



HAL
open science

An efficient interface capturing method for a large collection of interacting bodies immersed in a fluid

Meriem Jedouaa, Charles-Henri Bruneau, Emmanuel Maitre

► To cite this version:

Meriem Jedouaa, Charles-Henri Bruneau, Emmanuel Maitre. An efficient interface capturing method for a large collection of interacting bodies immersed in a fluid. *Journal of Computational Physics*, 2019, 378, pp.143-177. 10.1016/j.jcp.2018.11.006 . hal-01236468v2

HAL Id: hal-01236468

<https://hal.science/hal-01236468v2>

Submitted on 13 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An efficient interface capturing method for a large collection of interacting bodies immersed in a fluid.

M. Jedouaa^{a,b,*}, C-H. Bruneau^b, E. Maitre^a

^aLaboratoire Jean Kuntzmann, Univ. Grenoble Alpes and CNRS, Grenoble, France

^bUniv. Bordeaux IMB, INRIA Bordeaux-Sud-Ouest Team MEMPHIS, CNRS UMR 5251

5

Abstract

An efficient method to capture an arbitrary number of fluid/structure interfaces in a level-set framework is built, following ideas introduced for contour capturing in image analysis. Using only three label maps and two distance functions we succeed in locating and evolving the bodies independently in the whole domain and get the distance between the closest bodies in order to apply a collision force whatever the number of cells is. The method is applied to rigid solid bodies in order to compare to the results available in the literature. In that case, a global penalization model uses the label maps to follow the solid bodies all together without a separate computation of each body velocity. Numerical simulations are performed in two- and three-dimensions. An application to immersed vesicles is also proposed and shows the capability and efficiency of the method to handle numerical contacts between elastic bodies at low resolution. Two-dimensional simulations of vesicles under various flow conditions are presented.

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

1. Introduction

Numerical simulations of fluid/structure and structure/structure interaction (FSI) have attracted an increasing interest and several methods have been proposed during the last decades.

A popular and wide spread method is the Arbitrary Lagrangian Eulerian approach (ALE) introduced by Donea in 1982 (see [1]) and extensively studied by several teams [2, 3, 4, 5, 6]. 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 [7] has been a long standing difficulty which has been worked around only recently [8, 9, 10, 11, 12]. Moreover, the computational grid has to be remeshed when the elements get too distorted, which could be a very costly procedure in dimension three.

Another method, introduced by Cottet and Maitre in [13, 14], consists in using a purely Eulerian formulation for describing the fluid/structure interaction. This approach was inspired by the immersed boundary method of Peskin [15] where the forces at the interface were described in a Lagrangian manner. In the model [13], a level set method is used to capture the interface. The level set method was developed in [16] to treat problems involving interfaces. It is used in many domains because of its numerous advantages: its ease of implementation, topological changes are directly handled and one single level set function can capture an

*Corresponding author

Email addresses: mjedouaa@yahoo.fr (M. Jedouaa), charles-henri.bruneau@math.u-bordeaux.fr (C-H. Bruneau), emmanuel.maitre@univ-grenoble-alpes.fr (E. Maitre)

arbitrary number of interfaces. This last property is largely used as it can be a very efficient tool to capture several interfaces, for instance when merging and splitting of interfaces are allowed. The present work aims at dealing with a dense suspension of biological cells immersed in a fluid. In this kind of application, using one level set is not always sufficient as will be explained thereafter. The other existing level set models for capturing a large number of objects are either computationally expensive or cannot be used to handle collisions. Indeed, in [17] one level set function is required for each body, leading to a huge computational cost although contacts are easily avoided. In [18], a formulation using $\log_2 N$ level set functions to represent N different regions is designed. This model, based on the four color theorem, substantially reduces the number of level set functions and can handle very easily complex topologies. However, reconstruction of all distance between bodies can not be achieved, as a consequence this model is not able to deal with several pairwise interacting bodies immersed in a fluid.

Indeed, in that case, the investigation of fluid/structures interaction raises the problem of collisions between the bodies. These collisions can be handled if the distance between bodies can be computed from the localisation procedure. As confirmed by theoretical works [19], the hydrodynamical forces between two smooth bodies evolving in a Navier-Stokes flow do prevent contact in finite time. However, during numerical calculations, the flow between too close 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 [20] 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 overcoming this low resolution problem near contact by taking into account lubrication forces, when the gap between the bodies is very small [21, 22]. Due to the singular behavior of the forces, and the time discretization errors, this approach appears to be insufficient and might still lead to contacts and overlaps at low spatial resolution.

Other numerical strategies, less respectful of the underlying physics, consist in imposing a constraint on the particle motion by means of artificial short range repulsive forces [23, 17] or by directly enforce a minimal distance between the particles [21]. By contrast to the refinement strategy, these collision methods allow, in addition of handling overlaps and contacts between particles, to use a coarser discretization, reducing substantially the computational cost compared to the method proposed in [20].

In our work we consider such an algorithm, with short range repulsive forces, built on the multi geometric deformable model (MGDM) of interface capturing introduced by J. Bogovic [24] for image segmentation. The proposed algorithm can handle multiple deforming bodies and avoid collision using a short range repulsive force depending on the distance to the closest interface, following [17]. The main advantage of this method is that it requires, whatever the number of interacting objects, only five fields to

- (1) locate and evolve each structure in the domain,
- (2) specify a speed or a force independently for each structure,
- (3) handle numerical contacts between the structures.

This substantially reduces the computational cost, as will be illustrated below.

More precisely, the level set defined in a neighborhood of the interfaces is transported with the fluid velocity. A multi-label fast marching method is then performed in a narrow-band around the interfaces allowing to update the label and distance functions. This combines the advantage of the MGDM method which efficiently captures a large number of bodies and their relative neighbours and of the collision model introduced in [17] using a level set decomposition.

In order to validate the ability of this method to avoid numerical contacts and its efficiency to deal with a large number of structures, two applications are explored: rigid bodies and vesicles suspensions.

This paper is organized as follows: Section 2 presents the proposed model with a careful description of the three label maps and the two distance functions. In section 3, the repulsive forces used to avoid collisions are described. Sections 4 and 5 shows how to apply the method to the case of rigid bodies immersed in

an incompressible fluid. In section 6 is proposed a benchmark of numerical simulations starting with a qualitative study of the grid convergence on the sedimentation of rigid disks in two dimensions. Then, other simulations in two- and three-dimensions are performed and compared to the results of the literature. Section 7 is dedicated to the numerical simulations of immersed vesicles. Two-dimensional numerical results of vesicles under different flow conditions are presented. Finally, some conclusions are derived.

2. Domain labeling

In this section, we first recall some basic principles of the level set method. Then, we provide a description of the method used to capture multiple interfaces. This method, inspired by the multi geometric deformable model of J. Bogovic [24], is introduced in the context of several bodies immersed in a fluid. The main idea is to partition the entire fluid/structures domain into several objects. In order to locate the different objects in the domain we introduce a set of label maps and distance functions. Let $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) be a bounded domain which contains N bodies immersed in an incompressible fluid, we denote Ω_i and Γ_i the interior and the boundary of the i^{th} body. By the term body we refer in this article to a simply connected domain of the ambient space. We consider the fluid Ω_{N_f} as an object. Thus the fluid/structures domain Ω is partitioned into $N + 1$ objects as:

$$\begin{cases} \forall i \neq j, \Omega_i \cap \Omega_j &= \emptyset \\ \Omega_{N_f} &= \overline{\Omega \setminus \{\cup_{i=1}^N \Omega_i\}} \\ \Gamma_{N_f} &= \cup_{i=1}^N \Gamma_i. \end{cases} \quad (1)$$

2.1. Outline of the level set method

Pioneered by Osher and Sethian in [16], the level set method is very popular to address problems involving interfaces. It is widely used for numerical analysis of surfaces and shapes and in the context of fluid/structure interaction. The general idea of the level set method is to define a scalar function in the all computational domain that vanishes on the location of the interface to capture.

Let Ω be a bounded domain in \mathbb{R}^d ($d = 2$ or $d = 3$) partitioned into two sub domains Ω_1 and Ω_2 and Γ be the interface between Ω_1 and Ω_2 . The aim is to follow the evolution of the interface Γ that is defined as the zero level set of a smooth function ϕ , with a gradient modulus bounded away from zero in a neighborhood of this level set. At each time t , the interface Γ is characterized by:

$$\Gamma(t) = \{x \in \Omega, \phi(x, t) = 0\}$$

It is for example defined as:

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

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

$$n = \frac{\nabla \phi}{|\nabla \phi|} \quad \kappa = \nabla \cdot n.$$

The displacement of the interface is obtained by the evolution of the level set function ϕ . Let u be the velocity in the domain Ω , the level set function is the solution of the scalar transport equation:

$$\partial_t \phi + u \cdot \nabla \phi = 0. \quad (2)$$

The velocity field u may depend on the space, the time, the geometric properties of the curve and/or the physics of the problem. For instance, if we consider a problem of fluid/structures interaction this velocity field is the velocity of the fluid or of the structures (assuming continuity of velocity at interfaces). To avoid

situation where gradient modulus could become close to zero, we usually define the level set function as a signed distance function that is regular in each corresponding domain:

$$\phi(x) = \begin{cases} -d(x, \Gamma) & x \in \Omega_1 \\ d(x, \Gamma) & x \in \Omega_2 \end{cases} \quad (3)$$

where

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

105 We then perform redistancing during the algorithm moving the interfaces.

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 $\phi(x)$. In practice, a regularized version of these functions H_ε and ζ_ε is used on the interface in order to reduce grid effects:

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

and $\zeta_\varepsilon(\phi(x)) = H'_\varepsilon(\phi(x))$, where ε represents half of the interface thickness. Note that a classically mentioned advantage of the level set method is to handle automatically changes of topology. This property is problematic in the case of a collections of objects which are not supposed to merge (such as red blood cells), as splitting or merging of interfaces are directly taken into account by the level set function. Our algorithm
110 will be built to prevent this unwanted merging.

2.2. Level set functions for multiple interfaces

Let us consider N pairwise disjoint bodies Ω_i , $i \in \{1, \dots, N\}$ with $\Omega_i \subset \Omega$, and let us denote Γ_i the frontier of the cell Ω_i and Ω_{N_f} the fluid background subdomain with $N_f = N + 1$. A major advantage of the level set method is that one level set function can capture an arbitrary number of interfaces between the bodies and the fluid. Let ϕ be the level set function which captures the union of the N bodies Ω_i , $i \in \{1, \dots, N\}$, 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 is very time efficient for computations, since it is independent on the number of cells. However, it is not possible to specify a different velocity model and/or a force for each cell. If two bodies 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 bodies. So this method can lead to collision and merging of bodies. Another way to capture multiple bodies 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 body Ω_i by using the level set function ϕ_i and we get the distance between all the bodies. Indeed, the distance between two bodies 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 major disadvantage is its computational cost when the number of bodies increases.

2.3. Label maps

115 At every point x of the fluid/structures domain Ω , we define the label functions L_0, L_1, L_2 as:

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

The label map L_0 provides a partition labelling of the whole computational domain Ω into $N + 1$ different objects. The label map L_1 identifies the index in this labelling of the first closest object at all points in Ω . The label map L_2 identifies the index of the second closest object at all points in Ω .

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

In particular, for each point of a solid the label map L_1 corresponds to the index of the fluid. The label map function L_2 gives the index of the first closest body for any x in the structure domain, and the second closest body for x in the fluid domain.

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

In cases where two structures are at equal distance to a point, the label is one of these structures. We will see thereafter that this is not a problem in our algorithm, since repulsion forces acting on two equally distant objects will move them so that is equality case does not hold anymore after one time step. Figure 1 shows an illustration of the three label maps in the case of five structures immersed in a fluid.

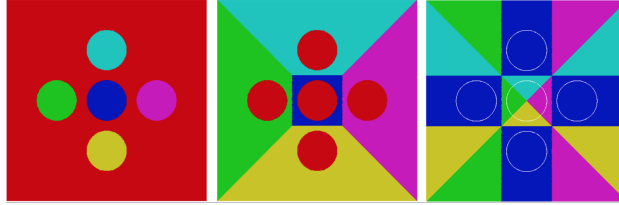


Figure 1: Illustration of the three label maps for a configuration of five bodies, from left to right: L_0 , L_1 and L_2 . Each object has a specific color and the red one corresponds to the fluid. The white contour represents the boundary of the bodies.

Taking advantage of this local configuration of the closest object, one can define two related distance functions.

2.4. Distance functions

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

$$\forall x \in \Omega, \begin{cases} \varphi_1(x) = d(x, \Gamma_{L_1(x)}) \\ \varphi_2(x) = d(x, \Gamma_{L_2(x)}) \end{cases} \quad (4)$$

The distance function $\varphi_1(x)$ 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 , φ_1 captures the union of all bodies interfaces and φ_2 provides the distance to the first closest body. As a consequence, on each point of a structure, we have the distance to the next closest one allowing to define a collision model to the closest interface, since avoiding contacts is equivalent to require:

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

Figure 2 shows an illustration of the two distance functions related to the label maps of the Figure 1.

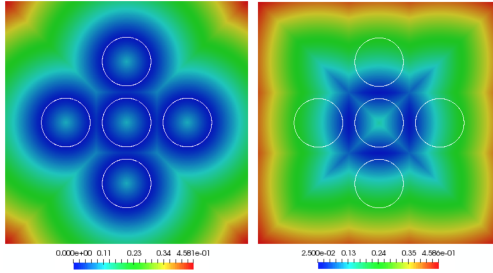


Figure 2: Illustration of the two distance functions for a configuration of five structures, from left to right: φ_1 and φ_2 . The white contour represents the boundary of the circular bodies.

3. Collisions strategy

As discussed before, it is crucial to develop a collision model to avoid contacts between bodies. In this section, we present the proposed collision model that consists in a short range repulsive force taking into account the interactions between the closest bodies. This short range repulsive force is inspired by the collision model introduced in [17].

3.1. Collision model using several level set functions

The collision model [17] 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. We consider N bodies immersed in a fluid and we denote by $F_{j,i}$ the force applied by the body Ω_j on 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 provided by $\phi_j(x)$ and the direction of the force $F_{j,i}$ is obtained directly by $\nabla\phi_j$.

Moreover, to locate the interface Γ_i we use a cut off function regularized on a thickness ε on each part of the interface. The short range repulsive force is formulated as:

$$\forall x \in \Omega, F_{j,i}(x) = \frac{k}{\varepsilon} \rho(x) \zeta\left(\frac{\phi_i(x)}{\varepsilon}\right) \frac{\nabla\phi_j(x)}{\phi_j(x)} \exp\left(-\frac{\phi_j(x)}{\varepsilon_b}\right) \quad (5)$$

Consequently, we obtain the following collision model:

$$\forall x \in \Omega, F_{\text{global}}(x) = \sum_{\substack{i,j=1 \\ i \neq j}}^N \rho(x) \frac{k}{\varepsilon} \zeta\left(\frac{\phi_i(x)}{\varepsilon}\right) \frac{\nabla\phi_j(x)}{\phi_j(x)} \exp\left(-\frac{\phi_j(x)}{\varepsilon_b}\right) \quad (6)$$

where k is a repulsive constant proportional to the square of the relative velocities of the corresponding bodies just before collision and ρ denotes the density. The coefficient ε represents the half thickness of the interface on which the repulsive forces are applied and ε_b corresponds to the rebound coefficient. In [17], this value is fixed to ε . The interaction forces decrease exponentially fast for far away structures, reducing the number of influent interacting neighbours. Nevertheless, this collision model accounts for all possible interactions between the N bodies. Consequently for N bodies captured by N level set functions, N^2 computations of the repulsive forces are required, which represents a huge computational effort for large N .

3.2. The proposed collision model using two distance functions

To reduce the high computational cost of (6), we adapt F_{global} so that it depends only on the two distance functions φ_1 and φ_2 . We set:

$$\forall x \in \Omega, F_{\text{label}}(x) = \frac{k}{\varepsilon} \rho(x) \zeta\left(\frac{\varphi_1(x)}{\varepsilon}\right) \frac{\nabla\varphi_2(x)}{\varphi_2(x)} \exp\left(-\frac{\varphi_2(x)}{\varepsilon_b}\right) \quad (7)$$

The term $\zeta\left(\frac{\varphi_1(x)}{\varepsilon}\right)$ provides the location of the union of all interfaces. Thus, this force has its support
 160 on a subset $\Gamma_\varepsilon = \{x \in \Omega, \varphi_1(x) \leq \varepsilon\}$. In numerical computations, the intensity of the force is controlled
 by the repulsive parameter $k = K(L_0, L_2)$, this parameter is proportional to the relative velocities between
 closest bodies just before collision. This coefficient is computed as:

$$K(L_0, L_2) = \max(K_{min}, |(U_{L_2} - U_{L_0}) \cdot \nabla \varphi_2|, ((U_{L_2} - U_{L_0}) \cdot \nabla \varphi_2)^2) \quad (8)$$

where U_{L_0} denotes the velocity of the body on which forces are applied, whereas U_{L_2} is the velocity of the
 first closest body. The term K_{min} corresponds to the minimal intensity of the force which is used when
 165 some particles are almost at rest and hence, relative velocities are negligible.

This collision model takes into account the interaction between the closest bodies 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
 170 of its interface.

For instance, in Figure 1, 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 body and so forth. On the four surrounding bodies, only the forces
 exerted by the blue body is applied.

175 The advantage of this formulation is that we get rid of the sum in (6) leading to a considerable saving of
 computational cost. In this section, we consider the case $\rho = 1$ to compare the two collision forces.

3.3. Comparison of the two collision models

In this subsection, we compare the two collision models introduced above in two-dimensions and for
 $\rho = 1$. We consider the case of N circular rigid bodies having the same radius R and we set $\rho = 1$ to
 180 compare the two forces. The following result provides an estimate of the difference introduced by using
 the label force (7) instead of the collision model (6). As expected, the difference is smaller for large bodies
 and/or small ε .

Proposition 1. *Assuming that the N bodies are disks, such as:*

$$\forall x \in \Omega, \quad \varphi_2(x) \geq \varepsilon$$

then:

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

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

Proof. Let $L_i(x)$ denote the i^{th} closest object to x and $\phi_{L_i(x)}$ the distance function associated to the object
 $\Omega_{L_i(x)}$, it comes:

$$\forall x \in \Omega, \forall i \in \{1, \dots, N\}, \quad \phi_{L_i(x)}(x) = d(x, \Gamma_{L_i(x)}).$$

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

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

Then, the model (6) can be written using these functions:

$$\forall x \in \Omega, \quad F_{global}(x) = \sum_{i=2}^N \frac{k}{\varepsilon} \zeta_\varepsilon(\varphi_1(x)) \frac{\nabla \phi_{L_i(x)}(x)}{\phi_{L_i(x)}(x)} \exp\left(-\frac{\phi_{L_i(x)}(x)}{\varepsilon}\right).$$

As defined in subsection 2.4, we have $\varphi_2 = \phi_{L_2}$ and using (7) leads to:

$$\forall x \in \Omega, \quad F_{\text{global}}(x) - F_{\text{label}}(x) = \sum_{i=3}^N \frac{k}{\varepsilon} \zeta_\varepsilon(\varphi_1(x)) \frac{\nabla \phi_{L_i(x)}(x)}{\phi_{L_i(x)}(x)} \exp\left(-\frac{\phi_{L_i(x)}(x)}{\varepsilon}\right). \quad (9)$$

185 Note that these short-range repulsive forces tends to zero out of a cut-off radius, thus only the interactions with the $N_n = \mathcal{O}(1)$ closest bodies of a given body will be numerically significant. However if we consider all interactions we get:

$$\forall x \in \Omega, \quad F_{\text{global}}(x) - F_{\text{label}}(x) \simeq \sum_{i=3}^N \frac{k}{\varepsilon} \zeta_\varepsilon(\varphi_1(x)) \frac{\nabla \phi_{L_i(x)}(x)}{\phi_{L_i(x)}(x)} \exp\left(-\frac{\phi_{L_i(x)}(x)}{\varepsilon}\right). \quad (10)$$

Considering that $\forall i \in \{2, \dots, N_n\}$, ϕ_{L_i} are distance functions we get:

$$\|F_{\text{global}} - F_{\text{label}}\|_{L^1} \leq \left\| \frac{k}{\varepsilon} \zeta_\varepsilon(\varphi_1) \right\|_{L^1(\Omega)} \left\| \sum_{i=3}^{N_n} \frac{1}{\phi_{L_i}} \exp\left(-\frac{\phi_{L_i}}{\varepsilon}\right) \right\|_{L^\infty(\Omega)}.$$

Moreover by definition we have:

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

leading to :

$$\|F_{\text{global}} - F_{\text{label}}\|_{L^1} \leq (N-2) \left\| \frac{k}{\varepsilon} \zeta_\varepsilon(\varphi_1) \right\|_{L^1(\Omega)} \left\| \frac{1}{\phi_{L_3}} \exp\left(-\frac{\phi_{L_3}}{\varepsilon}\right) \right\|_{L^\infty(\Omega)}.$$

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

$$\left\| \frac{1}{\varepsilon} \zeta_\varepsilon(\varphi_1) \right\|_{L^1(\Omega)} = N(2\pi R + \mathcal{O}(\varepsilon)),$$

Moreover, therefore,

$$\|F_{\text{global}} - F_{\text{label}}\|_{L^1(\Omega)} \leq N(2\pi Rk + \mathcal{O}(\varepsilon)) \left\| \frac{1}{\phi_{L_3}} \exp\left(-\frac{\phi_{L_3}}{\varepsilon}\right) \right\|_{L^\infty} \quad (11)$$

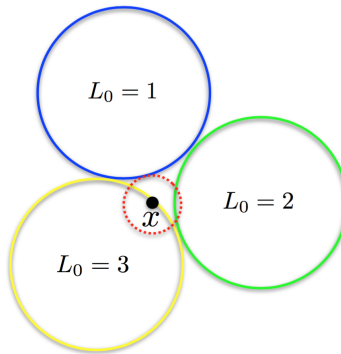


Figure 3: Configuration of three bodies. The circles represented are of radius $R + \varepsilon$.

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. 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, 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. 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 $r \rightarrow \frac{1}{r} \exp(-\frac{r}{\varepsilon})$ is decreasing, we obtained the announced estimation. \square

190

Remark 1. 1. 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, the coefficient ε depends on the spatial discretization step and is taken equal or larger than $\frac{R}{10}$. Moreover, the influence of the repulsive force imposed by the first closest cell at all points F_{label} , is the most influential on the dynamics of the cells as it is the largest.

195

2. Moreover, as previously stated, due to the exponential decay the repulsion forces magnitudes, only the closest bodies will have significant contribution to the global repulsion at some point of a given body. Therefore conceptually the leading coefficient N in (11) could be replaced by N_n , where N_n is the maximal number of bodies that could be in contact with a given body. This is dependent of the dimension and could depend also on the body if those have different sizes.

200

For a more in-depth study of the collision model proposed, we refer to Meriem Jedouaa PhD Thesis ([25], in English) where the full fluid-structure models are compared in the case of Stokes flows.

4. Evolution of the label maps and distance functions in the general case

This section 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.

205

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. As the proposed collision model is a short-range repulsive force, we only evolve L_1 and φ_1 in a thin band close to the interfaces, we denote by d_{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 meet (just before collision) and is achieved in a thin band of size d_{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 close to the interfaces.

210

This procedure is summarized as:

215

1. Transport a level set function ϕ which captures the union of all interfaces with the fluid velocity U ,
2. Evolve the label maps L_0 , by redefining its values near the interfaces,
3. Perform a multi label fast marching to update L_1 , φ_1 in a thin band around the interfaces,
4. If bodies are nearby, perform a multi label fast marching to update L_2 , φ_2 .

4.1. Global level set field ϕ

220

We define the level set function ϕ which captures the union of all interfaces as:

$$\phi(x) = \begin{cases} -C & x \in \cup_{i=1}^N \Omega_i, d(x, \cup_{i=1}^N \Gamma_i) > d_{L_1} \\ -d(x, \cup_{i=1}^N \Gamma_i) & x \in \cup_{i=1}^N \Omega_i, d(x, \cup_{i=1}^N \Gamma_i) \leq d_{L_1} \\ d(x, \cup_{i=1}^N \Gamma_i) & x \notin \cup_{i=1}^N \Omega_i, d(x, \cup_{i=1}^N \Gamma_i) \leq d_{L_1} \\ C & x \notin \cup_{i=1}^N \Omega_i, d(x, \cup_{i=1}^N \Gamma_i) > d_{L_1} \end{cases} \quad (12)$$

where $C \gg 1$. This level set field is a signed distance function close to the interfaces, it is negative inside each structure and positive outside. Denoting by U the fluid velocity in the entire computational domain Ω , the level set function is solution of the following scalar transport equation:

$$\begin{cases} \partial_t \phi + U \cdot \nabla \phi = 0 \\ \phi(x, 0) = \phi_0(x) \end{cases} \quad (13)$$

225 in a neighborhood of the interfaces.

4.2. Redefinition of the label 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:

$$\forall x \in \Omega, \forall i \in \{1, \dots, N\}, L_0^n(x) = \begin{cases} i \neq N_f & \text{if } \phi^n(x) < 0 \\ N_f & \text{if } \phi^n(x) \geq 0 \end{cases} \quad (14)$$

At each time step, we change the label value L_0 , near the interface at the points where the condition (14) is not verified. Namely, if the level set function is non negative, we set the new value of L_0 to the label of the fluid ($L_0 = N_f$) and if the level set function is negative and the label function L_0 was the label of the fluid, we assign the new value of L_0 to the value of the neighbour of x whose label is different from N_f . We consider here that the level set function is transported with an explicit scheme under a CFL condition. For all point $x = (i, j)$ we denote by $\text{Neighbours}(x)$ the set which contains the four closest points to x :

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

The whole process is sketched as:

230

Algorithm 1 Redefinition of L_0

```

for ( $x \in \Omega$ ) do
  if  $\phi^{n+1}(x) \geq 0$  then
     $L_0^{n+1}(x) = N_f$ 
  else
    if ( $L_0^n(x) = N_f$ ) then
      for ( $y \in \text{Neighbours}(x)$ ) do
        if ( $L_0^n(y) \neq N_f$ ) then
           $L_0^{n+1}(x) = L_0^n(y)$ 
        end if
      end for
    end if
  end if
end for

```

This simple strategy is allowed only if the distance between two bodies is strictly greater than two spatial discretization steps. It will have to be taken into account by our collision model.

4.3. Multiple label fast marching method

We describe here the local fast marching algorithm that allows to update φ_1 , φ_2 and the label maps L_1 and L_2 in a vicinity of the interfaces. This local fast marching is an extension of the fast marching method [26] that was introduced in [27] and [24]. To this end, we solve two eikonal equations of the form:

$$\forall x \in \Omega, |\nabla d(x)| = 1$$

This equation is solved using a first order numerical scheme [28] and has to be supplemented by a boundary condition which depends on φ_1 or φ_2 .

235

4.3.1. Computation of L_1 and φ_1

The first step of the proposed procedure is the computation of the distance function φ_1 and the values L_1 by solving the following equation:

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

This process is executed in a narrow-band of size d_{L_1} .

At initialization, we set $\varphi_1(x)$ to $\frac{\varphi_1(x)}{|\nabla\phi(x)|}$ close to the interfaces, this allows to propagate all the interfaces simultaneously.

240 There are three sets of points: alive(A), narrow-band (NB) and far away (F). At initialisation, 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 have to be computed and added to the narrow band.

245 Moreover, each contour is associated to the number of objects. To this end, we introduce a function lab which corresponds to the number of the interface that spreads.

During the propagation of the interfaces, the point x_m for which $\varphi_1(x_m)$ is the lowest value of φ_1 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 φ_1 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 φ_1 has reached the narrow-band value d_{L_1} . The steps performed during the initialization phase are provided by the algorithm2.

Algorithm 2 Initialization phase

```

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

```

Algorithm 3 Iterative phase: computation of φ_1 and L_1

```
dist=0
while ((NB  $\neq$   $\emptyset$ ) and (dist  $\leq$  NB $_{L_1}$ )) do
  Find  $x_m$  such as  $d(x_m) = \min_{(x \in NB)} d(x)$ 
  dist = max(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  $L_1(\text{Neighbors}(y))=\text{lab}(x_m)$ 
        Add  $y$  in NB
      end if
    end for
  end if
end while
```

4.3.2. Computation of the label map L_2 and φ_2

As the procedure to update L_1 and φ_1 is only performed in a narrow-band of size d_{L_1} , the computation
255 of L_2 and φ_2 will be activated between two particles only if two or more particles are at a distance less or
equal to $2d_{L_1}$.

If that is the case, we solve the following equation:

$$\begin{cases} \forall x \in \Omega, & |\nabla d(x)| = 1 \\ \forall x \in \Gamma_{L_1}, & d(x) = \varphi_2(x) \end{cases} \quad (15)$$

in a narrow-band of size d_{L_2} . Γ_{L_1} corresponds to the intersection of the closest objects in the fluid domain, each point $x \in \Gamma_{L_1}$ is defined by:

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

Only the points close to the interface Γ_{L_1} are visited as the algorithm stops propagation when the
260 distance function φ_2 reaches the distance value d_{L_2} .

Figure 4 provides an illustration of the results obtained for the label map L_1 and L_2 for a configuration of
20 rectangular particles and $d_{L_1} = 10h$ and $d_{L_2} = 5h$. We observe that the computation of L_2 is achieved
only for the closest structures allowing to reduce the CPU time. Moreover, the short-range repulsive force
is not exerted on some particles as the distance is greater than $20h$ thus if $\varepsilon = h$ the exponential term of
265 the force is lower than 10^{-9} .

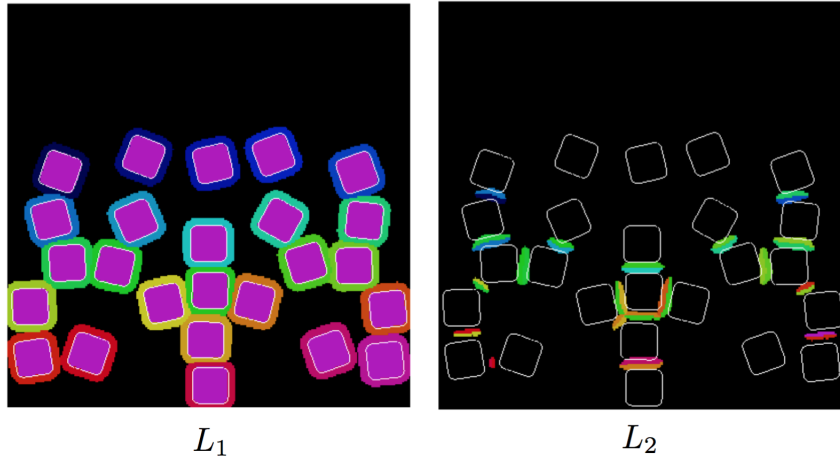


Figure 4: Illustration of the multi label fast marching algorithm for 20 objects. The black color corresponds to the undefined values of L_1 and L_2 . The white contours corresponds to the zero value of the level set function ϕ

4.3.3. Computational Complexity

We provide here the computational complexity of the multi-label fast marching (MLFMM) algorithm presented above. To this end, we add particles in the computational domain Ω next to each other so that the particles are close enough leading to the computation of L_2 as soon as there are two bodies in the domain. The CPU time is represented in Figure 5 for different number of particles close to each other for $d_{L_1} = 10h$ and $d_{L_2} = 5h$ where h denotes the spatial discretization step.

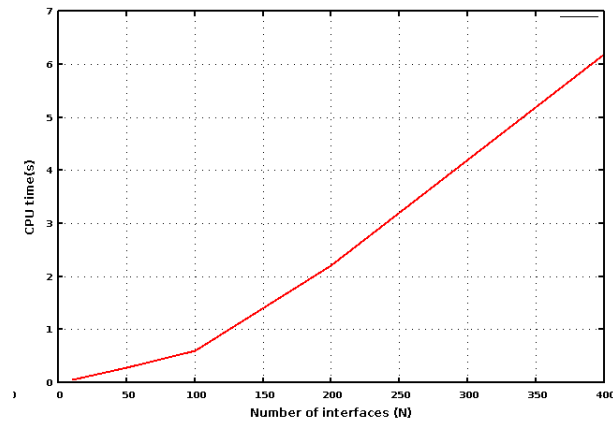


Figure 5: CPU time of the proposed multi label fast marching algorithm.

5. Application to rigid bodies

In this section, we propose an application of the model to rigid bodies immersed in an incompressible fluid.

First, 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 numerical illustrations.

5.1. Volume penalization models

280 We consider N bodies evolving in an incompressible fluid denoted Ω_{N_f} and the entire domain Ω is partitioned in the same way as (1). In this case each body $\Omega_i, 1 \leq i \leq N$ represents a rigid body. A volume penalization method is adopted to take into account the immersed rigid bodies.

5.1.1. Flow configuration

We denote by ρ_f and μ the constant density and the viscosity of the fluid and by U and p the flow velocity and the pressure. The flow is governed by the viscous incompressible Navier-Stokes equations:

$$\begin{cases} \rho_f(\partial_t U + (U \cdot \nabla)U) - \nabla \cdot (\mu \nabla U) + \nabla p = 0 & \text{in } \Omega_{N_f} \times (0, T) \\ \nabla \cdot U = 0 & \text{in } \Omega_{N_f} \times (0, T) \end{cases}$$

with the condition at the interfaces of the solid bodies:

$$\forall i \in \{1, \dots, N\}, U = u_i \text{ on } \Gamma_i \quad (16)$$

285 where u_i is the rigid motion of the body Ω_i . In order to get rid of the boundary conditions (16) on the solid boundaries, we use a penalization method ([29, 30, 17]). This method consists in adding a penalization term in the Navier-Stokes equations to impose the rigid 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. If we denote by χ_i the characteristic function of the body Ω_i and by λ the penalization parameter we obtain the following model:

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U) - \nabla \cdot (\mu \nabla U) + \nabla p \\ \quad = \rho g + \lambda(\sum_{i=1}^N \chi_i(u_i - U)) + F_{\text{global}} + F_{\text{wall}} & \text{in } \Omega_T = \Omega \times (0, T) \\ \nabla \cdot U = 0 & \text{in } \Omega_T \end{cases} \quad (17)$$

complemented with homogeneous Dirichlet boundary and initial conditions, where g is the gravity force, F_{global} is the collision force (7) and F_{wall} is a repulsive force carry on by the solid boundaries of the domain Ω on the rigid bodies. The rigid velocity u_i is obtained by averaging translation and angular velocities over the solid body Ω_i ([30]):

$$u_i = u_i^t + u_i^\theta = \frac{1}{|\Omega_i|} \int_{\Omega_i} \rho \chi_i u dx + \left(J_i^{-1} \int_{\Omega_i} \rho \chi_i u \times (z - x_i^g) dz \right) \times (x - x_i^g) \quad (18)$$

295 where J_i is the inertial matrix of the body Ω_i and x_i^g its center. As the rigid bodies move with the fluid, the displacement of χ_i is:

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

Finally, denoting by ρ_i the density of the immersed bodies Ω_i , we obtain the following density function:

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

Let us note that the coefficient $\lambda \gg 1$ means that the regions occupied by the solid bodies are considered as porous media with a very small permeability. Numerically, the penalisation parameter λ is fixed to 10^{10} .

300 5.1.2. The penalization model using several level set functions

The characteristic function χ_i can also be deduced considering a level set function ϕ_i that captures the interface Γ_i . This level set function evolves as:

$$\partial_t \phi_i + u_i \cdot \nabla \phi_i = 0$$

and using the regularized Heaviside function χ_i can be smoothed as:

$$\chi_i^\varepsilon = 1 - H_\varepsilon(\phi_i).$$

So the model becomes:

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U) - \nabla \cdot (\mu \nabla U) + \nabla p \\ \quad = \rho g + \lambda(\sum_{i=1}^N \chi_i^\varepsilon(u_i - U)) + F_{\text{global}} + F_{\text{wall}} & \text{in } \Omega_T \\ \nabla \cdot U = 0 & \text{in } \Omega_T \\ \partial_t \phi_i + u_i \cdot \nabla \phi_i = 0 & \text{in } \Omega_T \end{cases} \quad (19)$$

This system is dependent on the number of structures causing a high computational cost if one wishes to simulate a large number of bodies.

5.1.3. The proposed penalization model

305 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 the penalization model.

To this end, 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:

$$\forall x \in \Omega, \quad L_{0,1}(x) = \begin{cases} L_0(x) & \text{if } (L_0(x) \neq N_f) \\ L_1(x) & \text{otherwise} \end{cases} \quad (20)$$

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

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

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

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

310 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 $U_i, i \in \{1, \dots, N\}$. More precisely, for each $x \in \Omega$, $y \rightarrow U_{L_{0,1}(x)}(y)$ is the rigid velocity of the solid body $\Omega_{L_{0,1}(x)}$ obtained by averaging the translation and angular velocities over the solid.

315

Setting:

$$\forall x \in \Omega, \quad |\Omega_{L_{0,1}(x)}| = \int_{\Omega_{L_{0,1}(x)}} \rho(z) dz = \int_{\Omega} \rho(z) \chi_{L_{0,1}(x)}(z) dz \quad (21)$$

we obtain the following formulation .

$\forall x \in \Omega, \forall y \in \Omega,$

$$\begin{aligned} U_{L_{0,1}(x)}(y) &= \frac{1}{|\Omega_{L_{0,1}(x)}|} \int_{\Omega} \rho_{L_{0,1}(x)}(z) \chi_{L_{0,1}(x)}(z) U(z) dz \\ &+ \left(J_{L_{0,1}(x)}^{-1} \int_{\Omega} \rho_{L_{0,1}(x)}(z) \chi_{L_{0,1}(x)}(z) U(z) \times (z - x_{L_{0,1}(z)}^g) dz \right) \times (y - x_{L_{0,1}(y)}^g). \end{aligned} \quad (22)$$

where $J_{L_{0,1}(x)}$ and $x_{L_{0,1}(x)}^g$ are the inertial matrix and center of gravity of solid $\Omega_{L_{0,1}(x)}$.

In a neighborhood of a particle Ω_i that is when $L_{0,1} = i$, the defined global velocity corresponds exactly

to the rigid velocity U_i .

320

Finally, the proposed penalized model is provided by:

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla)U) - \mu \Delta U + \nabla P \\ \quad = \rho g + \lambda(\chi_{L_{0,1}}(U_{L_{0,1}} - U)) + F_{\text{label}} + F_{\text{wall}} & \text{in } \Omega_T \\ \nabla \cdot U = 0 & \text{in } \Omega_T \\ \partial_t \phi + U \cdot \nabla \phi = 0 & \text{in } \Omega_T \end{cases} \quad (23)$$

This model has to be completed with the initial and boundary conditions on the velocities and pressure. On the implementation side, all rigid velocities are computed incrementally, involving only one iteration on the mesh grid. Thanks to the label maps, we have suppressed the dependence on the number of bodies in the repulsive force and in the penalization term. The only dependence on the number of bodies comes from the storage of the N level set functions. Therefore, this is a very desirable model to simulate a large number of interacting bodies. In the case of circular rigid bodies, the level set equation is replaced in the previous model (23) by the transport of the gravity center of the N bodies and a reconstruction of the associated level set function is performed:

$$\partial_t x_i^g = u_i^t, \quad 1 \leq i \leq N$$

5.2. Discretization and Numerical implementation

This part elucidates the numerical resolution of the system (23), 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.

325

5.2.1. 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 $U^n \approx U(\cdot, t^n)$. First, we compute an intermediate State U^* from:

$$U^* = U^n - \Delta t(U^n \cdot \nabla)U^n + \frac{\Delta t \mu}{\rho^n} \Delta U^n + \Delta t g - \frac{\Delta t \nabla p^n}{\rho^n}. \quad (24)$$

Then we solve the pressure from the equation:

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^n} \right) = \frac{\text{div}(U^*)}{\Delta t} + \nabla \cdot \left(\frac{\nabla p^n}{\rho^n} \right) \quad (25)$$

330 so that the velocity:

$$\bar{U} = U^* - \frac{\Delta t}{\rho_0} (\nabla p^{n+1} - \nabla p^n) \quad (26)$$

is divergence free. The pressure equation (25) can be solved directly by a conjugate gradient algorithm or approximated with a relaxation procedure as follows:

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

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 \bar{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

335

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 5th 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 6 shows that the results obtained with $\rho_0 = \rho_f$ gives the worst approximation

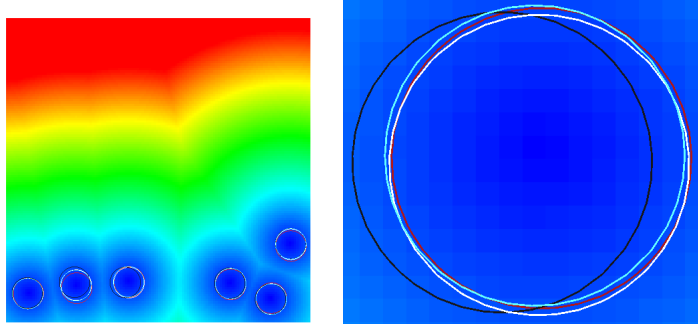


Figure 6: Flow field for six rigid disks falling under gravity (left) and a zoom of the location of the interface for different values of ρ_0 (right). The conjugate gradient method is in red, $\rho_0 = \rho_f$ in black, $\rho_0 = \rho_{L_{0,1}}$ in white and $\rho_0 = (\rho_{L_{0,1}} + \rho_f)/2$ in blue. The background color shows the level set amplitude.

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.

Penalization algorithm 1

In the general case the algorithm performs the following steps:

1. Compute and add the repulsive force $U_{\text{col}} = \bar{U} + \Delta t F_{\text{label}}$,
2. Compute the translation velocity $U_{L_{0,1}}^t$ of the body $\Omega_{L_{0,1}}$, $U_{L_{0,1}}^t = \frac{\int_{\Omega} \rho^n U_{\text{col}} \cdot \chi_{L_{0,1}}^n dx}{\int_{\Omega} \rho^n \chi_{L_{0,1}}^n dx}$,
3. Compute the angular velocity $W^{L_{0,1}}$ of the body $\Omega_{L_{0,1}}$

$$W^{L_{0,1}} = J_{L_{0,1}}^{-1} \int_{\Omega} \rho^n r^n \times U_{\text{col}} \cdot \chi_{L_{0,1}}^n dx \times r^n, \quad \text{where } r^n = (x - x_{L_{0,1}}^g)$$

4. Compute the rigid velocity $U_{L_{0,1}}$ of the body $\Omega_{L_{0,1}}$, $U_{L_{0,1}} = U_{L_{0,1}}^t + W_{L_{0,1}}^t$
5. Correct the velocity using an implicit treatment of the penalization term

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

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:

$$x_g(L_{0,1}) = \frac{\int_{\Omega_{L_{0,1}}} \rho X}{\int_{\Omega_{L_{0,1}}} \rho} \quad (27)$$

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:

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

355 *Penalization algorithm 2*

The second algorithm consists in reconstructing the N level set functions using the fact that objects are rigid. The center of gravity is moved according to the velocity field, and the distance function the corresponding solid is recovered by a rigid motion of the initial one. Then, the steps 6 and 7 of the previous algorithm are modified as follows:

- 360 6. Transport the solid bodies by a rigid motion from the initial condition.
 7. Redefine the label maps and distance functions:

$$\forall x \in \Omega, \forall i \in \{1, \dots, N + 1\}, \begin{cases} L_0^{n+1}(x) = i \text{ if } \phi_i^{n+1} \leq 0 \\ L_1^{n+1}(x) = \arg \min_{j \neq L_0^{n+1}(x)} \phi_j^{n+1} \\ L_2^{n+1}(x) = \arg \min_{j \notin \{L_0^{n+1}(x), L_1^{n+1}(x)\}} \phi_j^{n+1} \end{cases}$$

$$\forall x \in \Omega, \begin{cases} \varphi_1^{n+1}(x) = d(x, \Gamma_{L_1^{n+1}(x)}) \\ \varphi_2^{n+1}(x) = d(x, \Gamma_{L_2^{n+1}(x)}) \end{cases}$$

Comparison of the two algorithms of evolutions

365 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. We fix the narrow-band sizes S_{L_1} to $10h$ and S_{L_2} to $5h$, where h is the space discretization step. Table 1 shows the CPU time of both algorithms, we can see that the CPU time of algorithm 2 is faster. However, the difference is small and the Algorithm 2 is provided in the case of circular rigid disks so that
 370 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 leading to an increase of the CPU time. It is hence most efficient to use the algorithm 1 for general shape of bodies.

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 1: Computational time of both algorithms.

6. Numerical illustrations

375 In this section, we present the numerical results obtained with the proposed model and give some comparisons with existing methods of the literature. In a first part, we investigate the space convergence of the method. Then we compare our model (23) to the model (19). In a third part, we provide a qualitative comparison of our model with those proposed in [31]. Finally, some simulations of dense suspensions of rigid bodies in two and three dimensions are presented.

For all of the simulations presented in this paper, the dynamic viscosity μ is set to 0.01, the density of the fluid is set to $\rho_f = 1$, and the density of the rigid bodies is the same for all the bodies.

380 Let $k = (k_x, k_y, k_z)^t$ be the repulsive coefficient between bodies and $k^{\text{wall}} = (k_x^{\text{wall}}, k_y^{\text{wall}}, k_z^{\text{wall}})^t$ be the repulsive coefficient exerted by the walls, the value of the components depends on the amplitude of the force.

6.1. Grid sensitivity

A grid convergence 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): (128×128) , (256×256) , (512×512) and (1024×1024) 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 (128×128) .

We take as test case the sedimentation of 25 circular rigid bodies having the same radius $R = 0.025 + \varepsilon$. The gravity force g is set to -980 and the repulsive coefficients on the bodies and on the walls are:

$$k_x = -g/10, \quad k_y = -g/10, \quad k_x^{\text{wall}} = -g/40, \quad k_y^{\text{wall}} = -g.$$

385 The repulsive forces are applied on a ring around the interface of thickness $1.5\Delta x^{G_1}$.

In Figure 7 the results obtained with the four different grids are presented, the white line corresponds to the numerical size of the rigid particles ($\phi = \varepsilon$). We can see that the dynamic of the rigid bodies and the interaction between them is quite similar for the different resolutions. The collisions are avoided thanks to the repulsive force. We observe the same phenomenon of kissing and tumbling of the bodies.

390 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
395 the various grids but is much closer on the two finest simulations. For the three dimensional case, we study the grid sensitivity using three different grids which contain respectively : (64^3) , (128^3) and (256^3) bodies on a uniform mesh. In the three dimensional case,

Figures 8 , 9 and 10 show the simulation of 200 rigid spheres of radius $R = 0.01$ falling under gravity.

400 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 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
405 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.

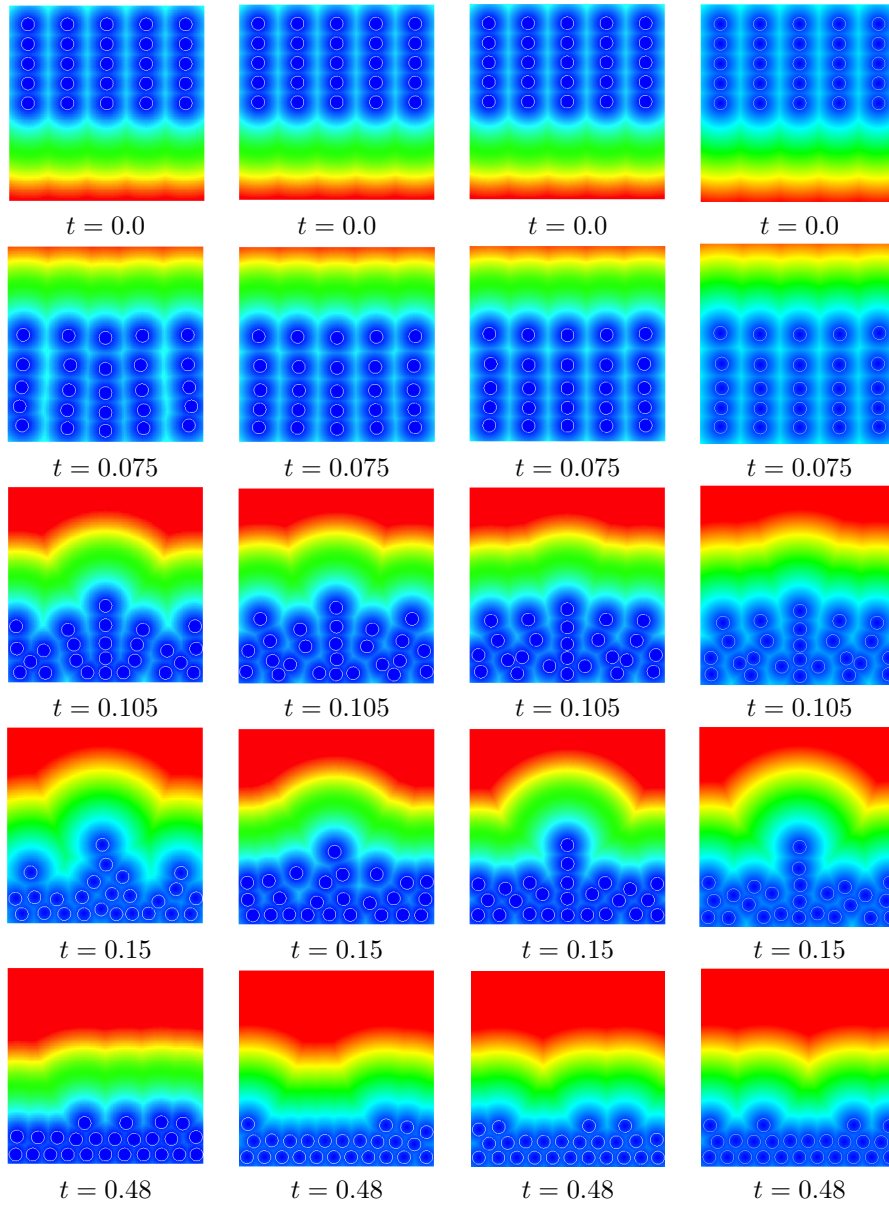


Figure 7: Study of the grid convergence with a test case of 25 rigid disks falling under gravity. From left to right, grid 128×128 , grid 256×256 , grid 512×512 and grid 1024×1024 . The background color shows the level set amplitude.

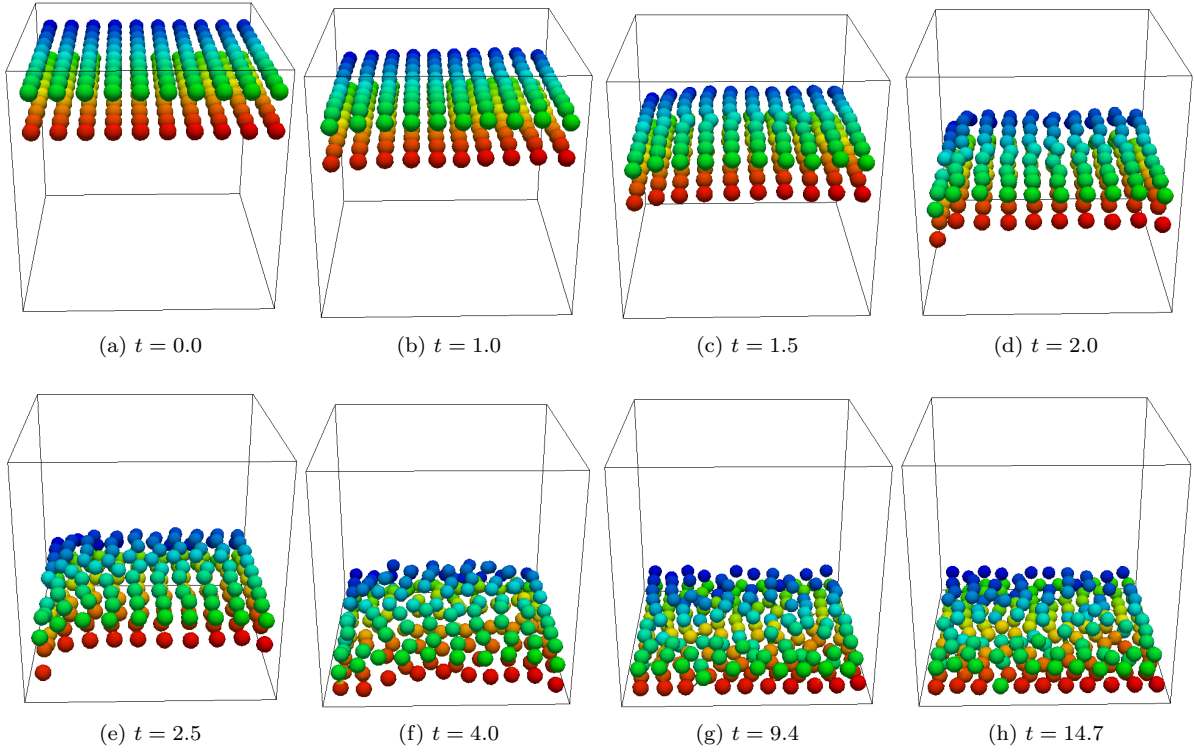


Figure 8: Simulation of 200 rigid spheres subject to gravity (grid resolution size 64^3). The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 200^{th} body and red for the fluid that is the 201^{th} object.

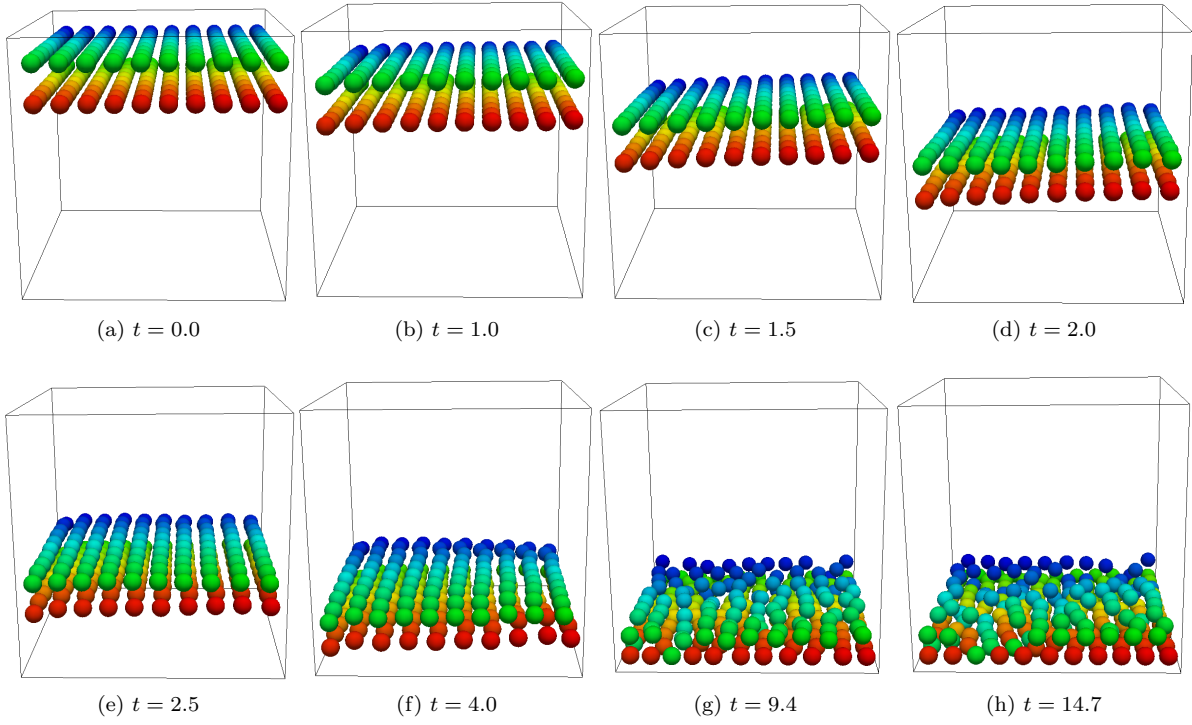


Figure 9: Simulation of 200 rigid spheres subject to gravity (grid resolution size 128^3). The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 200^{th} body and red for the fluid that is the 201^{th} object.

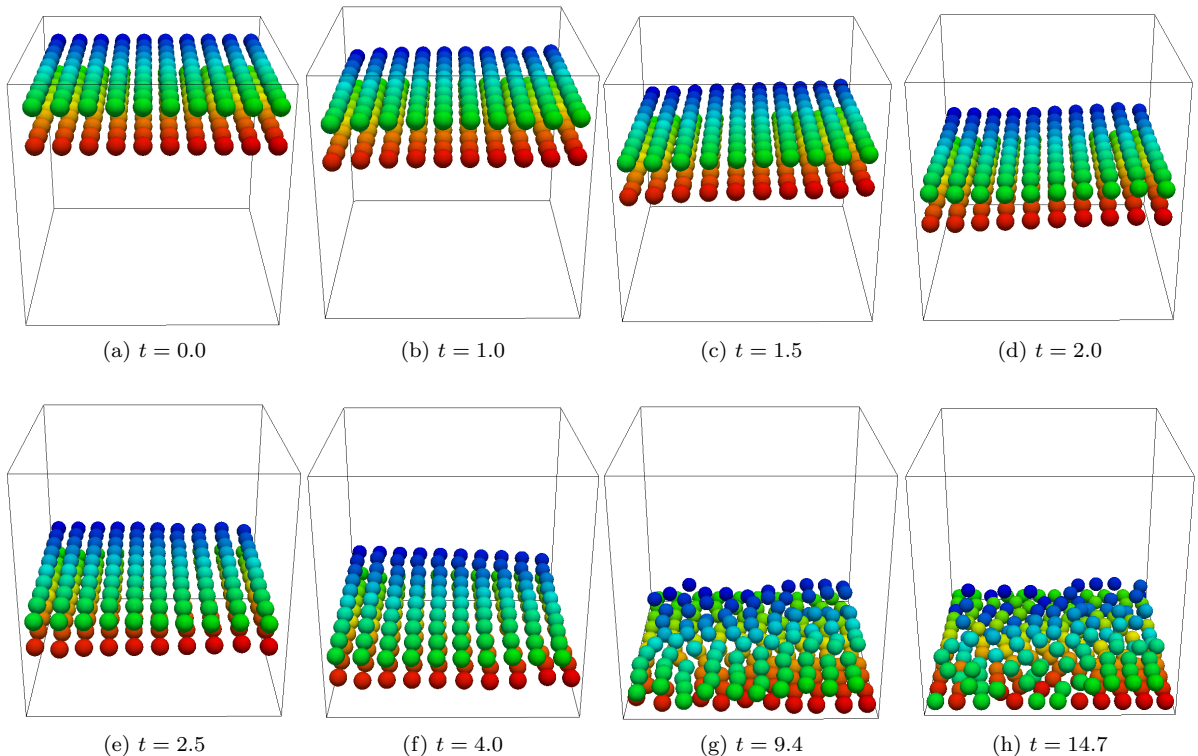


Figure 10: Simulation of 200 rigid spheres subject to gravity (grid resolution size 256^3). The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 200^{th} body and red for the fluid that is the 201^{th} object.

6.2. Comparison of the method with level set decomposition

We give here a comparison of our penalization model (23) and the penalization model that uses a level set decomposition (19)

6.2.1. Numerical efficiency

We investigate here the numerical efficiency of the two proposed algorithms (5.1) and (5.2) in the case of N rigid disks. The computational time is compared to the N level set method.

The algorithm associated to our model is given in section 5.2. 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 obvious that this algorithm depends on the number of bodies because of the N level set functions and their associated center of gravity. However, this part of the algorithm is very fast. This algorithm will allow us to compare the saving computational time which is induced by changing the existing collision model (6) and the 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 bodies. As noticed before the collision model (6) 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 bodies. The CPU time of the penalization model is larger in (19) because it depends on the number of bodies. Indeed, N rigid velocities must be computed to get the right velocity of each body. In the present algorithm, we must add the label redefinition that depends almost linearly on the number of bodies and so is quite cheap. For a low number of bodies the CPU time of both methods is close but increasing this number

430 from 2 to 400 bodies induces a total CPU time 8000 larger for the model (19) whereas with the present model the CPU time is only 25 times larger. So our method is around 340 times faster for 400 bodies.

Number of disks	Collision model (6) CPU time	Penalization model (19) 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 2: Averaged CPU time using the N level set decomposition

Number of disks	Model (7) CPU time	Model (23) 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 3: Averaged CPU time using the algorithm of section 5.2.1

6.2.2. Numerical comparison of the two collision models

To highlight the differences between the two collision models we first focus on 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 spatial discretization step $\Delta x = 7.8125 \cdot 10^{-2}$ 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 are:

$$k_x = -g/10, \quad k_y = -g/10, \quad k_x^{\text{wall}} = -g/40, \quad k_y^{\text{wall}} = -g.$$

435 in Figure 11, black and white lines represents the interfaces of the three different objects. The black line stands for the collision model (6) and the white line for the collision model (7) 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 12 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 14 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.

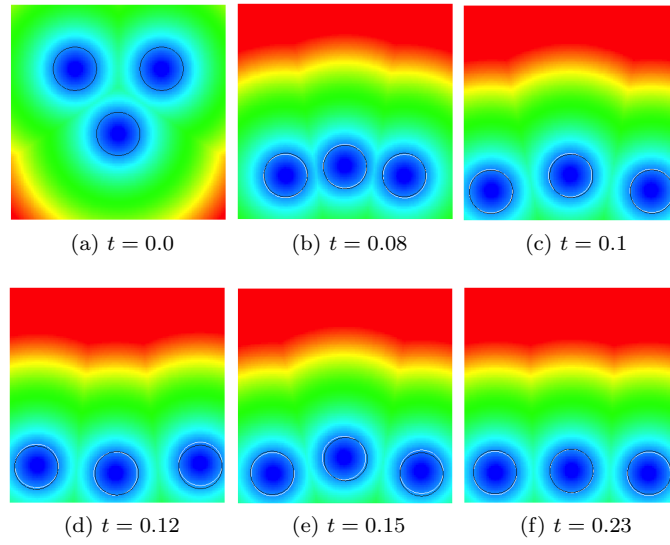


Figure 11: Comparison of the two collision models for three disks of radius $R = 0.1$. The background colors show the level set amplitude.

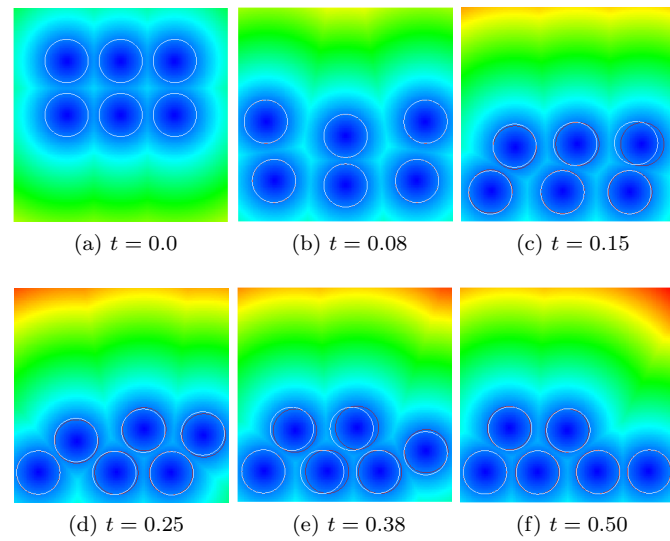


Figure 12: Comparison of the two collision models for six disks of radius $R = 0.1$. The background colors show the level set amplitude.

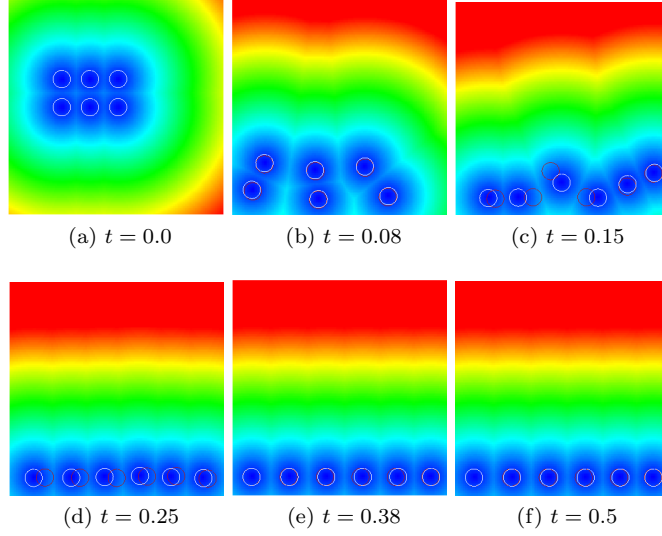


Figure 13: Comparison of the two collision models for six disks of radius $R = 0.03$. The background colors show the level set amplitude.

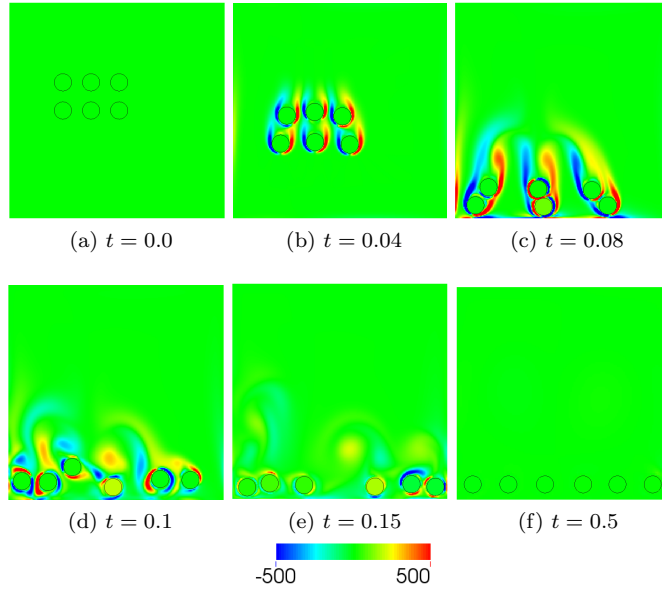


Figure 14: Six disks falling under gravity, colors represents the values of the vorticity field.

445 6.3. Comparison with the model introduced in [31]

In this part, we compare qualitatively our model to an existing method proposed in [31]. In that model, the solid bodies are taken into account by penalizing the strain tensor to enforce the rigid body motion (see [32]). A scheme for inelastic collisions is implemented imposing a minimal distance between bodies and therefore avoiding contacts (see [33] for more details). Moreover, the contacts with the four walls are also handled in the same way. The test case is the sedimentation of 100 rigid particles of radius $R = 0.01$ subject to the gravity force $g = -70$. The corresponding repulsive coefficients are:

$$k_x = -g/7, \quad k_y = -g/7, \quad k_x^{\text{wall}} = -g/28, \quad k_y^{\text{wall}} = -g.$$

450 The Figure 15 shows the results obtained with the FreeFem code implemented by A. Lefebvre (see [31]) 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 16. The computation on the FreeFEM++ code was performed on a non-symmetrical mesh so that the results in Figure 15 are unsymmetric from the beginning whereas our results, performed on a cartesian grid stay symmetric until time $t = 1.44$. However, we can see that the dynamics is globally the same and that the static equilibrium is reached at the same time $t = 4.8$.

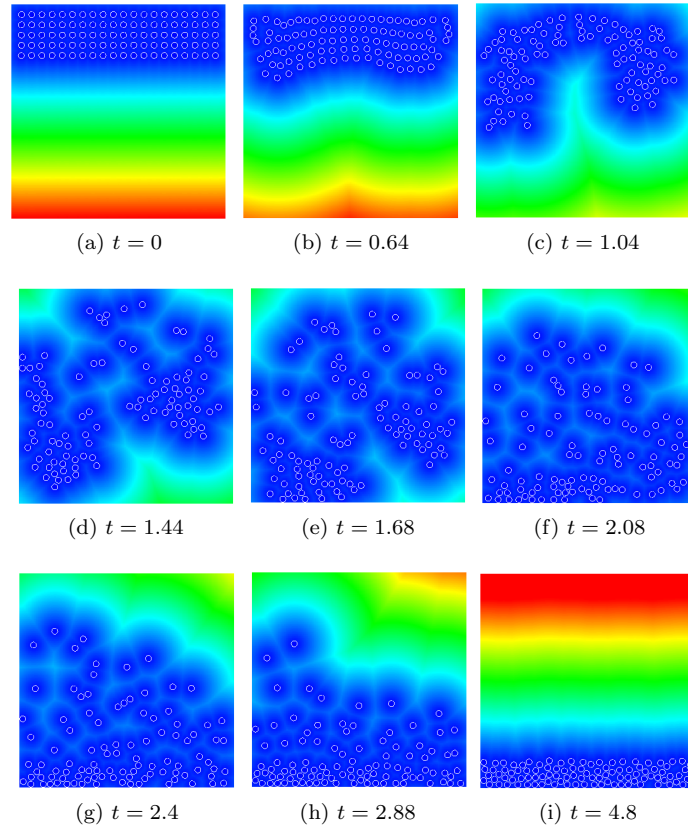


Figure 15: Simulation of 100 rigid particles submitted to gravity obtained with the FreeFem code [31]

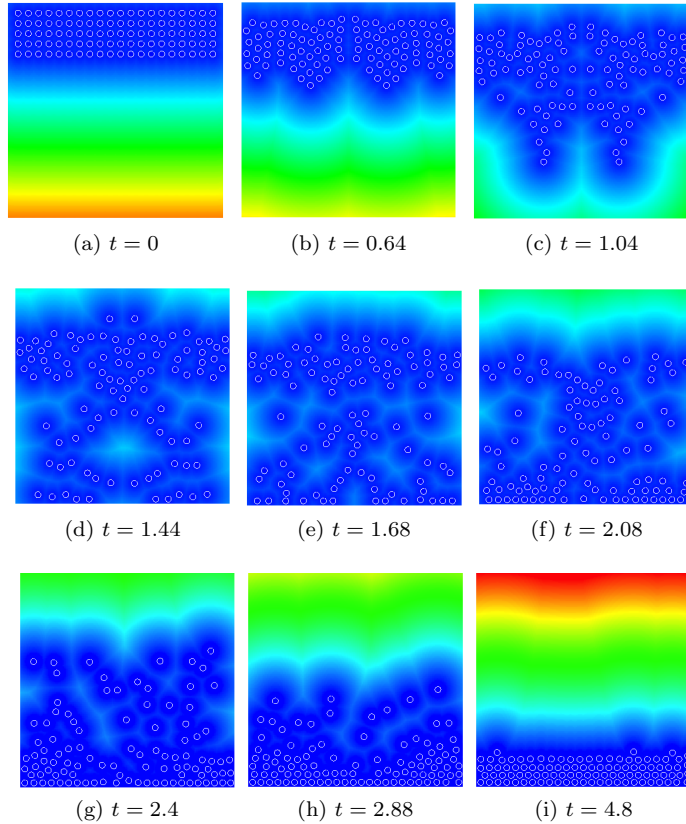


Figure 16: Simulation of 100 rigid particles submitted to gravity obtained with our model. The background colors show the level set amplitude.

7. 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
 455 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 $\sqrt{\pi}R$.
 460 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 18 and 17 show the sedimentation of 30 rigid squares of different sizes. In the presence of
 465 rigid particles, the hydrodynamical instabilities are more prominent, as initially the particles are closed, the
 symmetry loss is almost immediate. Once this flow 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 in Figures 20,19 where the vorticity field is represented at different time. At first the vorticity is
 470 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 point out that globally the behavior of the rigid bodies is 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 17 than the particles in Figure 18, 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 in Figure 21. 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 closed, meaning that the repulsive constant coefficient may not be well calibrated.

8. 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 Figure 22.

The second simulation addresses the 3D case. Figure 23 show the simulation of 500 rigid spheres of radius $R = 0.01$ falling under gravity using a grid of size 128^3 . The half thickness of the interface is $\varepsilon = \Delta x$. 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). In that case the repulsive forces are negligible. The interactions between bodies occur at once after $t = 1.5$ on the fine mesh. As a consequence, the equilibrium state is reached fastly around $t = 2.5$ instead of $t = 9.4$.

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 $\varepsilon = 1.5\Delta_x \simeq 5.9 \cdot 10^{-3}$. 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.

9. Application to vesicle simulation

The purpose of this section 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 describe the Eulerian fluid/elastic membrane coupling model. Then, we present the numerical procedure used for the simulations. Finally, a validation of the model in the case of multiple vesicles under various flow are presented.

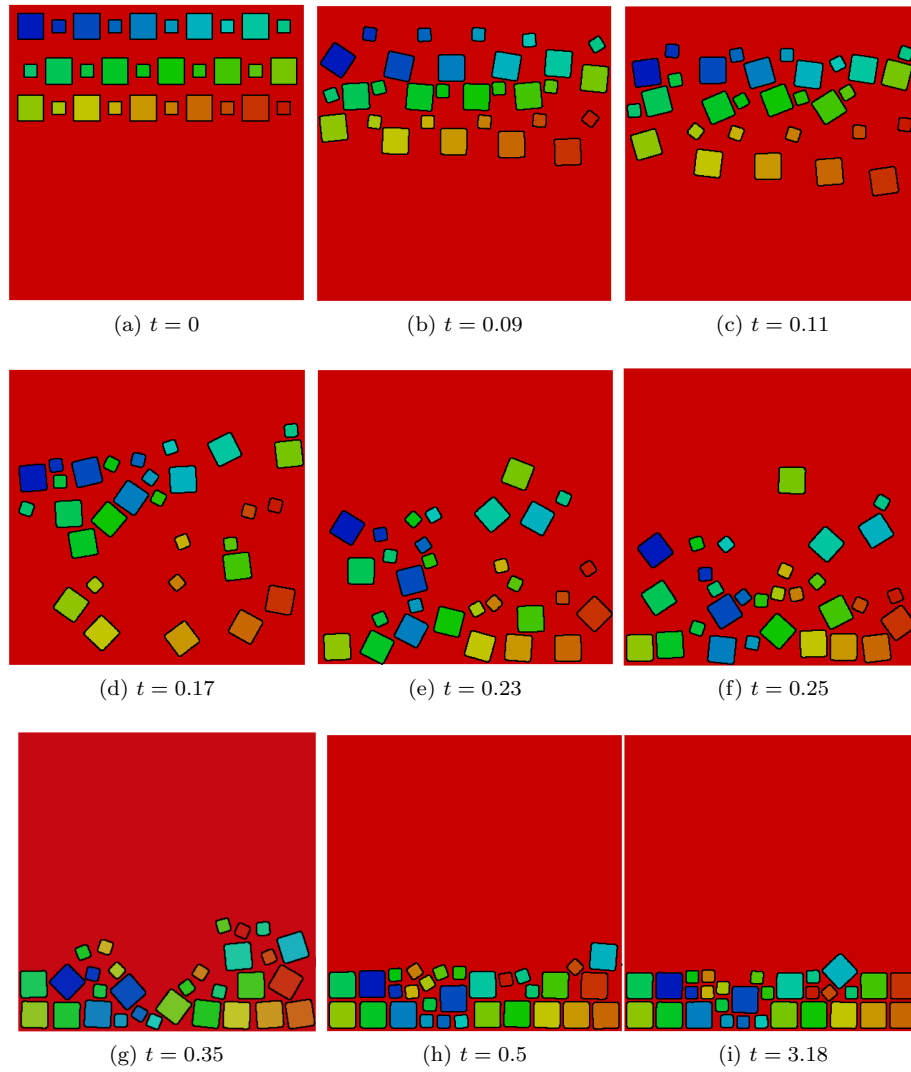


Figure 17: Simulation of 30 square rigid bodies of different size falling under gravity. The intensity of the force depends on the relative velocities of the closest particles.

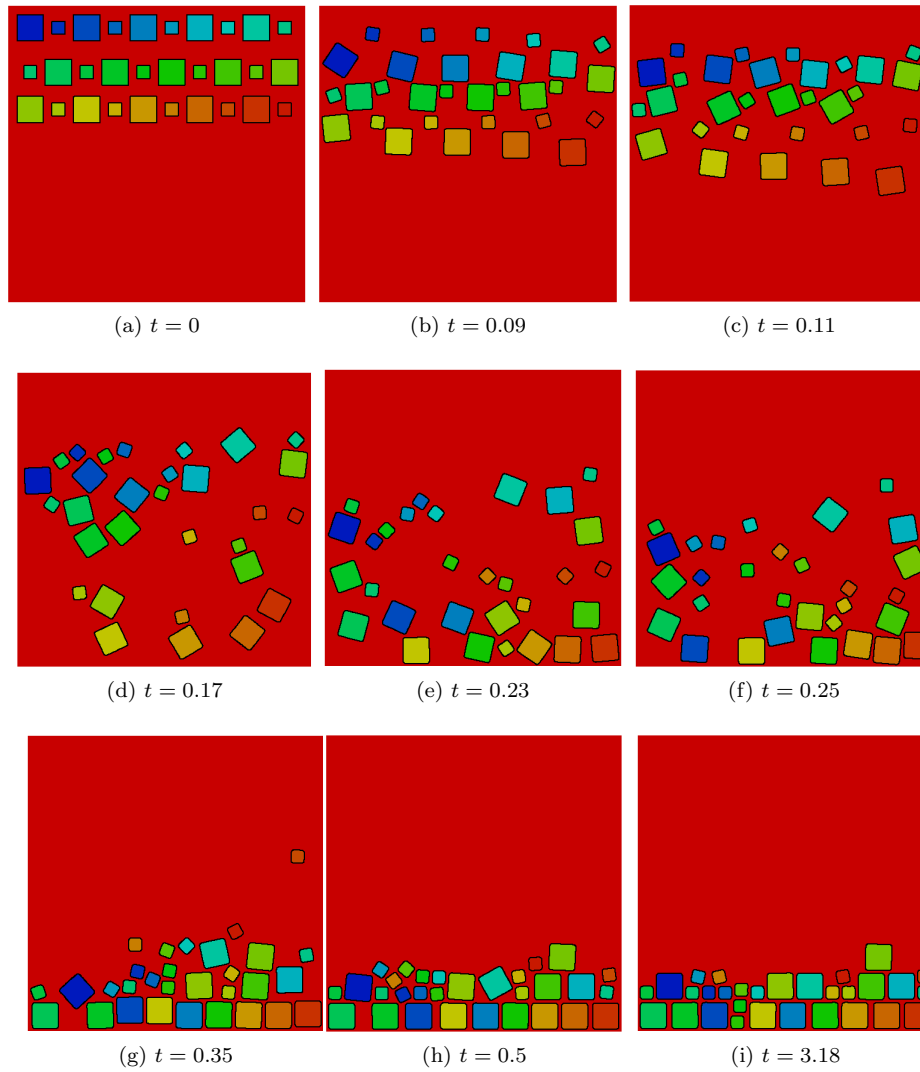


Figure 18: Simulation of 30 square rigid bodies of different size falling under gravity. The intensity of the force is constant $k = 100$.

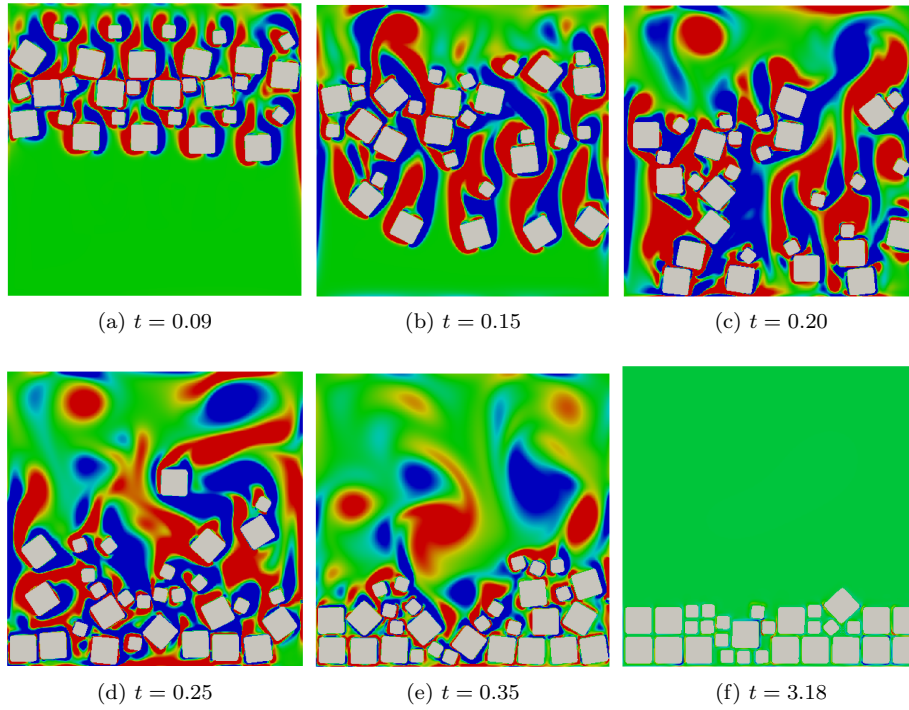


Figure 19: Simulation of 30 square rigid bodies of different size falling under gravity. The intensity of the force depends on the relative velocities of the closest particles. The background colors represents the vorticity field intensity.

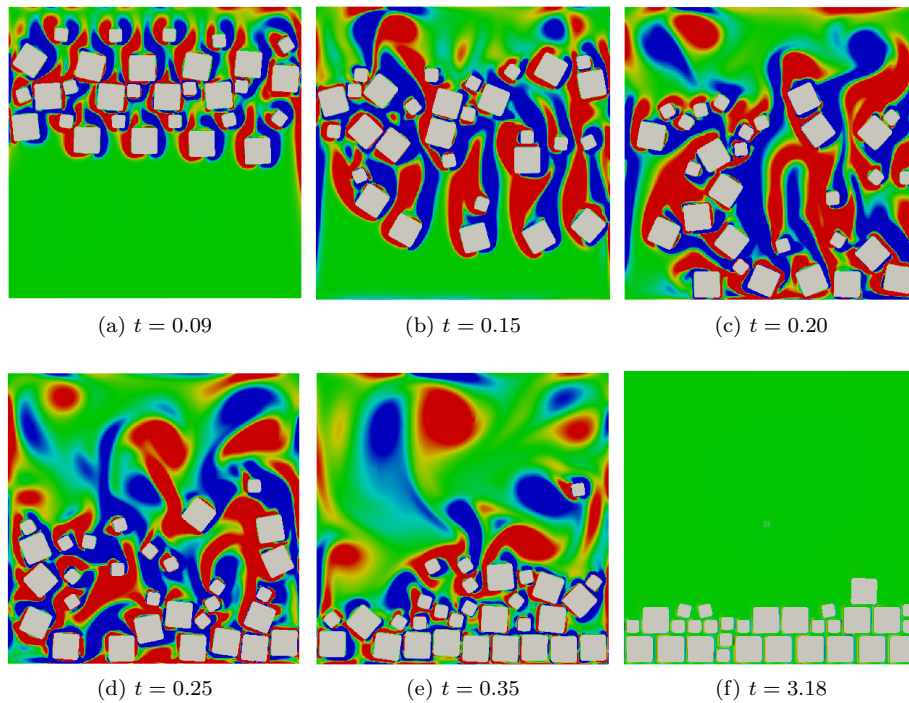


Figure 20: Simulation of 30 square rigid bodies of different size falling under gravity. The intensity of the force is constant. The background colors represents the vorticity field intensity.

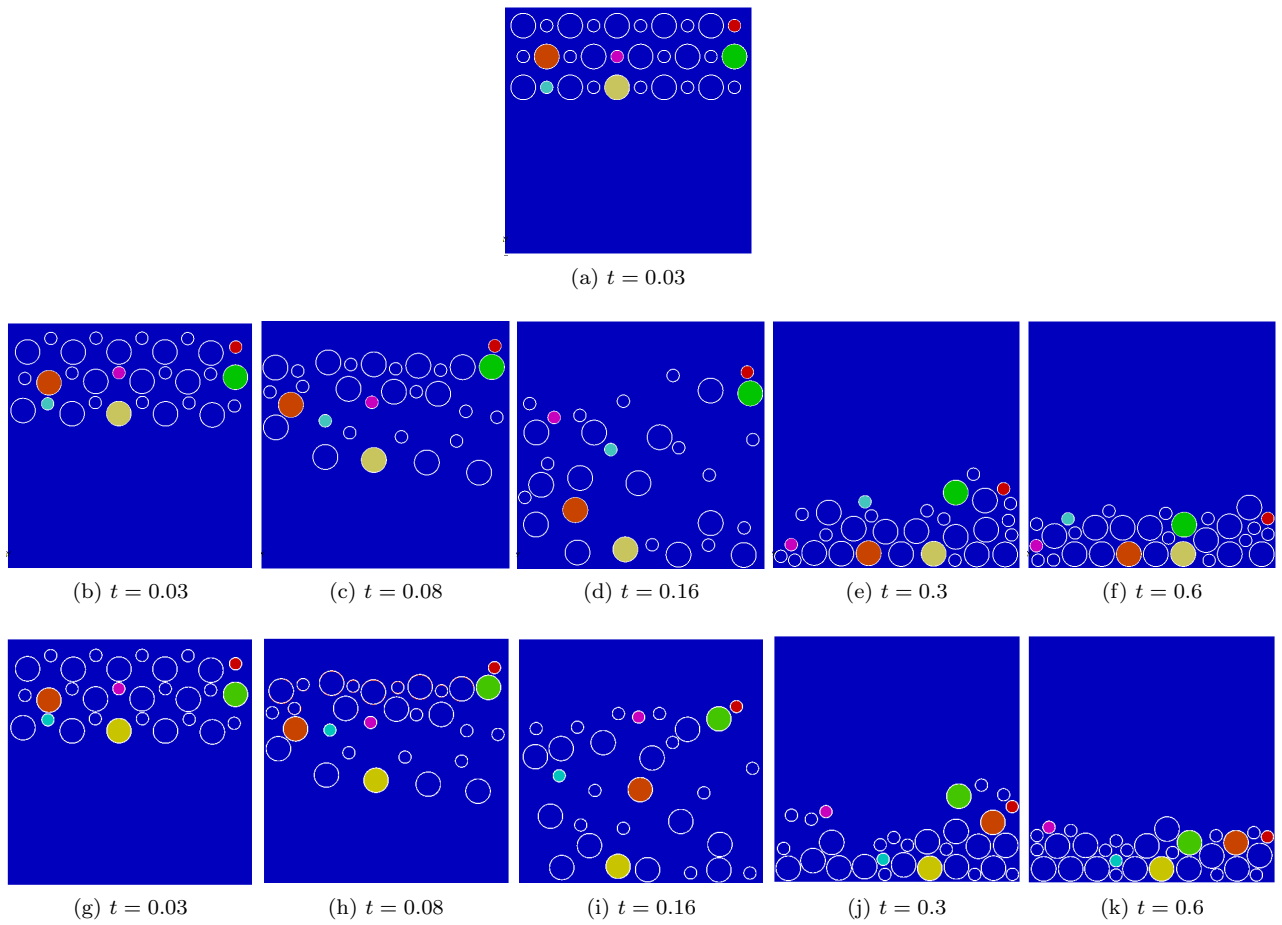


Figure 21: Simulations of 30 rigid bodies of different radii ($R = 0.05$ or $R = 0.025$) falling under gravity. The top simulation has been achieved using a constant intensity of the force whereas for the bottom one it depends on relative velocities.

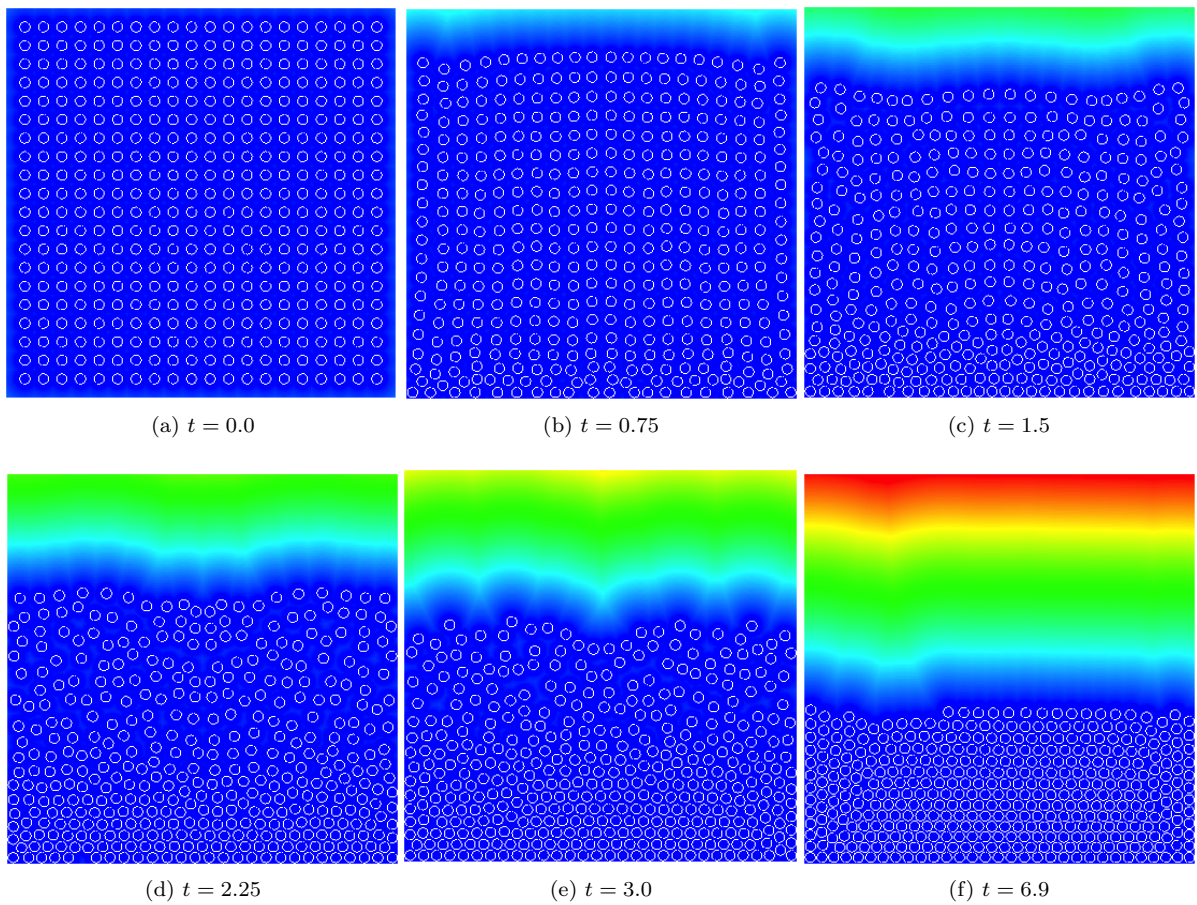


Figure 22: Simulation of 400 rigid disks submitted to gravity (the white line corresponds to the level line $\phi = \varepsilon$). The background colors show the level set amplitude.

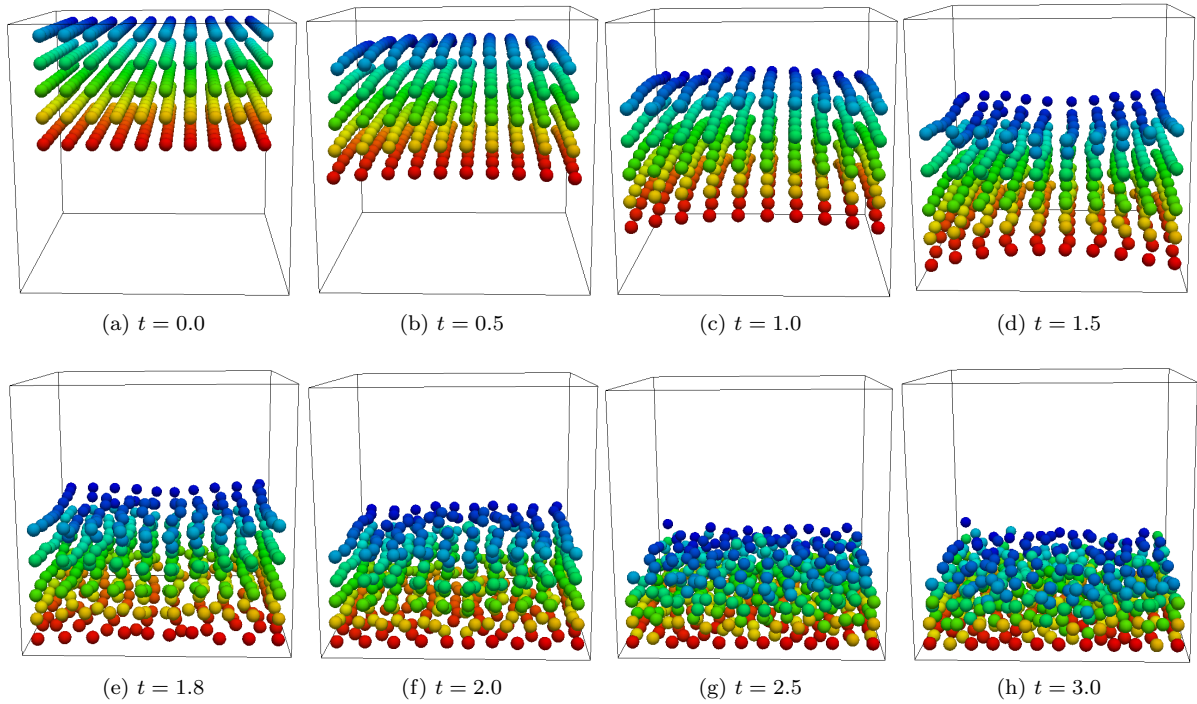


Figure 23: Simulation of 500 rigid spheres subject to gravity (grid resolution size 128^3). The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 500^{th} body and red for the fluid that is the 501^{th} object.

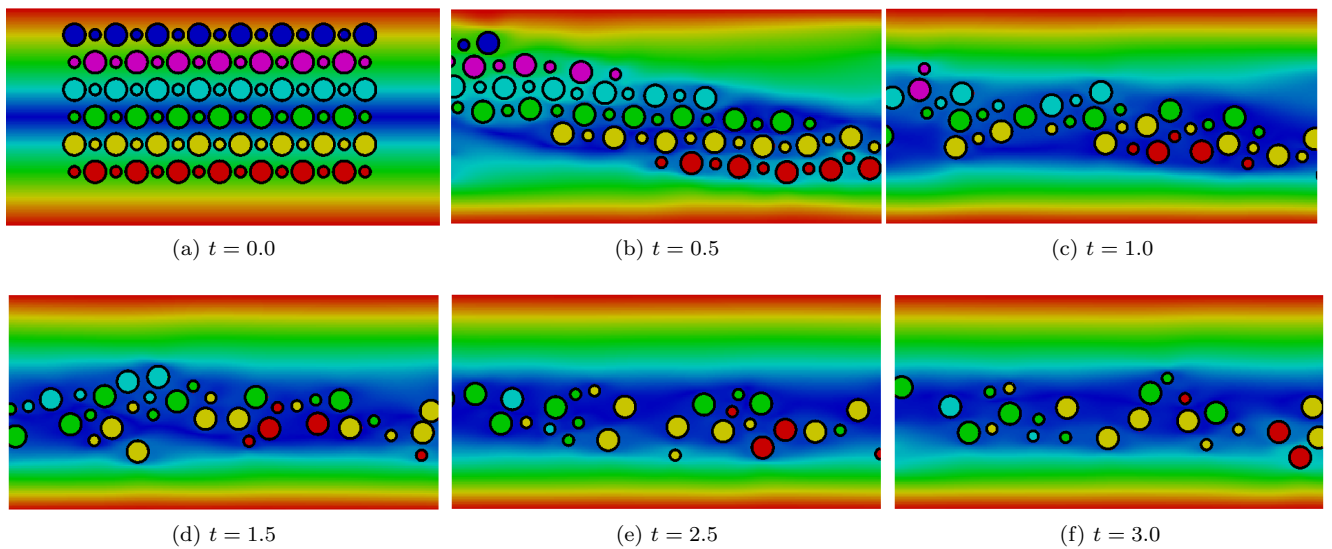


Figure 24: Rigid spheres suspended in a shear flow. The rigid disks are colored with the values of the label map L_0 and the background colors indicates the magnitude of the velocity.

9.1. Eulerian fluid/elastic membranes coupling model

We present here the Eulerian fluid/elastic membrane coupling model introduced in [13] in the case of multiple vesicles and using the label maps and distance functions.

9.1.1. Computation of the elastic and bending forces

525 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.

To reduce the number of grid points required in the narrow-gap between two interfaces, we use the distance function φ_2 to compute the forces of each vesicle Γ_i^ε in the regions where the level set function ϕ is not defined as the signed distance function to Γ_i . By doing this, the stencil required to compute a geometrical property at a point x belonging to Γ_i^ε is prolonged to the regions where $L_2 = i$. Numerically, we introduce a function which indicate the points where this procedure has to be applied. This procedure is used in order to compute the curvature and other derivatives of order higher than two of the level set function ϕ .

535 9.1.2. Collision model

The short-range repulsive force 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:

$$\forall x \in \Omega, \quad |\Gamma_{L_{0,1}}(x)| = \int_{\Gamma_{L_{0,1}}(x)} dz = \int_{\Omega} \frac{1}{\varepsilon} \zeta_\varepsilon(\phi_{L_{0,1}}(x)(z)) dz \quad (28)$$

we obtain the following average velocity over the membrane boundary:

$\forall x \in \Omega,$

$$U_a(L_{0,1}(x)) = \frac{1}{|\Gamma_{L_{0,1}}|} \int_{\Gamma_{L_{0,1}}(x)} U(z) dz = \int_{\Omega} \frac{1}{\varepsilon} \zeta_\varepsilon(\phi_{L_{0,1}}(x)(z)) U(z) dz \quad (29)$$

545 then, at all points $x \in \Gamma_i^\varepsilon, i = (1, \dots, N)$ 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:

$$\forall x \in \Omega, \quad F_{\text{label}}(x) = \frac{|(U_a(L_2(x)) - U_a(L_{0,1}(x))) \cdot \nabla \varphi_2(x)|}{\varepsilon} \rho(x) \zeta_\varepsilon(\varphi_1(x)) \frac{\nabla \varphi_2(x)}{\varphi_2(x)} \exp\left(-\frac{\varphi_2(x)}{\varepsilon_b}\right) \quad (30)$$

9.1.3. Complete model

550 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 :

$$\begin{cases} \rho(\phi)(\partial_t U + (U \cdot \nabla)U) - \text{div}(\mu(\phi)D(U)) + \nabla P = F_e + F_c + F_{\text{label}} & \text{in } \Omega_T = \Omega \times (0, T) \\ \text{div } u = 0 & \text{in } \Omega_T = \Omega \times (0, T) \\ \partial_t \phi + U \cdot \nabla \phi = 0 & \text{in } \Omega_T = \Omega \times (0, T) \\ \partial_t e + U \cdot \nabla e = -e(n(\phi) \otimes n(\phi)) : D(U) & \text{in } \Omega_T \end{cases} \quad (31)$$

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

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

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 length, velocity, density and viscosity scales. Considering the following characteristic quantities :

$$\begin{cases} x = L_r x', & y = L_r y', & z = L_r z' \\ U = U_r U', & P = \rho_r \frac{U_r}{L_r} P', & t = \frac{L_r}{U_r} t' \\ \mu = \mu_r \mu', & \rho = \rho_r \rho', & \varepsilon = L_r \varepsilon' \\ \phi = L_r \phi', & \varphi_1 = L_r \varphi_1', & \varphi_2 = L_r \varphi_2' \end{cases}$$

Differentiating (and dropping '), we obtain the following dimensionless system:

$$\begin{cases} \rho(\phi)(\partial_t U + (U \cdot \nabla)U) - \frac{1}{R_e} \operatorname{div}(\nu(\phi)D(U)) + \nabla P = \frac{1}{W_e} \bar{F}_e(\phi) + \frac{1}{W_c} \bar{F}_c(\phi) + F_{\text{label}} & \text{in } \Omega_T \\ \operatorname{div} U = 0 & \text{in } \Omega_T \\ \partial_t \phi + U \cdot \nabla \phi = 0 & \text{in } \Omega_T \\ \partial_t e + U \cdot \nabla e = -e(n(\phi) \otimes n(\phi)) : D(U) & \text{in } \Omega_T \end{cases} \quad (32)$$

555 where the physical parameters are given by: $R_e = \frac{L_r U_r \rho_r}{\mu_r}$, $W_e = \frac{\mu_r U_r}{\lambda}$, $W_c = \frac{\mu_r U_r L_r^2}{\alpha}$

9.2. 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. 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. 560 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.

9.2.1. 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:

$$\Delta t \leq \frac{\Delta x}{|U|_\infty}$$

The explicit treatment of the fluid/elastic coupling induces stability conditions. The Brackbill condition provided in [34] gives a stability condition in the case of perfect fluids separated by an interface with surface 565 tension, expressed as:

$$\Delta t \leq \sqrt{\frac{W_e}{2\pi}} (\Delta x)^{\frac{3}{2}} \quad (33)$$

where W_e decreases while the surface tension coefficient increases. This condition is valid for high Reynolds number, where viscosity of the fluid could be neglected. In the case of small W_e , which is the case of red blood cells, the time step has to be chosen very small. In the case of small Reynolds number, [35] exhibited the stability condition:

$$\Delta t \leq \sqrt{\frac{W_e}{R_e}} \Delta x \quad (34)$$

570 An unifying condition has been proposed by [36, 37] 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 and build a maximal time-step on the above conditions.

575 *9.2.2. 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: $d_{L_1} = 10h$ 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 $d_{L_2} = 10h$ this allows to compute correctly the bending and elastic forces when the vesicles get closer by means of the reconstituted level set functions. 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

590 *9.3. Numerical illustrations*

This section is dedicated to the numerical results obtained with the proposed model. We first provide a grid convergence of the algorithm then numerical simulations of multiple vesicles in several type of external hydrodynamical flow are presented.

9.3.1. Grid sensitivity

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 and at initialization the shape of vesicles are Cassini Oval 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 26, their shape progressively change during their motion in the canal. Figure 25 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.

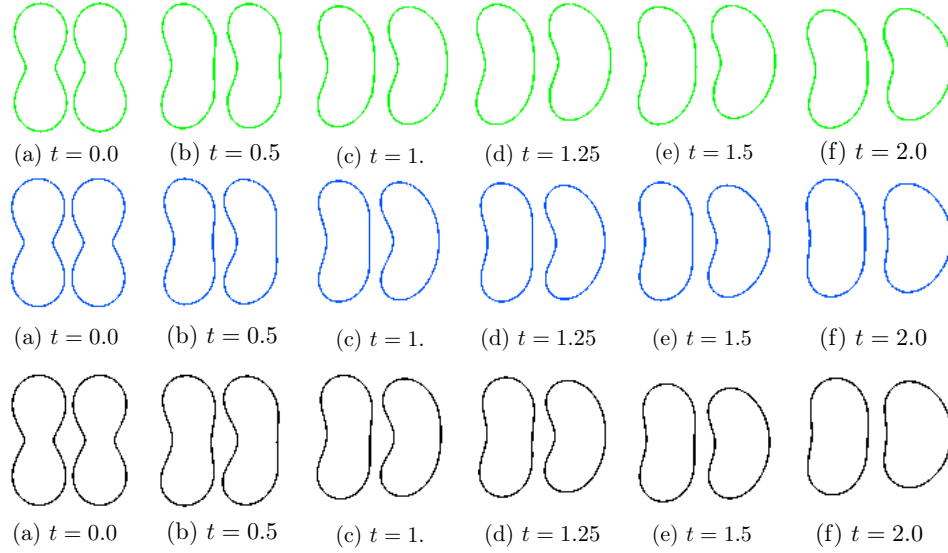


Figure 25: Simulation of two vesicles in a Poiseuille flow performed on three grid levels. From top to bottom, the associated discretization space steps are $h = 7.81 \times 10^{-3}$, $h \simeq 3.90 \times 10^{-3}$, and $h \simeq 1.95 \times 10^{-3}$.

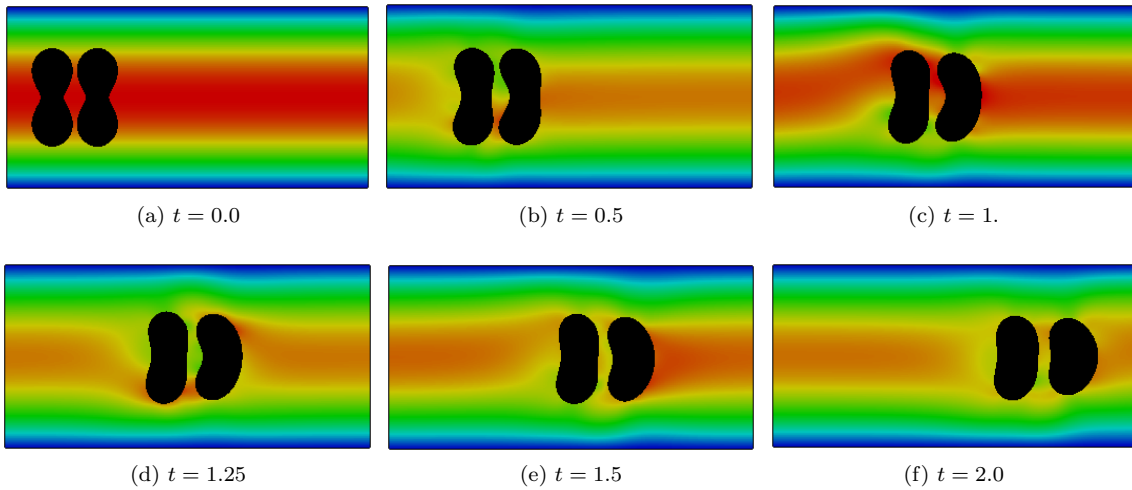


Figure 26: Deformation of two vesicles in a Poiseuille flow performed on the grid G_{512} of size (1024×512) .

9.3.2. 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. We set the following parameters:

$$R_e = 0.01, W_e = 0.005, W_c = 30$$

The simulation represented in Figure 27 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.

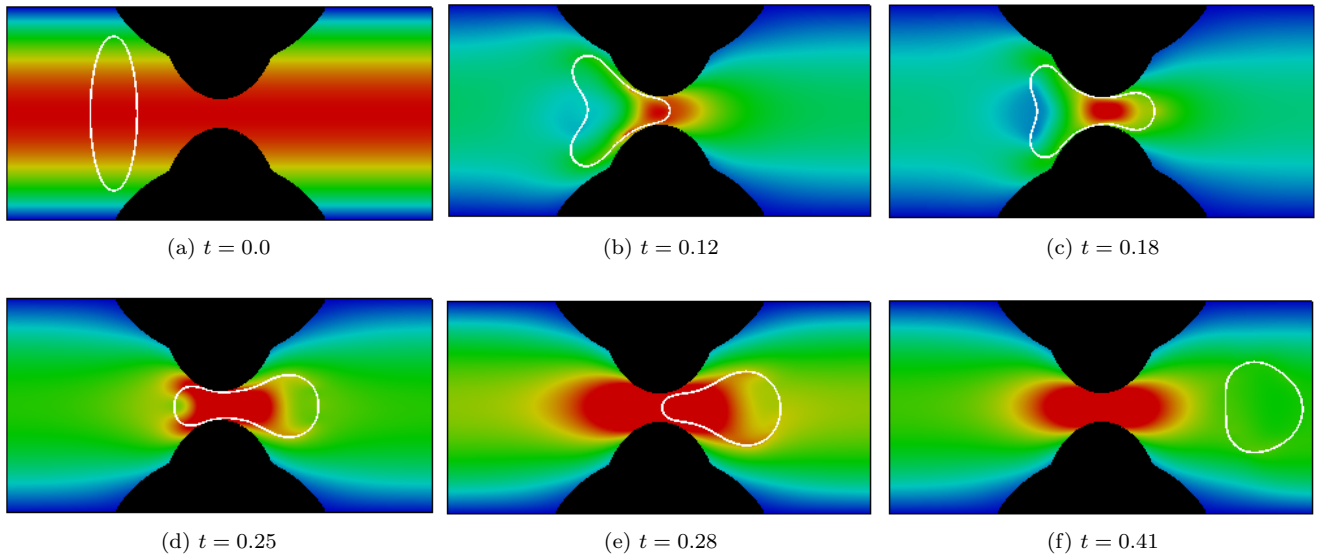


Figure 27: Simulation of a vesicle passing through a bifurcation. The background color corresponds to the magnitude of the velocity. The white contour represents the isoline $\phi = \varepsilon$.

9.3.3. 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 in Figure 28, 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.

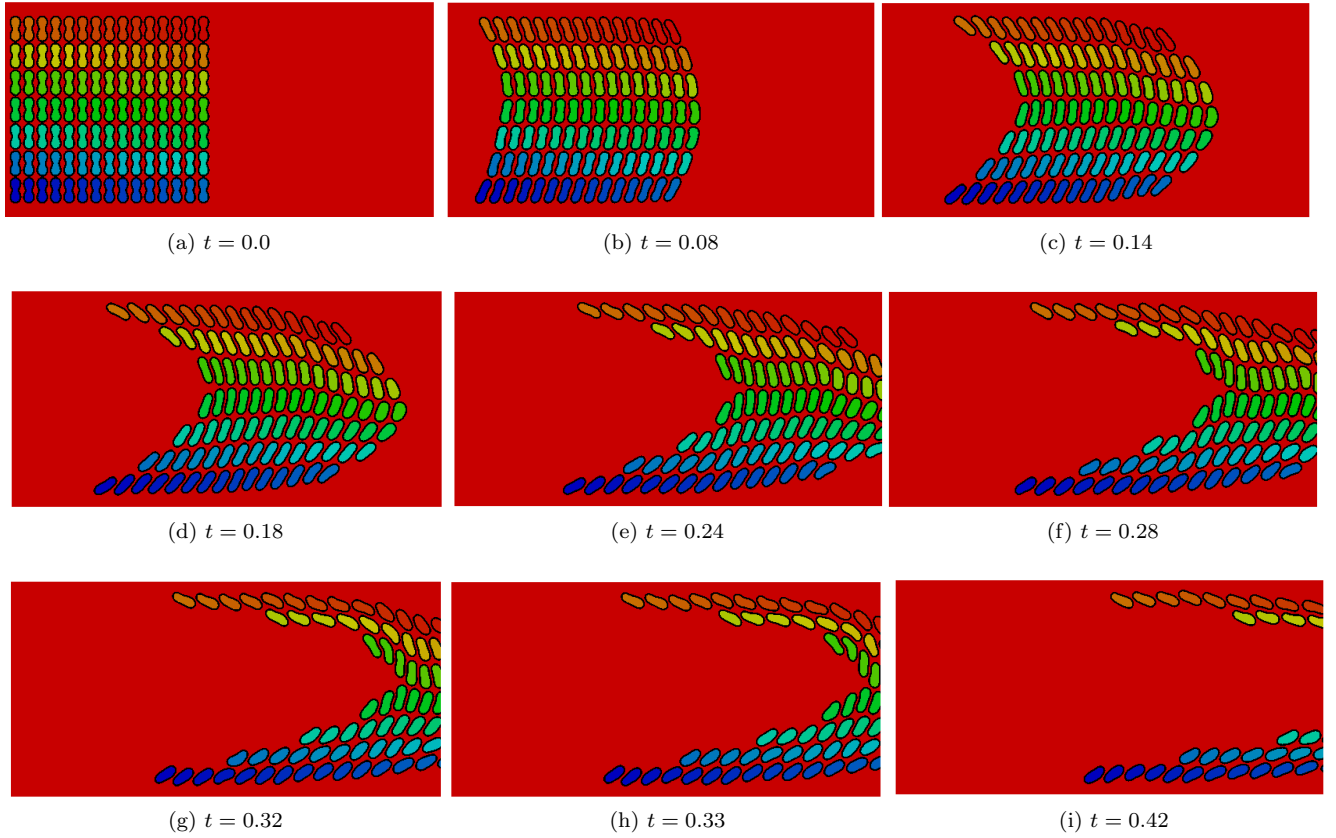


Figure 28: Simulation of 105 vesicles in a Poiseuille flow. The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 105th body and red for the fluid that is the 106th object.

10. Conclusions

In this paper, we introduced a new model to simulate efficiently a large number of interacting bodies immersed in a fluid. This model involved three label maps and two distance functions which allow to locate the bodies and their closer neighbours in the domain. A collision model depending on the distance between the closest bodies is proposed. This model which is totally independent on the number of bodies, is compared both theoretically and numerically to the model introduced in [17]. We present an application to rigid structures with a penalisation model that only depends on five advected field functions.

Numerical results are in good agreement with the results of the literature at least qualitatively. Compared to a model which is totally dependent on the number of bodies, our model substantially reduces the CPU time. A numerical test on bodies of various radii shows that the collision model is efficient even when the strength of the force is very different.

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. 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.

Acknowledgment

Most of the numerical simulations have been run on PLAFRIM platform supported by IMB University of Bordeaux and INRIA Bordeaux - Sud Ouest.

References

- 655 [1] J. Donea, S. Giuliani, J. Halleux, An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions, *Computer methods in applied mechanics and engineering* 33 (1) (1982) 689–723.
- [2] C. Grandmont, Y. Maday, Fluid-structure interaction: a theoretical point of view, *Revue européenne des éléments finis* 9 (6-7) (2000) 633–653.
- 660 [3] M. A. Fernández, J.-F. Gerbeau, 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) (2007) 794–821.
- [4] J. Hron, S. Turek, A monolithic FEM/multigrid solver for an ALE formulation of fluid-structure interaction with applications in biomechanics, Springer, 2006.
- [5] T. Richter, T. Wick, Finite elements for fluid–structure interaction in ale and fully eulerian coordinates, *Computer Methods in Applied Mechanics and Engineering* 199 (41) (2010) 2633–2642.
- 665 [6] T. Wick, Fluid-structure interactions using different mesh motion techniques, *Computers & Structures* 89 (13) (2011) 1456–1467.
- [7] C. Conca, A. Osses, J. Planchard, Added mass and damping in fluid-structure interaction, *Computer methods in applied mechanics and engineering* 146 (3) (1997) 387–405.
- [8] M. A. Fernández, J.-F. Gerbeau, C. Grandmont, A projection algorithm for fluid–structure interaction problems with strong added-mass effect, *Comptes Rendus Mathématique* 342 (4) (2006) 279–284.
- 670 [9] M. A. Fernández, Coupling schemes for incompressible fluid-structure interaction: implicit, semi-implicit and explicit, *SeMA Journal* 55 (1) (2011) 59–108.
- [10] M. A. Fernández, Incremental displacement-correction schemes for incompressible fluid-structure interaction, *Numerische Mathematik* 123 (1) (2013) 21–65.
- 675 [11] M. A. Fernández, M. Landajuela, M. Vidrascu, Fully decoupled time-marching schemes for incompressible fluid/thin-walled structure interaction, *Journal of Computational Physics* 297 (2015) 156–181.
- [12] M. A. Fernández, J. Mullaert, 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) (2015) 199–229.
- 680 [13] G.-H. Cottet, E. Maitre, A level-set formulation of immersed boundary methods for fluid-structure interaction problems, *Comptes Rendus Mathématique* 338 (7) (2004) 581–586.
- [14] G.-H. Cottet, E. Maitre, A level set method for fluid-structure interactions with immersed surfaces, *Mathematical models and methods in applied sciences* 16 (03) (2006) 415–438.
- [15] C. S. Peskin, The immersed boundary method, *Acta numerica* 11 (2002) 479–517.
- 685 [16] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations, *Journal of computational physics* 79 (1) (1988) 12–49.
- [17] M. Coquerelle, 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) (2008) 9121–9137.
- [18] M. Y. Wang, 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) (2004) 469–496.
- 690 [19] M. Hillairet, Lack of collision between solid bodies in a 2d incompressible viscous flow, *Communications in Partial Differential Equations* 32 (9) (2007) 1345–1371.
- [20] H. H. Hu, Direct simulation of flows of solid-liquid mixtures, *International Journal of Multiphase Flow* 22 (2) (1996) 335–352.
- 695 [21] B. Maury, A many-body lubrication model, *Comptes Rendus de l’Académie des Sciences-Series I-Mathematics* 325 (9) (1997) 1053–1058.
- [22] S. Dance, M. Maxey, Incorporation of lubrication effects into the force-coupling method for particulate two-phase flow, *Journal of computational Physics* 189 (1) (2003) 212–238.
- [23] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, A distributed lagrange multiplier/fictitious domain method for particulate flows, *International Journal of Multiphase Flow* 25 (5) (1999) 755–794.
- 700 [24] J. A. Bogovic, J. L. Prince, P.-L. Bazin, A multiple object geometric deformable model for image segmentation, *Computer Vision and Image Understanding* 117 (2) (2013) 145–157.
- [25] M. Jedouaa, Interface capturing methods for interacting immersed objects, Ph.D. thesis, Université Grenoble Alpes (2017).
- [26] J. A. Sethian, A fast marching level set method for monotonically advancing fronts, *Proceedings of the National Academy of Sciences* 93 (4) (1996) 1591–1595.
- 705 [27] E. Sifakis, G. Tziritas, Moving object localisation using a multi-label fast marching algorithm, *Signal Processing: Image Communication* 16 (10) (2001) 963–976.
- [28] E. Rouy, A. Tourin, A viscosity solutions approach to shape-from-shading, *SIAM Journal on Numerical Analysis* 29 (3) (1992) 867–884.
- 710 [29] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, *Numerische Mathematik* 81 (4) (1999) 497–520.

- [30] N. Patankar, A formulation for fast computations of rigid particulate flows, Center for Turbulence Research Annual Research Briefs 2001 (2001) 185–196.
- [31] A. Lefebvre, Fluid-particle simulations with freefem++, in: ESAIM: Proceedings, Vol. 18, EDP Sciences, 2007, pp. 120–132.
- 715 [32] J. Janela, A. Lefebvre, B. Maury, A penalty method for the simulation of fluid-rigid body interaction, in: ESAIM: Proceedings, Vol. 14, EDP Sciences, 2005, pp. 115–123.
- [33] B. Maury, A time-stepping scheme for inelastic collisions, Numerische Mathematik 102 (4) (2006) 649–679.
- [34] J. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, Journal of computational physics 100 (2) (1992) 335–354.
- 720 [35] C. Galusinski, P. Vigneaux, Level-set method and stability condition for curvature-driven flows, Comptes Rendus Mathématique 344 (11) (2007) 703–708.
- [36] G.-H. Cottet, E. Maitre, A semi-implicit level set method for multiphase flows and fluid–structure interaction problems, Journal of Computational Physics 314 (2016) 80–92.
- 725 [37] C. Bost, Méthodes level-set et pénalisation pour le calcul d’interactions fluide-structure, Ph.D. thesis, Université Joseph-Fourier-Grenoble I (2008).