations with source terms (elastic, curvature forces) combined with a scalar transport equation of the level set function ϕ_1 and the stretching e_1.

Find (U, P, ϕ) solution of :

$$\begin{cases} \rho(\phi)(\partial_t U + (U \cdot \nabla)U) - \operatorname{div}(\mu(\phi)D(U)) + \nabla P = F_e(\phi) + F_c(\phi) & \text{in } \Omega_T = \Omega \times (0,T) \\ \operatorname{div} U = 0 & \text{in } \Omega_T = \Omega \times (0,T) \\ \partial_t \phi + U \cdot \nabla \phi = 0 & \text{in } \Omega_T = \Omega \times (0,T) \\ \partial_t e + U \cdot \nabla e = -e(n(\phi) \otimes n(\phi)) : D(U) & \text{in } \Omega_T \end{cases} \tag{6.17}$$

The density and viscosity functions are provided by:

$$\mu(\phi_1) = \mu_1 + (\mu_f - \mu_1)H_\varepsilon(\phi_1), \quad \rho(\phi_1) = \rho_1 + (\rho_f - \rho_1)H_\varepsilon(\phi_1)$$

where μ_1 and ρ_1 denotes the viscosity and density of the fluid enclosed by the vesicle.

6.2.2 Case of multiple vesicles

We consider here the case of N vesicles immersed in an incompressible fluid flow. The computational domain $\Omega \in \mathbb{R}^d, d = 2$ is hence decomposed as

$$\Omega = \Omega_{N_\mathrm{f}} \cup \left(\bigcup_{i=1}^{N} \Omega_i \right)$$

where Ω_{N_f} is the fluid subdomain and Ω_i is the region occupied by the ith vesicle and Γ_i denotes its boundary. Each vesicle boundary Γ_i encloses an incompressible fluid flow of viscosity μ_i and density ρ_i governed by the incompressible Navier-Stokes equations as the surrounded fluid Ω_{N_f}.

For the simulation of multiple vesicles immersed in a fluid, with repulsion forces, one can uses N level set functions to capture the N interfaces, and N elastic forces, N bending

forces have to be computed inducing a substantial computational cost. These level set functions are then transported by the fluid velocity in order to evolve the vesicles.
Instead of doing this, we propose to use the proposed capturing method. A level set function ϕ captures the union of all interfaces and is defined as a signed distance function in a vicinity of the vesicles interfaces. We introduce a function $e = |\nabla\phi|$ which takes into account the stretching of each membrane.
According to the level set function, the stretching and the N reconstituted level set fields which are defined using the label maps, only one elastic and bending force is computed for the whole set of vesicles. To handle numerical contacts between vesicles, the short-range repulsive force depending on the relative velocities is added to the equations, this is combined with two transport equations: the advection of the level set function ϕ by the fluid velocity and the transport equation of the stretching e.

Computation of the elastic and bending forces

In the context of multiple vesicles, the use of one level set function to capture the union of all interfaces requires to maintain a sufficiently large minimal distance between the vesicles in order to compute correctly the high order derivatives of the level set function which appear in the elastic and bending forces. Indeed, at each point x contained in a regularized interface Γ_i^ε, the stencil used to compute the derivatives has to be included in the subdomain which corresponds to $L_{0,1} = i$, so that the value $\phi(x)$ is the distance from x to Γ_i.
To reduce the number of grid points required in the narrow-gap between two interfaces, the high derivatives of the level set function can be computed by means of the reconstructed level set functions. By doing this, the stencil required to compute a geometrical property at a point x belonging to an interface Γ_i^ε is prolongated to the regions where $L_2 = i$.
Numerically, this procedure does not depend on the number of vesicles as one can introduce a function F which indicates if a point y is associated to the right boundary:

$$\forall x \in \Omega, \forall y \in \Omega, F(x,y) = \begin{cases} L_{0,1}(y) & \text{if } L_{0,1}(x) = L_{0,1}(y) \\ L_2(y) & \text{if } L_{0,1}(x) = L_2(y) \end{cases}$$

For instance, to compute the partial derivative $\partial_{xx}\phi$ at a grid point $x = (i,j)$, if we denote by $x_1 = (i+1,j)$ and $x_2 = (i-1,j)$ its neighbours, using a centered scheme, instead of computing :

$$\partial_{xx}\phi = \frac{\phi(x_1) - 2\phi(x) + \phi(x_2)}{(\Delta x)^2} \tag{6.18}$$

we set:

$$\partial_{xx}\phi = \frac{\phi_{F(x,x_1)}(x_1) - 2\phi_{F(x,x)}(x) + \phi_{F(x,x_2)}(x_2)}{(\Delta x)^2}.$$

where $\phi_{F(x,x)}(x) = \phi(x)$, if $F(x,x_1) = L_{0,1}(x)$ and $F(x,x_2) = L_{0,1}(x)$, we compute exactly (6.18). This procedure is used in order to compute the curvature and other derivatives of order higher than two of the level set function ϕ. Finally, the elastic and bending forces depend on the reconstituted level set functions, the stretching e and the level set function ϕ, we denote by ϕ_F these functions which will use either the label map $L_2(x)$ or $L_{0,1}(x)$ depending on the location of x.

$$F_e = (\nabla(E'(e)) - \nabla \cdot (E'(e)n(\phi_F))\, n(\phi_F))\, \zeta(\frac{\phi}{\varepsilon|\nabla\phi|}) \tag{6.19}$$

and the bending force is computed as:

$$F_c = \nabla \cdot \left(-G(\kappa(\phi_F))\frac{\nabla\phi_F}{|\nabla\phi_F|} + \frac{1}{|\nabla\phi_F|}\mathbb{P}_{\nabla\phi^\perp}\left(\nabla\left(|\nabla\phi_F|G'(\kappa(\phi_F))\right)\right)\right)|\nabla\phi_F|\zeta\left(\frac{\phi}{\varepsilon}\right). \tag{6.20}$$

where

$$\mathbb{P}_{\nabla\phi^\perp} = \mathbb{I}_d - n(\phi)\otimes n(\phi)$$

is the orthogonal projector operator.

Collision model

The short-range repulsive force introduced in Chapter 3 is used here, to avoid numerical contacts between vesicles. Moreover, to compute accurately the curvature and elastic forces on the membranes, a minimal distance has to be maintained between the membranes which corresponds to the interfaces Γ_i^ε of thickness ε. We hence fix the activation distance of the repulsive force ε_b to 2ε. The intensity of the short-range repulsive force is computed according to the relative velocities between closest vesicles just before collision. Each vesicle velocity is computed by averaging the fluid velocity on each membrane Γ_i^ε, so that setting:

$$\forall x \in \Omega, \quad |\Gamma_{L_{0,1}(x)}| = \int_{\Gamma_{L_{0,1}(x)}} dz = \int_\Omega \frac{1}{\varepsilon}\zeta_\varepsilon(\phi_{L_{0,1}(x)}(z))dz \tag{6.21}$$

we obtain the following average velocity over the membrane boundary:
$\forall x \in \Omega$,

$$U_a(L_{0,1}(x)) = \frac{1}{|\Gamma_{L_{0,1}}|}\int_{\Gamma_{L_{0,1}(x)}} U(z)dz = \int_\Omega \frac{1}{\varepsilon}\zeta_\varepsilon(\phi_{L_{0,1}(x)}(z))U(z)dz \tag{6.22}$$

then, at all points $x \in \Gamma_i^\varepsilon, i = (1,..,N)$ where the label map $L_2(x)$ is defined, the intensity of the force is computed as:

$$K_{L_2(x),L_{0,1}(x)} = |(U_a(L_2(x)) - U_a(L_{0,1}(x)))\cdot\nabla\varphi_2(x)|$$

where $U_a(L_2(x))$ denotes the averaged velocity associated to the vesicle $\Omega_{L_2(x)}$, this term provides the averaged velocity of the second closest vesicle to x. Then, at each point x where $\Omega_{L_{0,1}(x)}$ and $\Omega_{L_2(x)}$ are at a distance ε_b this value is fixed. Finally, the short range repulsive force is computed as:

$$\forall x \in \Omega,\ F_{\text{label}}(x) = \frac{|(U_a(L_2(x)) - U_a(L_{0,1}(x)))\cdot\nabla\varphi_2(x)|}{\varepsilon}\rho(x)\zeta_\varepsilon(\varphi_1(x))\frac{\nabla\varphi_2(x)}{\varphi_2(x)}\exp\left(-\frac{\varphi_2(x)}{\varepsilon_b}\right) \tag{6.23}$$

The complete model

The fluid-structure interaction system hence reduces to the incompressible Navier-Stokes equations with source terms (elastic, curvature forces) combined with a scalar transport equation of the level set function ϕ and the stretching e.

Find (U, P, ϕ) solution of :

$$\begin{cases} \rho(\phi)(\partial_t U + (U \cdot \nabla)U) - \operatorname{div}(\mu(\phi)D(U)) + \nabla P = F_e + F_c + F_{\text{label}} & \text{in } \Omega_T = \Omega \times (0,T) \\ \operatorname{div} u = 0 & \text{in } \Omega_T = \Omega \times (0,T) \\ \partial_t \phi + U \cdot \nabla \phi = 0 & \text{in } \Omega_T = \Omega \times (0,T) \\ \partial_t e + U \cdot \nabla e = -e(n(\phi) \otimes n(\phi)) : D(U) & \text{in } \Omega_T \end{cases} \tag{6.24}$$

Denoting by $\rho_{L_{0,1}(x)}$ the density of the vesicle $\Omega_{L_{0,1}(x)}$ we obtain the following density function:

$$\rho_x = \rho_f(1 - \chi_{L_{0,1}(x)}) + \chi_{L_{0,1}(x)}\rho_{L_{0,1}(x)},\ \mu_x = \rho_f(1 - \chi_{L_{0,1}(x)}) + \chi_{L_{0,1}(x)}\mu_{L_{0,1}(x)}$$

This model involved several parameters, in order to achieve numerical simulations, it is most convenient to introduce dimensionless parameters. For this purpose, we introduce reference values. Let L_r, U_r, ρ_r and μ_r denote the characteristic lenght, velocity, density and viscosity scales. Considering the following characteristic quantities :

$$\begin{cases} x = L_r x', \quad y = L_r y', \quad z = L_r z' \\ U = U_r U', \quad P = \rho_r \dfrac{U_r}{L_r} P', \quad t = \dfrac{L_r}{U_r} t' \\ \mu = \mu_r \mu', \quad \rho = \rho_r \rho', \quad \varepsilon = L_r \varepsilon' \\ \phi = L_r \phi', \quad \varphi_1 = L_r \varphi_1', \quad \varphi_2 = L_r \varphi_2' \end{cases}$$

Differentiating (and dropping '), we obtain the following dimensionless system:

$$\begin{cases} \rho(\phi)(\partial_t U + (U \cdot \nabla)U) - \dfrac{1}{R_e}\operatorname{div}(\nu(\phi)D(U)) + \nabla P = \dfrac{1}{W_e}\bar{F}_e(\phi) + \dfrac{1}{W_c}\bar{F}_c(\phi) + F_{\text{label}} & \text{in } \Omega_T \\ \operatorname{div} U = 0 & \text{in } \Omega_T \\ \partial_t \phi + U \cdot \nabla \phi = 0 & \text{in } \Omega_T \\ \partial_t e + U \cdot \nabla e = -e(n(\phi) \otimes n(\phi)) : D(U) & \text{in } \Omega_T \end{cases} \tag{6.25}$$

The first physical parameter involved in this simulation is the Reynolds number Re corresponds to the ratio between the inertial and viscous effects:

$$R_e = \frac{L_r U_r \rho_r}{\mu_r}$$

The Weissenberg number W_e represents the ratio between the characteristic time of the fluid and the relaxation time of the membrane submitted to the elastic force, it is provided by :

$$W_e = \frac{\mu_r U_r}{\lambda}$$

The third physical parameter W_c is the capillary number associated to the bending force which measures the magnitude of the bending force over the hydrodynamic forces:

$$W_c = \frac{\mu_r U_r L_r^2}{\alpha}$$

The last dimensionless numbers involved in this simulation are the viscosity and density ratio between the fluids contained in the vesicles and the outer fluid Ω_{N_f}.

For the numerical simulations, we take as reference the outer fluid's data: $\rho_r = \rho_f$ and $\mu_r = \mu_f$. Moreover, if one takes as characteristic lenght, the lenght of a capillary, one obtains:

$$\rho_r = 1000,\ \mu_r = 10^{-3},\ L_r = 10^{-5}$$

The reference density and viscosity chosen corresponds to the material properties of water, as plasma is mainly composed of water. The obtained Reynolds number is of order 10^{-2}, the parameter W_e has to be chosen in order to impose small variation of the stretching of the membrane.
In order to impose the rigidity of the membrane λ one can look at the typical time scale of the bending force $t_\alpha = \dfrac{\mu_r L^3}{\alpha}$ and the elastic force $t_\lambda = \dfrac{\mu_r L}{\lambda}$. To ensure the inextensibility of the membrane, the time scale associated to the elastic force has to be smaller than the one which corresponds to the bending force. The ratio R between these two time scales given by:

$$R_t = \frac{\alpha}{\lambda L_r^2}$$

has to take values between 10^{-3} and 10^{-5}.

6.3 Numerical procedure

The system is discretized by a finite difference method on a staggered grid (MAC type) with a finer resolution for the level set function as stated in Chapter 1. The fluid velocity and pressure are discretized on the coarse grid while all the other functions (label maps, distance functions and level set functions) are located on the finer grid. To compute the fluid velocity a Projection method is used, the diffusion term are treated implicitly and an explicit scheme is used for the forces and the convective term.

6.3.1 Stability conditions

The advection equation of the level set function is treated explicitly and to ensure the stability of the scheme, we impose the stability condition CFL:

$$\Delta t \leq \frac{\Delta x}{|U|}$$

The explicit treatment of the elastic/coupling induces stability conditions. The Brackbill condition provided in [15] gives a stability condition in the case of perfect fluids separated by an interface with surface tension, this condition is expressed as:

$$\Delta t \leq \sqrt{\frac{W_e}{2\pi}}(\Delta x)^{\frac{3}{2}} \tag{6.26}$$

This condition does not take into account the viscosity of the fluid, moreover in the case of small W_e, the time step has to be very small. In the case of viscous fluids, others stability conditions have been found, in [60] the stability condition is provided by:

$$\Delta t \leq \sqrt{\frac{W_e}{R_e}}(\Delta x) \tag{6.27}$$

For small Reynolds number, this condition allows to use larger time step than with the Brackbill condition. One can refer to [12] where the stability of schemes for immersed elastic membranes have been studied.

For the curvature force, no condition exists, one can however take into account the typical velocity induced by the bending force:

$$U_\alpha = \frac{\alpha}{\mu_r L_r^2}$$

These lead to the condition:

$$\Delta t \leq \frac{\Delta_x}{\max(|U|, U_\alpha, \frac{W_e}{R_e})}$$

6.3.2 Algorithm

The level set function ϕ and the distance functions are updated at each time step using the multi-label fast marching on the finer grid. The narrow-band size set for the simulations are: $NB_{L_1} = 10h$ where h corresponds to the discretization space step associated to the coarser grid (the fluid grid) and we fix a high narrow-band size for the computation of φ_2 and L_2 we set $NB_{L_2} = 10h$ this allows to compute correctly the bending and elastic forces when the vesicles get closer by means of the reconstituted level set functions. In the case of a variable density, we use a Boussinesq Approximation.

At each time step, the algorithm performed the following steps:

1. Computation of the fluid velocity U^{n+1} using an implicit scheme and a projection method of Chorin type according to the forces $F_e^n, F_c^n, F_{\text{label}}^n$

2. Interpolation of the fluid velocity U^{n+1} on the finer grid providing U_g^{n+1}

3. Advection of the level set function ϕ^n with the interpolated fluid velocity in a vicinity of the vesicles:
$$\phi^{n+1} = \phi^n - \Delta t U_g^{n+1} \cdot \nabla \phi^n,$$

4. Transport of the stretching e with the interpolated fluid velocity:
$$e^{n+1} = e^n - \Delta t U_g^{n+1} \cdot \nabla e^n - e^n(n(\phi^n) \otimes n(\phi^n)) : D(U_g^{n+1}),$$

5. Update of the distance function $\varphi_1^{n+1} = |\phi^{n+1}|$,

   Update L_0^{n+1},L_1^{n+1} using ϕ^{n+1}

   Perform the multi label fast marching method described in Chapter 3

6. Update the level set function as a signed distance function in thin narrow-bands around the vesicles

7. Update the N reconstituted level set functions

6.4 Numerical illustrations

This section is dedicated to the numerical results obtained with the proposed model. We present some numerical simulations of multiple vesicles in several type of external hydrodynamical flow in order to validate the proposed collision model which depends on the relative velocities between closest vesicles. For all the simulations presented in this section, the rebound parameter $\varepsilon_b = 2\varepsilon$. We do not present the computational cost according to the number of vesicles as the complexity of the algorithm is similar than in the case of rigid bodies.

6.4.1 Grid sensitivity

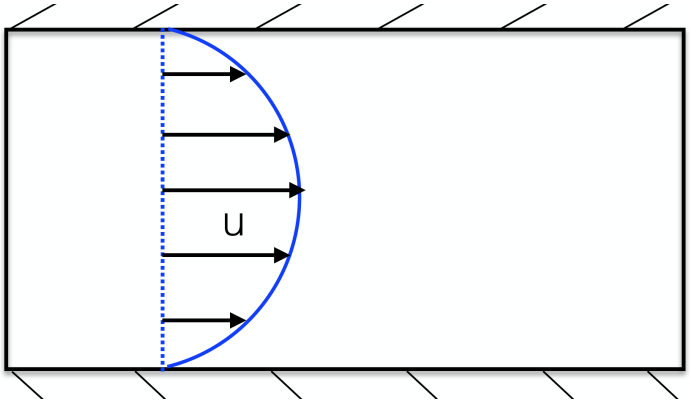


Figure 6.4: Poiseuille flow

A grid convergence is carried out in the case of two vesicles evolving in a Poiseuille type flow. The computational domain is a rectangle of size $[0, 2] \times [0, 1]$.
To study the grid convergence the simulations are performed on three grid levels (G_1, G_2,G_3) which contain respectively : (256×128), (512×256), (1025×512) cells on a uniform mesh. The regularization parameter ε is fixed to Δx^{G_1} where Δx^{G_1} denotes the mesh size corresponding to the coarsest grid $\Delta x^{G_1} \simeq 0.0078$. The three grids used to discretize the level set functions, label maps, distance functions and the stretching are twice finer than the three coarsest grid.

The velocity profile of the Poiseuille viscous fluid flow is parabolic as illustrated in Figure 6.4. At initialization, the shape of the vesicles are Cassini Oval and are defined using the implicit function:

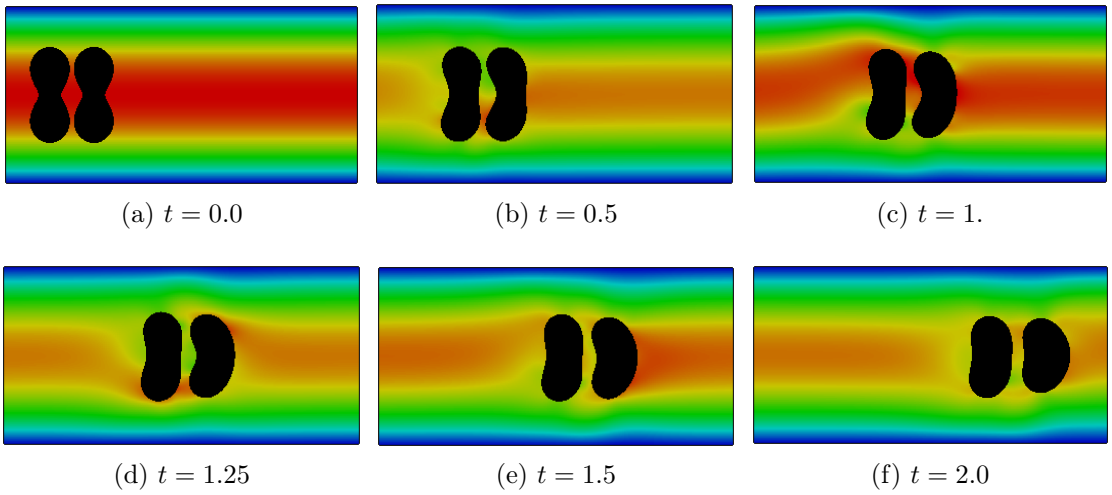
$$((x-a)^2 + y^2)((x+a)^2 + y^2) = b^4$$

For both vesicles we set the parameters $a = 0.18$ and $b = 0.1747$, the associated reduced area is around 0.8. Using a fast marching procedure we initialize ϕ to a signed distance function to the interfaces defined as the minimum of the two implicit functions. The physical parameters fixed in this study are:

$$R_e = 1.10^{-1}, W_e = 0.05, W_c = 200$$

At initialization, the vesicles are located at the entrance section of the domain. As illustrated in Figure 6.5, their shape progressively change during their motion in the canal. Figure 6.6 shows the results obtained with the three resolutions at different times. We

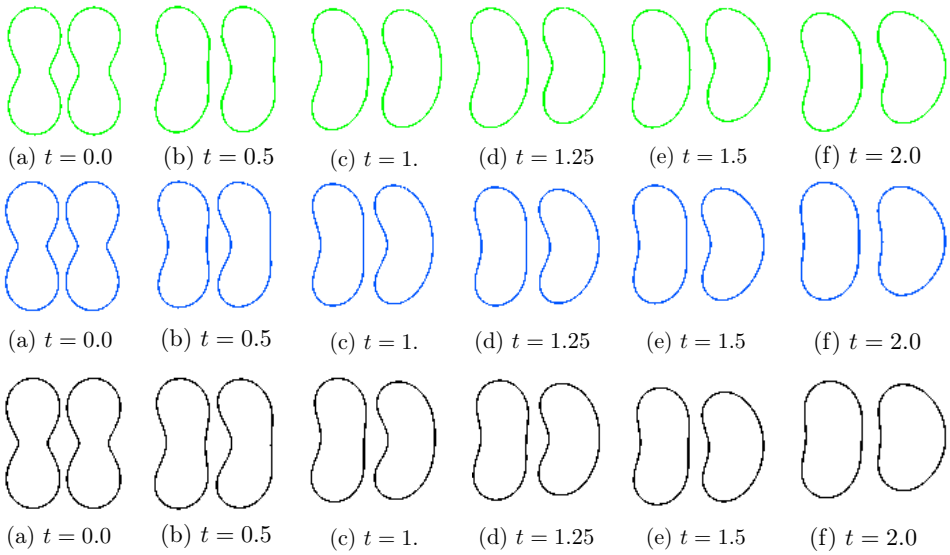
can see that the deformation of the vesicles are similar for the three resolutions, we note however that at time $t = 0.5$ the deformation of the two finer grids are closer than the one obtained with the coarsest grid. At the end of the simulations, the shape of the right vesicle corresponds to the equilibrium shape of a vesicle having a reduced area around 0.8 whereas the second vesicle adopts a parachute shape. In light of these results, the grid convergence is achieved for the coarsest grid.



(a) $t = 0.0$ (b) $t = 0.5$ (c) $t = 1.$

(d) $t = 1.25$ (e) $t = 1.5$ (f) $t = 2.0$

Figure 6.5: Deformation of two vesicles in a Poiseuille flow performed on the grid G_{512} of size (1024×512).



(a) $t = 0.0$ (b) $t = 0.5$ (c) $t = 1.$ (d) $t = 1.25$ (e) $t = 1.5$ (f) $t = 2.0$

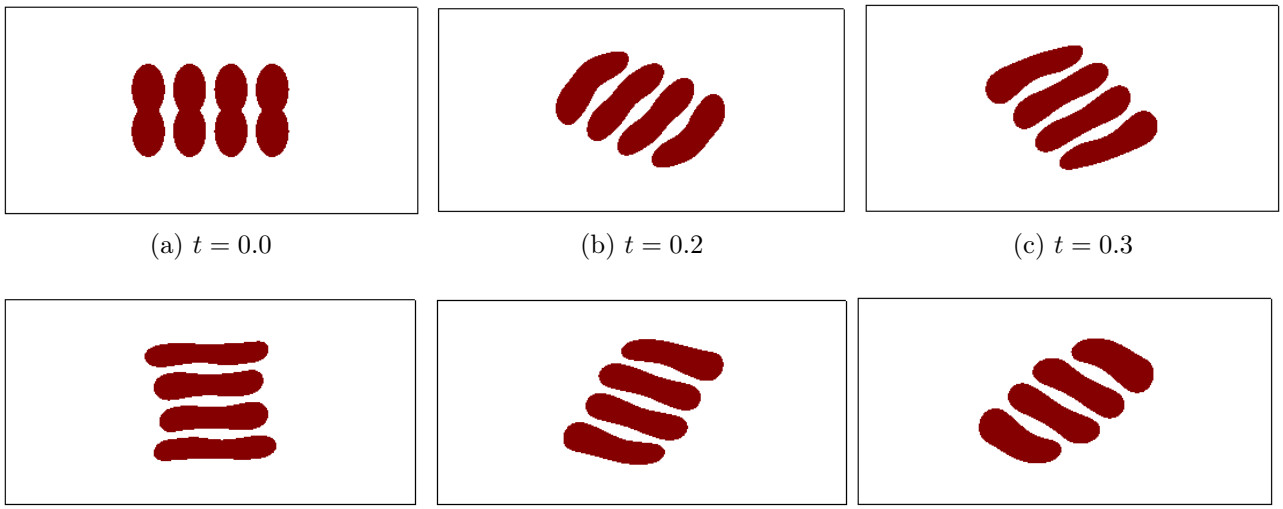
(a) $t = 0.0$ (b) $t = 0.5$ (c) $t = 1.$ (d) $t = 1.25$ (e) $t = 1.5$ (f) $t = 2.0$

(a) $t = 0.0$ (b) $t = 0.5$ (c) $t = 1.$ (d) $t = 1.25$ (e) $t = 1.5$ (f) $t = 2.0$

Figure 6.6: Simulation of two vesicles in a Poiseuille flow performed on three grid levels. From top to bottom, the associated discretization space steps are :$h = 7.81 \times 10^{-3}$, $h \simeq 3.90 \times 10^{-3}$, and $h \simeq 1.95 \times 10^{-3}$.

6.4.2 Vesicles in a shear flow

Next, we consider four vesicles in a shear flow of strenght 100 giving a Reynolds number of $R_e = 10^{-2}$. The computational domain $[0, 2] \times [0, 1]$ is represented by a Cartesian grid of size (512×256), the membrane thickness $\varepsilon = 1.5h$ where $h = 3.9.10^{-2}$ is the discretization space step. Periodic boundary conditions in the x-direction and homogeneous Dirichlet boundary in the y direction are imposed to the fluid velocity. Short time results are shown in Figure 6.7. We can see that the vesicles rotate collectively in the domain while deforming, at $t = 0.53$ the two centered vesicles have a biconcave shape, moreover the symetry is kept.



(a) $t = 0.0$ (b) $t = 0.2$ (c) $t = 0.3$

(d) $t = 0.53$ (e) $t = 0.7$ (f) $t = 0.75$

Figure 6.7: Deformation of four vesicles in a shear flow.

6.4.3 Sedimentation of two vesicles

We are interested here in the sedimentation of two vesicles in a viscous fluid flow. The computational domain Ω is a square of size $[0, 1] \times [0, 2]$. Simulations are performed on a grid of size (256×256) for the velocity and the pressure (the finer grid is hence of size (512×512)) and the half thickness of the interface is $\varepsilon = 1.5\Delta x$. The coefficient of gravity g is set to -9.8. The density ratio between the inner and outer fluids is set to 1.5.

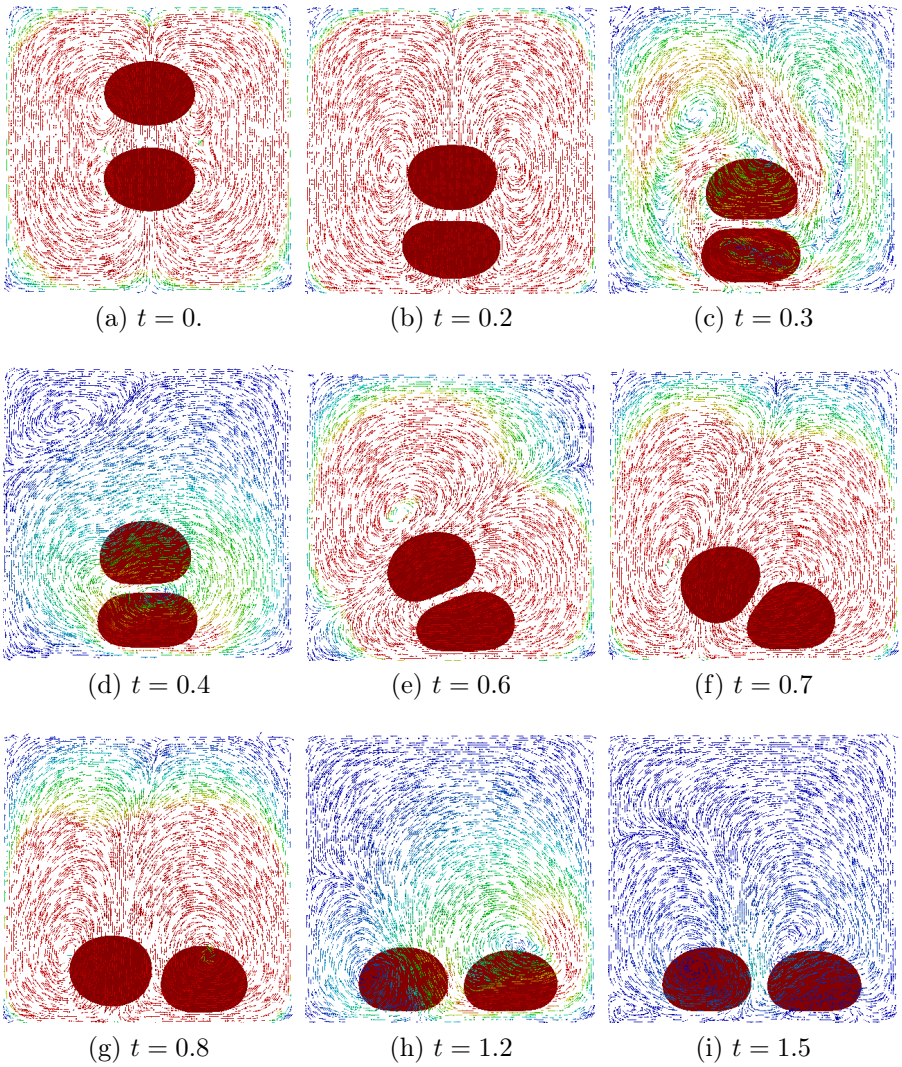
First, highly deformable vesicles are considered $W_e = 1$ and the bending modulus is very small so that the curvature forces are negligible. Results are shown on Figure 6.9. At initialization, the level set function ϕ is defined as a signed distance function to the interfaces of two ellipses of size $a = 0.15, b = 0.1$. We observe that the vesicle at the top of the domain, progressively deform and elongate while the velocity increases whereas the bottom vesicle at $t = 0.08$ adopts an ellipsoidal-cap shape. At $t = 0.13$, the two vesicles have reached the bottom wall, the collision model is active and avoid the vesicles from merging and being in contact with the wall, the two vesicles have a biconcave shape. Then, their shapes progressively change, the shape of the top vesicle far away from the second vesicle is spherical in order to maximize the area in contact with the bottom vesicle. Finally, a symmetry loss leads to a tumbling of the vesicle, the two vesicles continue

to deform until an equilibrium state is reached. The final shape of the vesicle is a typical shape deformation induced by gravity [83].

For the second test case, we consider vesicles which are highly inextensible and a high bending modulus, we set the following parameters:

$$R_e = 0.01, W_e = 0.005, W_c = 30$$

Results of the simulations are shown in Figure 6.8. Contrary to the first simulation, the vesicles deflect very slightly, the obtained deformed shape are different. Different vortices are created in the domain and in the narrow-gap between the two vesicles. Moreover, we can see that the intensity of the repulsive force is higher as compared to the first simulation the minimal distance kept between the two vesicles is at least twice smaller, this is due to the bending forces. The symmetry loss leads quickly to a tumbling of the vesicles. The final shapes are similar for the two simulations.

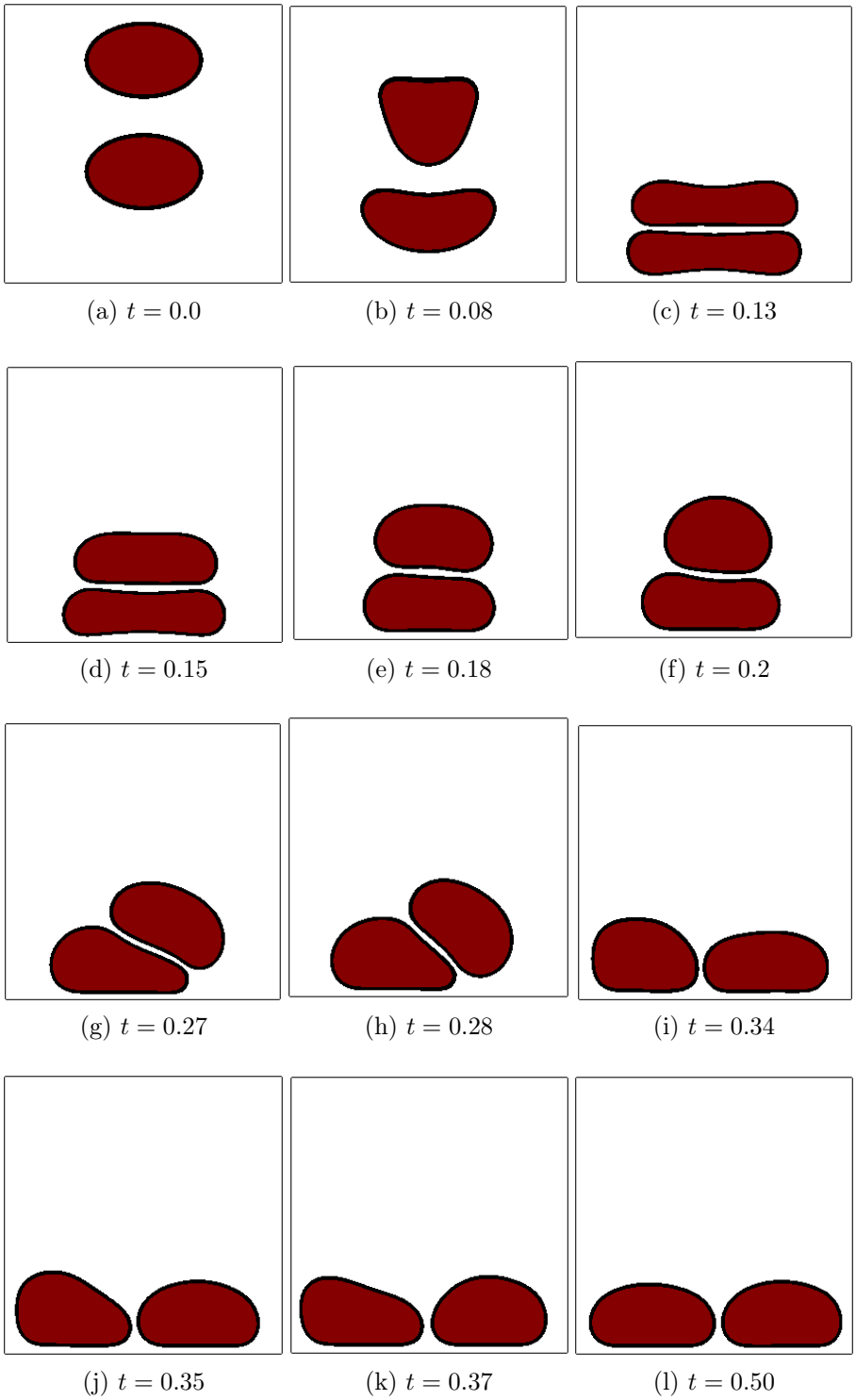


(a) $t = 0.$ (b) $t = 0.2$ (c) $t = 0.3$

(d) $t = 0.4$ (e) $t = 0.6$ (f) $t = 0.7$

(g) $t = 0.8$ (h) $t = 1.2$ (i) $t = 1.5$

Figure 6.8: Simulation of two vesicles falling under gravity. The background color shows the magnitude of the vorticity. (from blue to red).



(a) $t = 0.0$ (b) $t = 0.08$ (c) $t = 0.13$

(d) $t = 0.15$ (e) $t = 0.18$ (f) $t = 0.2$

(g) $t = 0.27$ (h) $t = 0.28$ (i) $t = 0.34$

(j) $t = 0.35$ (k) $t = 0.37$ (l) $t = 0.50$

Figure 6.9: Simulation of two vesicles falling under gravity. $R_e = 100, W_e = 1, W_c = 10^4$.

6.4.4 Vesicles passing through a bifurcation

We consider here the case of vesicles passing through a bifurcation, the velocity has a parabolic profile at the entrance and outlet sections.The computational domain $\Omega = [0, 2] \times [0, 1]$. In the case of a bifurcation, the proposed collision model failed at avoiding numerical contacts between vesicles and the wall. Instead, the intensity of the force is fixed, numerically we found that a coefficient $k = 10$ is sufficient to avoid contacts. Between two vesicles, the dependence of the collision model on relative velocities is kept. Two simulations are presented, for both simulations, we set the following parameters:

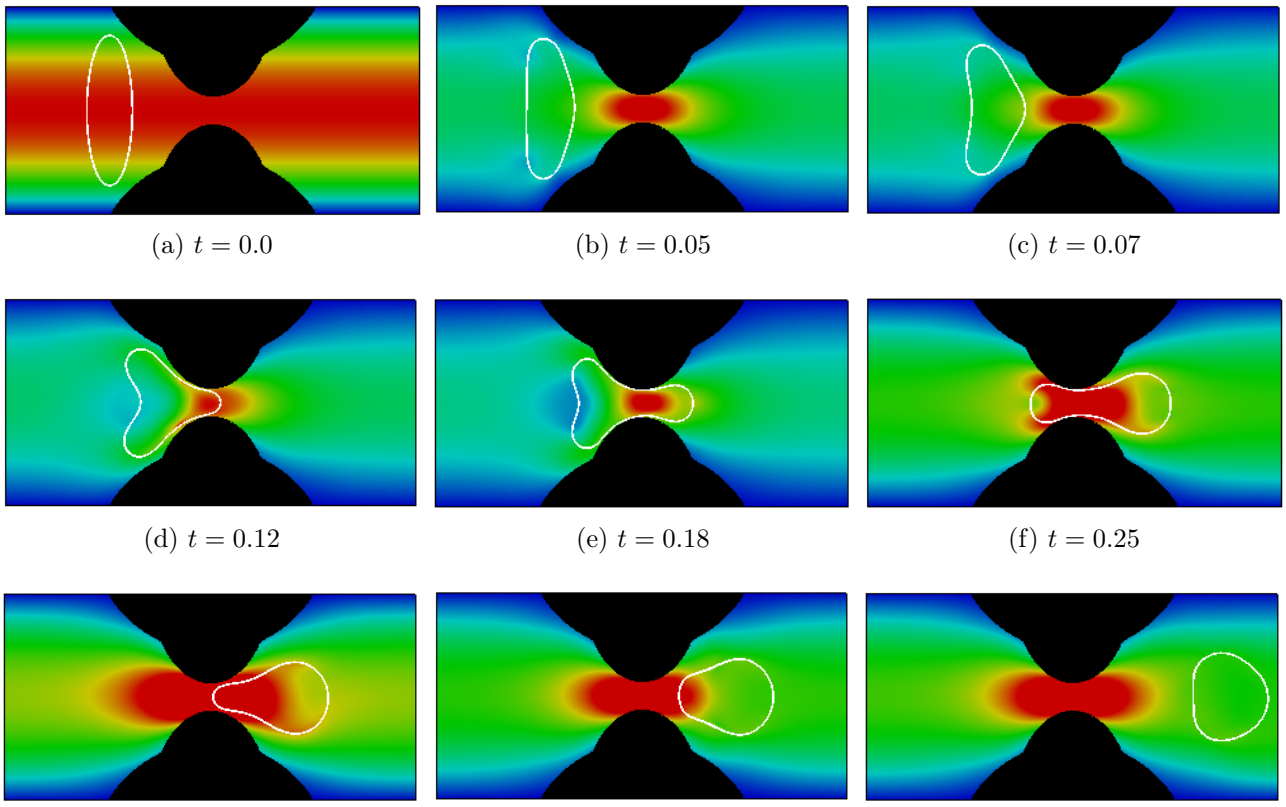
$$R_e = 0.01, W_e = 0.005, W_c = 30$$

The first simulation represented on Figure 6.10 is performed on a grid of size (1024×512). The initial shape of the vesicle is an ellipse of size $a = 0.1$ and $b = 0.4$, the height of the bifurcation is around four times smaller than the vesicle height ($b = 0.8$). We can see that the vesicle progressively deforms while passing through the bifurcation. During the deformation, the vesicle adopts different shape in order to pass through the channel. Finally, at the end of the simulation the vesicle has a parachute shape which is a typical shape of vesicle deformed in a Poiseuille type flow.

We also consider a test case of four vesicles passing through a rectangular bifurcation. At initialization, the interfaces are defined using an implicit function which corresponds to a Cassini Oval shape. The short-range repulsive force between the closest vesicles depends on the relative velocities. The obtained results are represented on Figure 6.11, the vesicles progressively advanced in the channel while deforming. At $t = 0.4$, the vesicles reached the bifurcation, then the right bottom vesicle goes up to pass through the bifurcation. Due to the repulsive forces, the vesicles located at the top stays at the top of the domain where the velocity magnitude is small. After passing through the bifurcation, the deformed vesicles have a circular shape.

6.4.5 Multiple vesicles in a Poiseuille flow

Finally, to present the capability of the method to deal with a large number of vesicles, we present a test case of 105 vesicles evolving in a Poiseuille flow. The computational domain $\Omega = [0, 4] \times [0, 2]$, as the size of the vesicles is small, the simulations are performed using a fine resolution, the grid contains (2048×1024) cells on an uniform mesh grid. As the level set grid is twice finer, the induced computational time of this simulation is very high, around 50 seconds per iteration. The obtained results are represented on Figure 6.12, colors indicate the value of the label map L_0,the white lines correspond to the isolines $\phi = 0$. At initialization, the region occupied by the vesicles represents around one-half of the computational domain, the configuration of the vesicle is seven layers of 15 vesicles. Each vesicle interface corresponds to a Cassini Oval with the parameters $a = 0.076$ and $b = 0.08$. We observe that depending on the layer, the vesicles adopt different shapes, these shapes are on one hand due to the pressure driven Poiseuille flow, to the elastic and bending forces and the interactions between the vesicles. At $t = 0.14$, we can see that the bottom blue left vesicle and the top red left vesicle advanced slowly than the other vesicles and hence interacts with their neighbors, the symmetry is lost and while the cells advances in the canal various shapes can be observed.

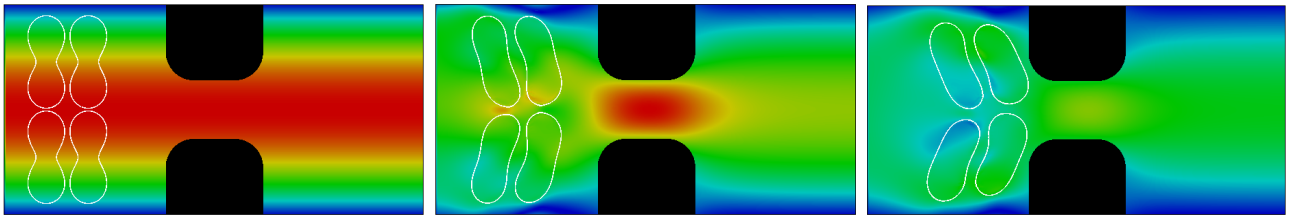


(a) $t = 0.0$ (b) $t = 0.05$ (c) $t = 0.07$

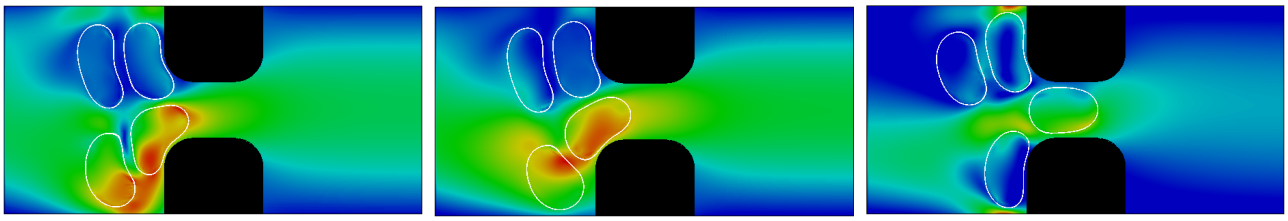
(d) $t = 0.12$ (e) $t = 0.18$ (f) $t = 0.25$

(g) $t = 0.28$ (h) $t = 0.30$ (i) $t = 0.41$

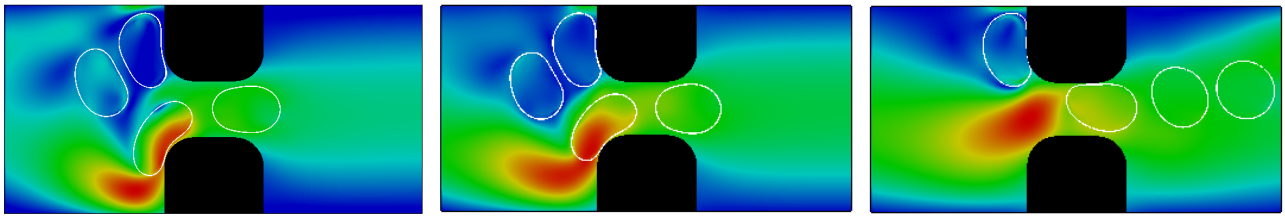
Figure 6.10: Simulation of a vesicle passing through a bifurcation. The background color corresponds to the magnitude of the velocity. The white contour represents the isoline $\phi = \varepsilon$.



(a) $t = 0.0$ (b) $t = 0.2$ (c) $t = 0.4$

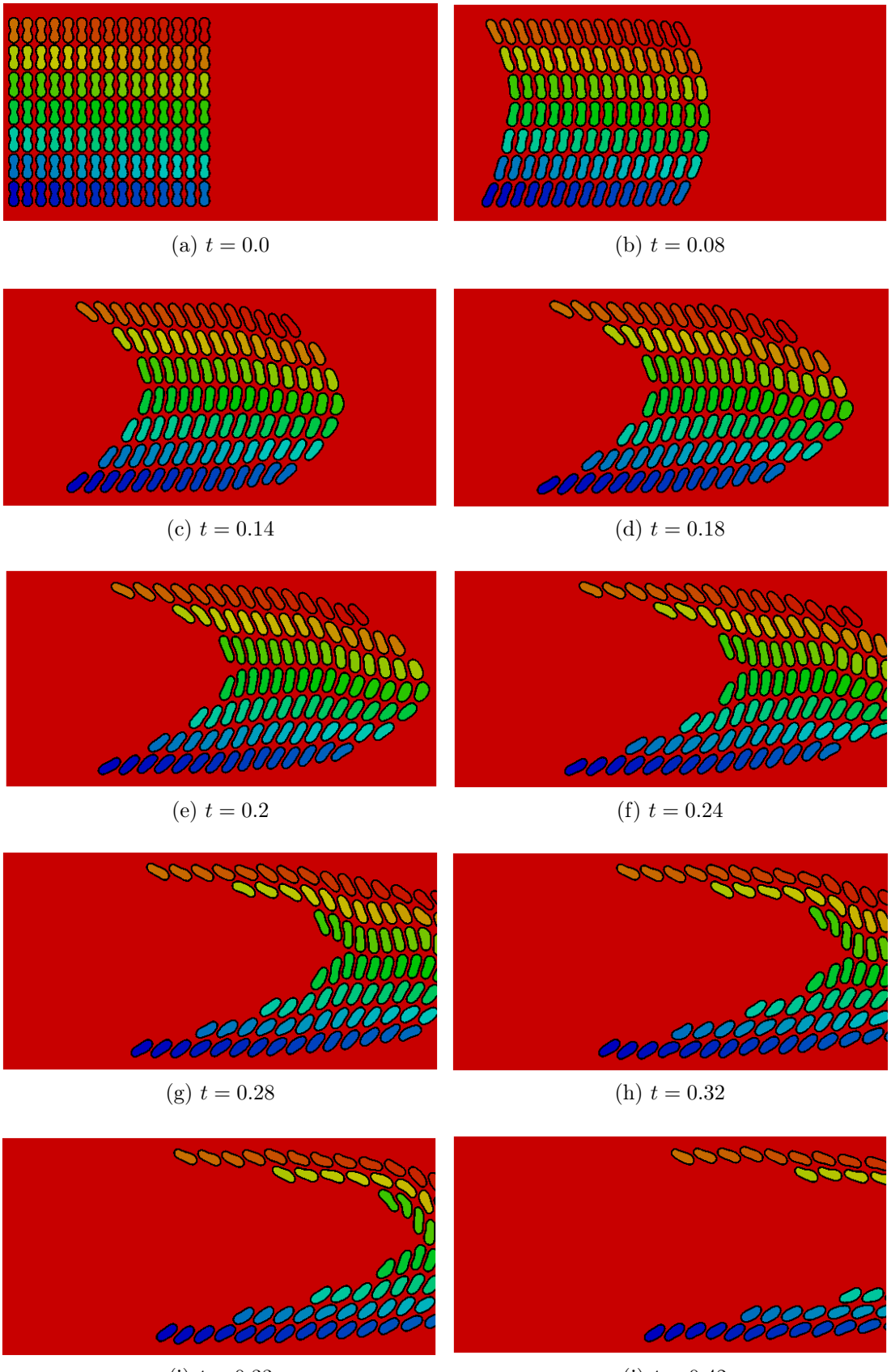


(d) $t = 0.8$ (e) $t = 0.95$ (f) $t = 1.3$



(g) $t = 2.15$ (h) $t = 2.5$ (i) $t = 3.25$

Figure 6.11: Simulation of four vesicles passing through a bifurcation. The background color corresponds to the magnitude of the velocity. The white contour represents the isoline $\phi = \varepsilon$.



(a) $t = 0.0$ (b) $t = 0.08$

(c) $t = 0.14$ (d) $t = 0.18$

(e) $t = 0.2$ (f) $t = 0.24$

(g) $t = 0.28$ (h) $t = 0.32$

(i) $t = 0.33$ (j) $t = 0.42$

Figure 6.12: Simulation of 105 vesicles in a Poiseuille flow. The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 105^{th} body and red for the fluid that is the 106^{th} object.

Conclusion

In this chapter, an application of the method to immersed deformable vesicles have been proposed. Using the level set function ϕ, the stretching e and the label maps one elastic and bending forces are computed to impose the inextensibility of the whole set of vesicles membranes as well as their resistance to bending. To handle numerical contacts, the relative velocities are computed by using the averaged fluid velocity on each vesicle membrane. Then, these obtained relative velocities provide the intensity of the short-range repulsive forces. A convergence study in the case of two vesicles evolving in a Poiseuille is presented. The dynamic behavior of vesicles under different boundary conditions have been presented and confirms the capability of the proposed method to deal with numerical contacts between vesicles at low resolution. In the case of vesicles passing through a bifurcation, the short-range repulsive force failed at avoiding contacts between vesicles and the wall, a constant intensity of the force has been imposed.

Conclusion

In this thesis, we have presented a numerical framework for the simulation of rigid and deformable bodies moving in a fluid. A fully Eulerian formulation of the fluid-structure interaction has been employed and combined to an efficient capturing method derived from the level set approach and a model introduced for image segmentation in [11]. Within this framework, a short-range repulsive force inspired by [31] is introduced in order to handle numerical contacts between the structures. We then applied this method to the simulation of rigid bodies and deformable vesicles in flow.

Chapter 1 was dedicated to an overview of the mathematical and numerical methods chosen. We described the level set method and its advantages for the simulation to multiple rigid bodies and vesicles. We also presented the projection method of Chorin type which we used to solve the fluid equations as well as the discretization of the fluid/structure domain.

In Chapter 2, we presented the capturing method which consists in three label maps and two distance functions. These five field functions allow to capture the whole set of structures interfaces while providing the distance between closest structures at all points of the computational domain. Moreover, the level set functions associated to each interface are recovered explicitly according to these functions.

In Chapter 3, we described the numerical strategy used to avoid numerical contacts between structures. A short-range repulsive force derived from [31] which depends on the two distance functions is introduced. This short-range repulsive force is applied on the surface of each structure and takes into account the interaction between closest cells. The intensity of this force is proportional to the relative velocities between closest bodies which are computed by means of the label maps. By considering multiple disks and a constant intensity of the force, the impact of the proposed repulsive force and the collision model [31] on the solution of two steady Stokes problems was investigated. We found that the error estimation between these two Stokes model depends on the size, the number of disks and the interface thickness of the disks where the forces are applied.

Chapter 4 was dedicated to the numerical procedure adopted to evolve the label maps and distance functions. As the repulsive force is only applied when the particles are closed, the evolution is only achieved in a thin-band around the bodies boundaries. A level set function which captures the whole set of structures interfaces is advected and allows to evolve explicity the label functions L_0 and L_1 close to the interfaces. Then, a multi-label fast marching algorithm is performed in the vicinity of the particles to provide φ_1 and L_1 and update the level set function as a signed distance function. When two bodies become close, a procedure to compute the distance between closest cells is activated.

Finally, we presented two applications of this method.

In Chapter 5, the rigid case was investigated. To take into account the rigid bodies, a penalization method was used and we proposed a penalization model which only depends on the label maps and distance functions. This model is compared to the penalization model used in [31] in a velocity and pressure formulation. The numerical results confirmed the proposition 1 presented in Chapter 3, the error estimation between the two collision models depend on the size and the number of particles. Moreover, compared to this model which is totally dependent on the number of bodies, the proposed penalization model substancially reduces the CPU time. We also made a comparison between the influence of the proposed collision model and the contact algorithm proposed by B. Maury. During numerical calculations, compared to the contact algorithm the symmetry is kept for a longer time with our collision model. However, on contrary to this contact algorithm, the proposed method does not allow to reproduce particle clustering. A comparison of simulations performed either with a collision model dependent on relative velocities or having a prescribed intensity is carried out. Due to the inherent complexity of the system, the dynamic of the particles is different, we observed however a similar global motion for both models.

The last chapter illustrated the capability and efficiency of the method to handle numerical contacts between deformable vesicles at low resolution. Using the level set function ϕ, the stretching e and the label maps one elastic and bending force are computed to impose the inextensibility of the whole set of vesicles membranes as well as their resistance to bending. To handle numerical contacts, the relative velocities are computed by using the averaged fluid velocity on each vesicle membrane. Then, these obtained relative velocities provide the intensity of the short-range repulsive forces. A convergence study in the case of two vesicles evolving in a Poiseuille is presented. The dynamic behavior of vesicles under different boundary conditions have been presented and confirms the capability of the proposed model to deal with numerical contacts between vesicles at low resolution. In the case of vesicles passing through a bifurcation, the short-range repulsive force failed at avoiding contacts between vesicles and the wall, a constant intensity of the force has hence been imposed.

This work offers many different perspectives and several improvements could be achieved.

First, even if we have lowered the computational cost of the multi-label fast marching, this procedure remains the most time consuming procedure when dealing with a high number of rigid bodies or vesicles, this is even worst when using finer grids to discretize the computational domain and in the three-dimensional case.
To reduce the computational cost of this procedure a parallel implementation seems necessary.
Recently, an interesting and highly efficient parallel implementation of the fast marching method has been proposed in [189]. This technique based on domain decomposition, is easy to implement and has similarities with the sequential narrow-band fast marching algorithm. The main idea is to partition the computational domain into several subdomains, each subdomain being mapped to one process. Each subdomain has its own heap structure and proceeds independently. At the points shared by different subdomains, each

process has to exchange information with all neighboring processes. And as the usual fast marching, a grid point is visited only once so that a grid point shared by two or several process is visited by only one of them.
In the case of several interfaces, the exchange between the processes will only occurs for particles shared by several subdomains, and as the interfaces are only propagated on thin band around particles, these exchanges will not be excessive. Moreover, as the multi-label fast marching can be viewed as a local fast marching for different regions of the computational domain (between closest bodies), an adaptation of this parallel implementation could offer a large computational saving.

An other important point which has to be improved concerns the collision model. The proposed short-range repulsive force allows to avoid numerical contact by mimic the behaviour of the normal component of the lubrication force. In future works, it would be interesting to study the influence of a tangential component of this force. Moreover, to take into account the underlying physics, a lubrication model should be included at the grid points contained in the interstitial gap between two closest particles, as even if the contact is avoided at this point the fluid equations at these points are not accurately solved.
We have presented an error estimation between the present repulsive force and the collision model [31] through simplified models of steady Stokes equations in two dimensions. In future works, it could be interesting to study problems which involve the transport equations of the level set functions. This will be a first step to study theoretically the influence of the two collision models on immersed structures (rigid, fluid or deformable bodies). First simplified models consist in replacing the advection of the gravity centers in each Stokes equations by transport equations of the level set functions as:

$$\partial_t \phi_i^1(x,t) + (U_1 \cdot \nabla \phi_i^1)(x - \phi_i^1 \nabla \phi_i^1, t) = 0, \partial_t \phi_i^2(x,t) + (U_2 \cdot \nabla \phi_i^2)(x - \phi_i^2 \nabla \phi_i^2, t) = 0$$

With these advection equations, the interfaces are still evolved with the right fluid velocity and the level set functions are at each time distance functions.

Finally, the proposed method could be used as a basis to study the behavior of vesicles suspensions. In this work, we only presented an illustration of the method to highlight its efficiency and ability to deal with deformable bodies in two dimension. An extension to dimension three and most realistic simulations could be performed and compared to experiments.

List of figures

List of Tables

Bibliography

[1] M Abkarian, C Lartigue, and A Viallat. Tank treading and unbinding of deformable vesicles in shear flow: determination of the lift force. *Physical review letters*, 88(6):068103, 2002. 135, 136

[2] Z Adamczyk, M Adamczyk, and TGM Van de Ven. Resistance coefficient of a solid sphere approaching plane and curved boundaries. *Journal of colloid and interface science*, 96(1):204–213, 1983. 6, 52

[3] D M Anderson, Geoffrey B McFadden, and Adam A Wheeler. Diffuse-interface methods in fluid mechanics. *Annual review of fluid mechanics*, 30(1):139–165, 1998. 23

[4] P Angot, C-H Bruneau, and P Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81(4):497–520, 1999. 13, 91

[5] C Antoci, M Gallati, and S Sibilla. Numerical simulation of fluid–structure interaction by sph. *Computers & Structures*, 85(11):879–890, 2007. 10

[6] S Bertoluzza, M Ismail, and B Maury. Analysis of the fully discrete fat boundary method. *Numerische Mathematik*, 118(1):49–77, 2011. 15

[7] T Biben, K Kassner, and C Misbah. Phase-field approach to three-dimensional vesicle dynamics. *Physical Review E*, 72(4):041921, 2005. 138, 139

[8] T Biben and C Misbah. An advected-field method for deformable entities under flow. *The European Physical Journal B-Condensed Matter and Complex Systems*, 29(2):311–316, 2002.

[9] T Biben and C Misbah. Tumbling of vesicles under shear flow within an advected-field approach. *Physical Review E*, 67(3):031908, 2003. 138

[10] GW Scott Blair. The importance of the sigma phenomenon in the study of the flow of blood. *Rheologica Acta*, 1(2-3):123–126, 1958. 134

[11] J A Bogovic, J. L Prince, and P-L Bazin. A multiple object geometric deformable model for image segmentation. *Computer Vision and Image Understanding*, 117(2):145–157, 2013. 7, 40, 42, 77, 158

[12] Claire Bost. *Méthodes Level-Set et pénalisation pour le calcul d'interactions fluide-structure*. PhD thesis, Université Joseph-Fourier-Grenoble I, 2008. 147

[13] F Bouchon, T Dubois, and N James. A second-order cut-cell method for the numerical simulation of 2d flows past obstacles. *Computers & Fluids*, 65:80–91, 2012. 18

[14] J Boussinesq. *Théorie analytique de la chaleur: mise en harmonie avec la thermodynamique et avec la théorie mécanique de la lumière*, volume 2. Gauthier-Villars, 1903. 21

[15] JU Brackbill, Douglas B Kothe, and Charles Zemach. A continuum method for modeling surface tension. *Journal of computational physics*, 100(2):335–354, 1992. 146

[16] H Brenner. The slow motion of a sphere through a viscous fluid towards a plane surface. *Chemical engineering science*, 16(3-4):242–251, 1961. 51, 53

[17] W-P Breugem. A combined soft-sphere collision/immersed boundary method for resolved simulations of particulate flows. In *ASME 2010 3rd Joint US-European Fluids Engineering Summer Meeting collocated with 8th International Conference on Nanochannels, Microchannels, and Minichannels*, pages 2381–2392. American Society of Mechanical Engineers, 2010. 54

[18] W-P Breugem. A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows. *Journal of Computational Physics*, 231(13):4469–4498, 2012. 13

[19] Jonathan M Bricker and Jason E Butler. Oscillatory shear of suspensions of non-colloidal particles. *Journal of Rheology (1978-present)*, 50(5):711–728, 2006. 55

[20] J-P Caltagirone. Sur l'intéraction fluide-milieu poreux; application au calcul des efforts exercés sur un obstacle par un fluide visqueux. *Comptes rendus de l'Académie des sciences. Série II, Mécanique, physique, chimie, astronomie*, 318(5):571–577, 1994. 13

[21] P B Canham. The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell. *Journal of theoretical biology*, 26(1):61–81, 1970. 136, 141

[22] I Cantat and C Misbah. Lift force and dynamical unbinding of adhering vesicles under shear flow. *Physical review letters*, 83(4):880, 1999. 135, 136

[23] V Caselles, F Catté, T Coll, and F Dibos. A geometric model for active contours in image processing. *Numerische mathematik*, 66(1):1–31, 1993. 40

[24] Philippe Chatelain, Alessandro Curioni, Michael Bergdorf, Diego Rossinelli, Wanda Andreoni, and Petros Koumoutsakos. Billion vortex particle direct numerical simulations of aircraft wakes. *Computer Methods in Applied Mechanics and Engineering*, 197(13):1296–1304, 2008. 11

[25] Y Cheny and O Botella. The ls-stag method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties. *Journal of Computational Physics*, 229(4):1043–1076, 2010. 18

[26] B Cichocki, ML Ekiel-Jeżewska, and E Wajnryb. Lubrication corrections for three-particle contribution to short-time self-diffusion coefficients in colloidal dispersions. *The Journal of chemical physics*, 111(7):3265–3273, 1999. 54

[27] D K Clarke, HA Hassan, and MD Salas. Euler calculations for multielement airfoils using cartesian grids. *AIAA journal*, 24(3):353–358, 1986. 18

[28] A Colagrossi and M Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics*, 191(2):448–475, 2003. 10

[29] C Conca, A Osses, and J Planchard. Added mass and damping in fluid-structure interaction. *Computer methods in applied mechanics and engineering*, 146(3):387–405, 1997. 11

[30] MDA Cooley and ME O'neill. On the slow motion generated in a viscous fluid by the approach of a sphere to a plane wall or stationary sphere. *Mathematika*, 16(01):37–49, 1969. 51, 52, 53

[31] M Coquerelle and G-H Cottet. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *Journal of Computational Physics*, 227(21):9121–9137, 2008. 7, 8, 49, 55, 57, 59, 73, 92, 94, 95, 100, 132, 158, 159, 160

[32] G-H Cottet and E Maitre. A level-set formulation of immersed boundary methods for fluid-structure interaction problems. *Comptes Rendus Mathematique*, 338(7):581–586, 2004. 8, 13, 23, 29, 133, 139, 140

[33] G-H Cottet and E Maitre. A level set method for fluid-structure interactions with immersed surfaces. *Mathematical models and methods in applied sciences*, 16(03):415–438, 2006. 13, 29, 139, 140

[34] G-H Cottet and Philippe Poncet. Advances in direct numerical simulations of 3d wall-bounded flows by vortex-in-cell methods. *Journal of Computational Physics*, 193(1):136–158, 2004. 11

[35] RG Cox. The motion of suspended particles almost in contact. *International Journal of Multiphase Flow*, 1(2):343–371, 1974. 52

[36] A Dagan. Numerical consistency and spurious boundary layer in the projection method. *Computers & fluids*, 32(9):1213–1232, 2003. 39

[37] Bart J Daly. Numerical study of two fluid rayleigh-taylor instability. *Physics of Fluids (1958-1988)*, 10(2):297–307, 1967. 24

[38] SL Dance, E Climent, and MR Maxey. Collision barrier effects on the bulk flow in a random suspension. *Physics of Fluids (1994-present)*, 16(3):828–831, 2004. 55

[39] SL Dance and MR Maxey. Incorporation of lubrication effects into the force-coupling method for particulate two-phase flow. *Journal of computational Physics*, 189(1):212–238, 2003. 7, 53

[40] KH De Haas, C Blom, D Van den Ende, MHG Duits, and J Mellema. Deformation of giant lipid bilayer vesicles in shear flow. *Physical Review E*, 56(6):7132, 1997. 135, 136

[41] J De Hart, FPT Baaijens, GWM Peters, and PJG Schreurs. A computational fluid-structure interaction analysis of a fiber-reinforced stentless aortic valve. *Journal of biomechanics*, 36(5):699–712, 2003. 12

[42] A Decoene, S Martin, and B Maury. Microscopic modelling of active bacterial suspensions. *Mathematical Modelling of Natural Phenomena*, 6(5):98–129, 2011. 54

[43] C Diaz-Goano, P Minev, and K Nandakumar. A lagrange multipliers/fictitious domain approach for particulate flow. In *International Conference on Large-Scale Scientific Computing*, pages 409–416. Springer, 2001. 55

[44] J Donea, S Giuliani, and JP Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer methods in applied mechanics and engineering*, 33(1):689–723, 1982. 11

[45] V Doyeux. *Modelisation et simulation de systemes multi-fluides. Application aux ecoulements sanguins.* PhD thesis, Université de Grenoble, 2014. 142

[46] German Drazer, Joel Koplik, Boris Khusid, and Andreas Acrivos. Deterministic and stochastic behaviour of non-brownian spheres in sheared suspensions. *Journal of Fluid Mechanics*, 460:307–335, 2002. 55

[47] L Durlofsky, J F Brady, and G Bossis. Dynamic simulation of hydrodynamically interacting particles. *Journal of fluid mechanics*, 180:21–49, 1987. 54

[48] Marco Ellero, Mar Serrano, and Pep Espanol. Incompressible smoothed particle hydrodynamics. *Journal of Computational Physics*, 226(2):1731–1752, 2007. 10

[49] D Enright, R Fedkiw, J Ferziger, and I Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational physics*, 183(1):83–116, 2002. 32, 33

[50] S Faure, A Lefebvre-Lepot, and B Semin. Dynamic numerical investigation of random packing for spherical and nonconvex particles. In *ESAIM: Proceedings*, volume 28, pages 13–32. EDP Sciences, 2009. 54

[51] M A Fernández. Coupling schemes for incompressible fluid-structure interaction: implicit, semi-implicit and explicit. *SeMA Journal*, 55(1):59–108, 2011. 11

[52] M A Fernández. Incremental displacement-correction schemes for incompressible fluid-structure interaction. *Numerische Mathematik*, 123(1):21–65, 2013.

[53] M A Fernández, J-F Gerbeau, and C Grandmont. A projection algorithm for fluid–structure interaction problems with strong added-mass effect. *Comptes Rendus Mathematique*, 342(4):279–284, 2006. 11

[54] M A Fernández, J-F Gerbeau, and C Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007. 11

[55] M A Fernández, M Landajuela, and M Vidrascu. Fully decoupled time-marching schemes for incompressible fluid/thin-walled structure interaction. *Journal of Computational Physics*, 297:156–181, 2015. 11

[56] M A Fernández, J. Mullaert, and M Vidrascu. Generalized robin–neumann explicit coupling schemes for incompressible fluid-structure interaction: Stability analysis and numerics. *International Journal for Numerical Methods in Engineering*, 101(3):199–229, 2015. 11

[57] A L Fogelson and C S Peskin. A fast numerical method for solving the three-dimensional stokes' equations in the presence of suspended particles. *Journal of Computational Physics*, 79(1):50–69, 1988. 13

[58] Antonio F Fortes, Daniel D Joseph, and Thomas S Lundgren. Nonlinear mechanics of fluidization of beds of spherical particles. *Journal of Fluid Mechanics*, 177:467–483, 1987. 101

[59] T Franke, Ronald HW Hoppe, C Linsenmann, Lothar Schmid, C Willbold, and A Wixforth. Numerical simulation of the motion of red blood cells and vesicles in microfluidic flows. *Computing and visualization in science*, 14(4):167–180, 2011. 138

[60] Cédric Galusinski and Paul Vigneaux. Level-set method and stability condition for curvature-driven flows. *Comptes Rendus Mathematique*, 344(11):703–708, 2007. 146

[61] Mattia Gazzola, Philippe Chatelain, Wim M Van Rees, and Petros Koumoutsakos. Simulations of single and multiple swimmers with non-divergence free deforming geometries. *Journal of Computational Physics*, 230(19):7093–7114, 2011. 71

[62] G Ghigliotti, T Biben, and C Misbah. Rheology of a dilute two-dimensional suspension of vesicles. *Journal of Fluid Mechanics*, 653:489–518, 2010. 137

[63] G Ghigliotti, A Rahimian, G Biros, and C Misbah. Vesicle migration and spatial organization driven by flow line curvature. *Physical review letters*, 106(2):028101, 2011. 137

[64] R A Gingold and J J Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977. 10

[65] R Glowinski, T-W Pan, T I Hesla, and D D Joseph. A distributed lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow*, 25(5):755–794, 1999. 7, 16, 55

[66] R Glowinski, T-W Pan, and J Periaux. A fictitious domain method for external incompressible viscous flow modeled by navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 112(1):133–148, 1994. 39

[67] R Glowinski, TW Pan, TI Hesla, DD Joseph, and J Periaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. *Journal of Computational Physics*, 169(2):363–426, 2001. 16, 100

[68] R Glowinsky, A J Kearsley, T-W Pan, and J Periaux. Fictitious domain methods for viscous flow simulation. Technical report, DTIC Document, 1995. 17

[69] A J Goldman, R G Cox, and H Brenner. Slow viscous motion of a sphere parallel to a plane wall—i motion through a quiescent fluid. *Chemical engineering science*, 22(4):637–651, 1967. 51

[70] D Goldstein, R Handler, and L Sirovich. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105(2):354–366, 1993. 13

[71] D Goldstein, R Handler, and L Sirovich. Direct numerical simulation of turbulent flow over a modelled riblet-covered surface. *Journal of Fluid Mechanics*, 302(10):333–376, 1995. 13

[72] J Gomes and O Faugeras. Reconciling distance functions and level sets. *Journal of Visual Communication and Image Representation*, 11(2):209–223, 2000. 29

[73] C Grandmont and Y Maday. Fluid-structure interaction: a theoretical point of view. *Revue européenne des éléments finis*, 9(6-7):633–653, 2000. 11

[74] J Han and G Tryggvason. Secondary breakup of axisymmetric liquid drops. i. acceleration by a constant body force. *Physics of Fluids*, 11(12):3650–3667, 1999. 22

[75] Francis H Harlow, J Eddie Welch, et al. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of fluids*, 8(12):2182, 1965. 24

[76] W Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Zeitschrift für Naturforschung C*, 28(11-12):693–703, 1973. 136, 141

[77] M Hillairet. Lack of collision between solid bodies in a 2d incompressible viscous flow. *Communications in Partial Differential Equations*, 32(9):1345–1371, 2007. 6, 49

[78] M Hillairet and T Takahashi. Collision in 3d fluid structure interactions problems. *SIAM J. Math. Anal*, 2009. 49

[79] C W Hirt and B D Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981. 23, 25

[80] J Hron and S Turek. *A monolithic FEM/multigrid solver for an ALE formulation of fluid-structure interaction with applications in biomechanics*. Springer, 2006. 11

[81] Howard H Hu. Direct simulation of flows of solid-liquid mixtures. *International Journal of Multiphase Flow*, 22(2):335–352, 1996. 6, 7, 53

[82] XY Hu and Nikolaus A Adams. An incompressible multi-phase sph method. *Journal of computational physics*, 227(1):264–278, 2007. 10

[83] ZH Huang, Manouk Abkarian, and Annie Viallat. Sedimentation of vesicles: from pear-like shapes to microtether extrusion. *New Journal of Physics*, 13(3):035026, 2011. 151

[84] S Huberson and A Jolles. Correction de l'erreur de projection dans les méthodes particules/maillage. *La Recherche aérospatiale*, 4:1–6, 1990. 11

[85] Morton A Hyman. Non-iterative numerical solution of boundary-value problems. *Applied Scientific Research, Section B*, 2(1):325–351, 1952. 13

[86] M Ismail. *Méthode de la frontière élargie pour la résolution de problèmes elliptiques dans des domaines perforés. Application aux écoulements fluides tridimensionnels*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2004. 15

[87] D Jacqmin. Calculation of two-phase navier–stokes flows using phase-field modeling. *Journal of Computational Physics*, 155(1):96–127, 1999. 22

[88] J Janela, A Lefebvre, and B Maury. A penalty method for the simulation of fluid-rigid body interaction. In *ESAIM: Proceedings*, volume 14, pages 115–123. EDP Sciences, 2005. 117

[89] DJ Jeffrey. *Low-Reynolds-number flow between converging spheres*. Cambridge Univ Press, 1982. 52

[90] A A Johnson and T E Tezduyar. Simulation of multiple spheres falling in a liquid-filled tube. *Computer Methods in Applied Mechanics and Engineering*, 134(3):351–373, 1996. 53

[91] G Kabacaoglu, B Quaife, and G Biros. Low-resolution simulations of vesicle suspensions in 2d. *arXiv preprint arXiv:1702.04709*, 2017. 137

[92] V Kantsler, E Segre, and V Steinberg. Dynamics of interacting vesicles and rheology of vesicle suspension in shear flow. *EPL (Europhysics Letters)*, 82(5):58005, 2008. 136

[93] V Kantsler and V Steinberg. Orientation and dynamics of a vesicle in tank-treading motion in shear flow. *Physical review letters*, 95(25):258101, 2005. 135, 136

[94] V Kantsler and V Steinberg. Transition to tumbling and two regimes of tumbling motion of a vesicle in shear flow. *Physical review letters*, 96(3):036001, 2006. 136

[95] B Kaoui, N Tahiri, T Biben, H Ez-Zahraouy, A Benyoussef, G Biros, and C Misbah. Complexity of vesicle microcirculation. *Physical Review E*, 84(4):041906, 2011. 137

[96] G Em Karniadakis, M Israeli, and S A Orszag. High-order splitting methods for the incompressible navier-stokes equations. *Journal of computational physics*, 97(2):414–443, 1991. 39

[97] S R Keller and R Skalak. Motion of a tank-treading ellipsoidal particle in a shear flow. *Journal of Fluid Mechanics*, 120:27–47, 1982. 135

[98] T Kempe and J Fröhlich. An improved immersed boundary method with direct forcing for the simulation of particle laden flows. *Journal of Computational Physics*, 231(9):3663–3684, 2012. 54

[99] J Kim and P Moin. Application of a fractional-step method to incompressible navier-stokes equations. *Journal of computational physics*, 59(2):308–323, 1985. 39

[100] S Kim and SJ Karrila. *Microhydrodynamics: Principles and Selected Applications*. Butterworth-Heinemann, 1991. 52, 53, 60

[101] Y Kim and M-C Lai. Simulating the dynamics of inextensible vesicles by the penalty immersed boundary method. *Journal of Computational Physics*, 229(12):4840–4853, 2010. 138

[102] W KLIP. Effect of vagal stimulation and acetylcholine on the ventricle anomalous viscosity of blood and the" summation phenomenon". *Circulation research*, 9(6):1380–1383, 1961. 134

[103] Petros Koumoutsakos. Multiscale flow simulations using particles. *Annu. Rev. Fluid Mech.*, 37:457–487, 2005. 11

[104] Petros Koumoutsakos and A Leonard. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *Journal of Fluid Mechanics*, 296:1–38, 1995. 11

[105] A Lefebvre. Fluid-particle simulations with freefem++. In *ESAIM: Proceedings*, volume 18, pages 120–132. EDP Sciences, 2007. 4, 90, 117, 119, 120, 121, 163, 164

[106] A Lefebvre. *Modélisation numérique d'écoulements fluide/particules*. PhD thesis, Université Paris Sud-Paris XI, 2007. 56

[107] A Lefebvre. Numerical simulation of gluey particles. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43(1):53–80, 2009. 54

[108] A Lefebvre-Lepot, B Merlet, and TN Nguyen. An accurate method to include lubrication forces in numerical simulations of dense stokesian suspensions. *Journal of Fluid Mechanics*, 769:369–386, 2015. 54

[109] R J Leveque and Z Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(4):1019–1044, 1994. 13

[110] XZ Li, D Barthes-Biesel, and A Helmy. Large deformations and burst of a capsule freely suspended in an elongational flow. *Journal of fluid mechanics*, 187:179–196, 1988. 136

[111] Z Li. An overview of the immersed interface method and its applications. *Taiwanese journal of mathematics*, 7(1):pp–1, 2003. 13

[112] X-D Liu, S Osher, and T Chan. Weighted essentially non-oscillatory schemes. *Journal of computational physics*, 115(1):200–212, 1994. 35

[113] L B Lucy. A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82:1013–1024, 1977. 10

[114] M A Mader, V Vitkova, M Abkarian, A Viallat, and T Podgorski. Dynamics of viscous vesicles in shear flow. *The European Physical Journal E: Soft Matter and Biological Physics*, 19(4):389–397, 2006. 135, 136

[115] E Maitre, T Milcent, G-H Cottet, A. Raoult, and Y Usson. Applications of level set methods in computational biophysics. *Mathematical and Computer Modelling*, 49(11):2161–2169, 2009. 29, 141

[116] Emmanuel Maitre, Chaouqi Misbah, Philippe Peyla, and Annie Raoult. Comparison between advected-field and level-set methods in the study of vesicle dynamics. *Physica D: Nonlinear Phenomena*, 241(13):1146–1157, 2012. 139

[117] SR Majumdar. Slow motion of an incompressible viscous liquid generated by the rotation of two spheres in contact. *Mathematika*, 14(1):43–46, 1967. 52

[118] A A Mammoli. The treatment of lubrication forces in boundary integral equations. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 462, pages 855–881. The Royal Society, 2006. 55

[119] AJ Markvoort, RA Van Santen, and PAJ Hilbers. Vesicle shapes from molecular dynamics simulations. *The Journal of Physical Chemistry B*, 110(45):22780–22785, 2006. 136

[120] A_ D_ Maude. End effects in a falling-sphere viscometer. *British Journal of Applied Physics*, 12(6):293, 1961. 51

[121] B Maury. A many-body lubrication model. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 325(9):1053–1058, 1997. 7, 53, 54

[122] B Maury. Direct simulations of 2d fluid-particle flows in biperiodic domains. *Journal of computational physics*, 156(2):325–351, 1999. 54

[123] B Maury. A fat boundary method for the poisson problem in a domain with holes. *Journal of scientific computing*, 16(3):319–339, 2001. 15

[124] B Maury. A time-stepping scheme for inelastic collisions. *Numerische Mathematik*, 102(4):649–679, 2006. 54, 90, 117

[125] B Maury. A gluey particle model. In *ESAIM: Proceedings*, volume 18, pages 133–142. EDP Sciences, 2007. 54

[126] B Maury. Numerical analysis of a finite element/volume penalty method. *SIAM Journal on Numerical Analysis*, 47(2):1126–1148, 2009. 14

[127] B Maury and J Venel. Handling of contacts in crowd motion simulations. *Traffic and Granular Flow'07*, pages 171–180, 2009. 54

[128] A Meunier and G Bossis. The influence of surface forces on shear-induced tracer diffusion in mono and bidisperse suspensions. *The European Physical Journal E*, 25(2):187–199, 2008. 53

[129] T Milcent. *Une approche eulérienne du couplage fluide-structure, analyse mathématique et applications en biomécanique*. PhD thesis, Université Joseph-Fourier-Grenoble I, 2009. 139, 140

[130] P Moin and J Kim. On the numerical solution of time-dependent viscous incompressible fluid flows involving solid boundaries. *Journal of computational physics*, 35(3):381–392, 1980. 39

[131] J J Monaghan. Simulating free surface flows with sph. *Journal of computational physics*, 110(2):399–406, 1994. 10

[132] JJ Monaghan. Extrapolating b splines for interpolation. *Journal of Computational Physics*, 60(2):253–262, 1985. 71

[133] S Nasseri, N Phan-Thien, and X-J Fan. Lubrication approximation in completed double layer boundary element method. *Computational Mechanics*, 26(4):388–397, 2000. 54

[134] ME ONeill and SR Majumdar. Asymmetrical slow viscous uid motions caused by the translation or rotation of two spheres. part ii: Asymptotic forms of the solutions when the minimum clearance between the spheres approaches zero. *Z. Angew. Math. Phys*, 21:180187, 1970. 52

[135] ME O'neill and K Stewartson. On the slow motion of a sphere parallel to a nearby plane wall. *Journal of Fluid Mechanics*, 27(04):705–724, 1967. 51

[136] S A Orszag, M Israeli, and M O Deville. Boundary conditions for incompressible flows. *Journal of Scientific Computing*, 1(1):75–111, 1986. 39

[137] S Osher and R Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006. 26

[138] S Osher and J A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988. 26

[139] N Parolini. Computational fluid dynamics for naval engineering problems. *Doctoral dissertation, Politecnico di Milano*, 2004. 29

[140] NA Patankar. A formulation for fast computations of rigid particulate flows. *Center for Turbulence Research Annual Research Briefs*, 2001:185–196, 2001. 14, 91

[141] C S Peskin. *Flow Patterns around Heart Valves: A digital computer method to solve the equations of motion*. PhD thesis, PhD thesis, Albert Eistein College of Medicine, 1972. 11, 12

[142] C S Peskin. Flow patterns around heart valves: a numerical method. *Journal of computational physics*, 10(2):252–271, 1972. 11

[143] C S Peskin. Numerical analysis of blood flow in the heart. *Journal of computational physics*, 25(3):220–252, 1977. 12, 28

[144] C Pozrikidis. The axisymmetric deformation of a red blood cell in uniaxial straining stokes flow. *Journal of Fluid Mechanics*, 216:231–254, 1990. 136

[145] C Pozrikidis. On the transient motion of ordered suspensions of liquid drops. *Journal of Fluid Mechanics*, 246:301–320, 1993. 136

[146] C Pozrikidis. Effect of membrane bending stiffness on the deformation of capsules in simple shear flow. *Journal of Fluid Mechanics*, 440:269–291, 2001. 11, 136

[147] B Quaife and G Biros. High-volume fraction simulations of two-dimensional vesicle suspensions. *Journal of Computational Physics*, 274:245–267, 2014. 137

[148] S Ramanujan and C Pozrikidis. Deformation of liquid capsules enclosed by elastic membranes in simple shear flow: large deformations and the effect of fluid viscosities. *Journal of Fluid Mechanics*, 361:117–143, 1998. 136

[149] T Richter and T Wick. Finite elements for fluid–structure interaction in ale and fully eulerian coordinates. *Computer Methods in Applied Mechanics and Engineering*, 199(41):2633–2642, 2010. 11

[150] W J Rider and D B Kothe. Reconstructing volume tracking. *Journal of computational physics*, 141(2):112–152, 1998. 25

[151] E Rouy and A Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29(3):867–884, 1992. 30, 78

[152] EM Saiki and S Biringen. Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. *Journal of Computational Physics*, 123(2):450–465, 1996. 13

[153] D Salac and M Miksis. A level set projection model of lipid vesicles in general flows. *Journal of Computational Physics*, 230(22):8192–8215, 2011. 139

[154] V K Saul'ev. On the solution of some boundary value problems on high performance computers by fictitious domain method. *Siberian Math. J*, 4(4):912–925, 1963. 13

[155] U Seifert. Configurations of fluid membranes and vesicles. *Advances in physics*, 46(1):13–137, 1997. 134

[156] J A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. 29, 30, 77

[157] J A Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, 1999. 26

[158] N Sharma and N-A Patankar. A fast computation technique for the direct numerical simulation of rigid particulate flows. *Journal of Computational Physics*, 205(2):439–457, 2005. 16, 17

[159] C-W Shu and S Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988. 35

[160] C-W Shu and S Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii. In *Upwind and High-Resolution Schemes*, pages 328–374. Springer, 1989. 35

[161] E Sifakis and G Tziritas. Moving object localisation using a multi-label fast marching algorithm. *Signal Processing: Image Communication*, 16(10):963–976, 2001. 77

[162] P Singh, TI Hesla, and DD Joseph. Distributed lagrange multiplier method for particulate flows with collisions. *International Journal of Multiphase Flow*, 29(3):495–509, 2003. 55

[163] A Smolianski. Finite-element/level-set/operator-splitting (felsos) approach for computing two-fluid unsteady flows with free moving interfaces. *International journal for numerical methods in fluids*, 48(3):231–269, 2005. 32

[164] S Sukumaran and U Seifert. Influence of shear flow on vesicles near a wall: a numerical study. *Physical Review E*, 64(1):011916, 2001. 135, 136

[165] M Sussman, Ann S Almgren, J B B, P Colella, L H Howell, and M L Welcome. An adaptive level set approach for incompressible two-phase flows. *Journal of Computational Physics*, 148(1):81–124, 1999. 32

[166] M Sussman and Elbridge G Puckett. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *Journal of computational physics*, 162(2):301–337, 2000. 33

[167] M Sussman, P Smereka, and S Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1):146–159, 1994. 32

[168] P Swarztrauber and R Sweet. *Efficient FORTRAN subprograms for the solution of elliptic partial differential equations*. National Center for Atmospheric Research Boulder, Colorado, 1975. 38

[169] R Temam. *Navier-stokes equations*, volume 2. North-Holland Amsterdam, 1984. 63

[170] A Ten Cate, CH Nieuwstad, JJ Derksen, and HEA Van den Akker. Particle imaging velocimetry experiments and lattice-boltzmann simulations on a single sphere settling under gravity. *Physics of Fluids (1994-present)*, 14(11):4012–4025, 2002. 54

[171] G Tryggvason. Numerical simulations of the rayleigh-taylor instability. *Journal of Computational Physics*, 75(2):253–282, 1988. 22

[172] J N Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995. 30

[173] H Udaykumar, Wei Shyy, and M Rao. Elafint-a mixed eulerian-lagrangian method for fluid flows with complex and moving boundaries. In *6th Joint Thermophysics and Heat Transfer Conference*, page 1996, 1996. 18

[174] HS Udaykumar, R Mittal, P Rampunggoon, and A Khanna. A sharp interface cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics*, 174(1):345–380, 2001. 18

[175] M Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476, 2005. 13

[176] M Uhlmann. Interface-resolved direct numerical simulation of vertical particulate channel flow in the turbulent regime. *Physics of Fluids (1994-present)*, 20(5):053305, 2008. 13

[177] S O Unverdi and G Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of computational physics*, 100(1):25–37, 1992. 28

[178] S K Veerapaneni, D Gueyffier, D Zorin, and G Biros. A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2d. *Journal of Computational Physics*, 228(7):2334–2353, 2009. 7, 11, 136

[179] S K Veerapaneni, A Rahimian, G Biros, and D Zorin. A fast algorithm for simulating vesicle flows in three dimensions. *Journal of Computational Physics*, 230(14):5610–5634, 2011. 7, 136, 137

[180] N Verdon, Patrice Laure, A Lefebvre-Lepot, and L Lobry. Problèmes de contact pour des particules en écoulement cisaillé. In *10e colloque national en calcul des structures*, pages Clé–USB, 2011. 54

[181] P Vigneaux. *Méthodes Level Set pour des problèmes d'interface en microfluidique*. PhD thesis, Université Sciences et Technologies-Bordeaux I, 2007. 32

[182] R Vignjevic, T De Vuyst, and J Campbell. The use of a homogeneous repulsive force for contact treatment in sph. *WCCM V, Vienna, Austria*, 2002. 10

[183] V Vitkova, M Mader, and T Podgorski. Deformation of vesicles flowing through capillaries. *EPL (Europhysics Letters)*, 68(3):398, 2004. 136

[184] M Y Wang and X Wang. "color" level sets: a multi-phase method for structural topology optimization with multiple materials. *Computer Methods in Applied Mechanics and Engineering*, 193(6):469–496, 2004. 41

[185] T Watanabe, S Oka, and M Yamamoto. A phenomenological theory of the sigma effect. *Proceedings of the Faculty of Science of Tokai University*, 3(1):25–32, 1968. 134

[186] E Weinan and J-G Liu. Vorticity boundary condition and related issues for finite difference schemes. *Journal of computational physics*, 124(2):368–382, 1996. 39

[187] T Wick. Fluid-structure interactions using different mesh motion techniques. *Computers & Structures*, 89(13):1456–1467, 2011. 11

[188] H J Wilson. Stokes flow past three spheres. *Journal of Computational Physics*, 245:302–316, 2013. 54

[189] Jianming Yang and Frederick Stern. A highly scalable massively parallel fast marching method for the eikonal equation. *Journal of Computational Physics*, 332:333–362, 2017. 159

[190] T Ye, R Mittal, HS Udaykumar, and Wei Shyy. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of computational physics*, 156(2):209–240, 1999. 18

[191] K Yeo and M R Maxey. Simulation of concentrated suspensions using the force-coupling method. *Journal of computational physics*, 229(6):2401–2421, 2010. 55

[192] Z Yu, A Wachs, and Y Peysson. Numerical simulation of particle sedimentation in shear-thinning fluids with a fictitious domain method. *Journal of non-newtonian fluid mechanics*, 136(2):126–139, 2006. 54

[193] S Yuu and Y Fukui. Measurement of fluid resistance correction factor for a sphere moving through a viscous fluid toward a plane surface. *AIChE Journal*, 27(1):168–170, 1981. 6, 52

[194] J Zhang, P C Johnson, and A S Popel. An immersed boundary lattice boltzmann approach to simulate deformable liquid capsules and its application to microscopic blood flows. *Physical biology*, 4(4):285, 2007. 138

[195] H Zhao, A HG Isfahani, L N Olson, and J B Freund. A spectral boundary integral method for flowing blood cells. *Journal of Computational Physics*, 229(10):3726–3744, 2010. 137

[196] H Zhao and E SG Shaqfeh. The dynamics of a vesicle in simple shear flow. *Journal of Fluid Mechanics*, 674:578–604, 2011. 137

[197] H Zhou and C Pozrikidis. Deformation of liquid capsules with incompressible interfaces in simple shear flow. *Journal of Fluid Mechanics*, 283:175–200, 1995. 136