

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

M. Jedouaa¹, C-H. Bruneau² & E. Maitre¹

¹Laboratoire Jean Kuntzmann, Univ. Grenoble Alpes and CNRS, Grenoble, France

² Univ. Bordeaux IMB, INRIA Bordeaux-Sud-Ouest Team MEMPHIS, CNRS UMR 5251

Corresponding author: meriem.jedouaa@imag.fr

Abstract: We propose a numerical method to capture an arbitrary number of fluid/solid interfaces in a level-set framework, following the ideas introduced for image segmentation. Using only three label maps and two distance functions it is possible to get the distance between the closest cells and to apply the collision force whatever the number of cells is. Consequently, the method is very efficient when dealing with a large number of cells. Numerical simulations are performed in two- and three-dimensions under gravity force.

Keywords: Level set method, Penalization method, Fluid/Structure interaction, Collision model.

1 Introduction

Numerical simulations of fluid/structure interaction (FSI) have attracted an increasing interest and several methods have been proposed during the last decades. A popular method is the Arbitrary Lagrangian Eulerian approach (ALE) introduced by Donea in 1982 (see [1]). 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. Another method, introduced by Cottet and Maitre in [2], is to use a purely Eulerian formulation for describing the fluid/structure interaction. Within this framework, a level set method is used to capture the interface.

Among the various problems that address the fluid/structure interaction, the simulation of dense suspensions is one of the most challenging one. In this work, we aim at dealing with two major difficulties that arises when dealing with dense suspensions: the high computational complexity due to a large number of cells and the numerical contacts. In [3], M. Hillairet proved that the hydrodynamical forces between two bodies following a Navier-Stokes flow prevent contact at finite time. Numerically, however, it is necessary to have enough discretization points between two interfaces in order to resolve these hydrodynamical forces. This lack of discretization points could lead to numerical contacts and coalescence of bodies. In order to resolve accurately these lubrication forces, one can refine the mesh near the inter particle gap. However, this strategy leads to a high computational cost as several refinements are necessary. Consequently, a collision model appears to be necessary to develop numerical simulations with tractable cost at relatively low resolution.

In this work, we present a new type of algorithm to enable these contacts efficiently by adding a short range repulsive force. This algorithm is derived from the multi geometric deformable model (MGDM) introduced by J. Bogovic [4] 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 [5]. The main advantages of this method is that it requires only five fields and one level set function to capture an arbitrary number of cells and it can, at the same time, prevent numerical contacts. Consequently, this approach provides a huge computational saving, as will be illustrated below. The level set function captures all interfaces and is transported with the fluid velocity. Then a local fast marching

algorithm is performed at each time step to find the the closest neighbours and their associated distance functions. In the case of spherical rigid structures it is possible to avoid it by advecting the center of each sphere and so another faster approach is employed (see [6]).

2 Description of the model

In this section, we present the model that is used to simulate a dense suspension of cells immersed in a fluid. This approach is an adaptation of the multi geometric deformable model (MGDM) [4] that was introduced by J.Bogovic for image analysis.

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

$$\left\{ \begin{array}{l} \forall i \neq j, \Omega_i \cap \Omega_j = \emptyset \\ \Omega_{N_f} = \overline{\Omega \setminus \bigcup_{i=1}^N \Omega_i} \\ \Gamma_{N_f} = \bigcup_{i=1}^N \Gamma_i. \end{array} \right. \quad (1)$$

In order to locate the different objects in the domain, we introduce a set of label maps and distance functions.

2.1 Label maps

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

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

where $d(x, \Gamma) = \min_{y \in \Gamma} \|x - y\|$. The label map L_0 provides a partition of the whole computational domain Ω into $N + 1$ different objects. L_1 identifies the index 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 .

As a consequence, $L_2(x)$ gives the index of the first closest structure for any x in the whole computational domain. Figure 1 shows an illustration of the three label maps in the case of five structures immersed in a fluid. We can see that for all points belonging to the surrounding four cells (green, yellow, purple and light blue objects), the closest structure is always the blue one. On the contrary, the label map L_2 has partitioned the blue object into four regions, each of them giving the color of the closest structure. Therefore, the three label maps provide an interesting local description of the entire fluid/structure domain Ω . This local representation of objects has to be completed with the two distance functions associated to L_1 and L_2 .

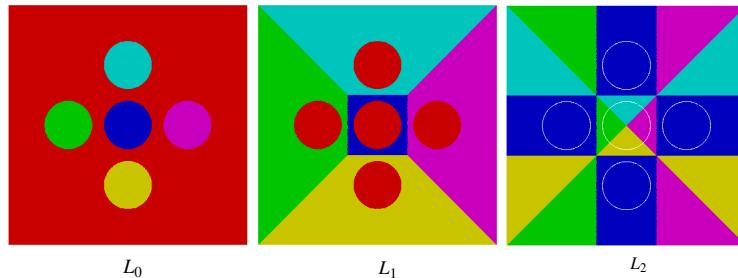


Figure 1: Illustration of the three label maps for a configuration of five cells. The white contour represents the boundary of the cells.

2.2 Distance functions

We define two distance functions φ_1 and φ_2 as:

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

At any point of the domain Ω , the distance function φ_1 captures the union of all cells interfaces and φ_2 gives the distance to the first closest cell. As a consequence, on each point of a cell, we have the distance to the closest one allowing to define a collision model to the closest interface. For a configuration of five bodies the Figure 2 shows an example of the two distance functions related to the Figure 1.

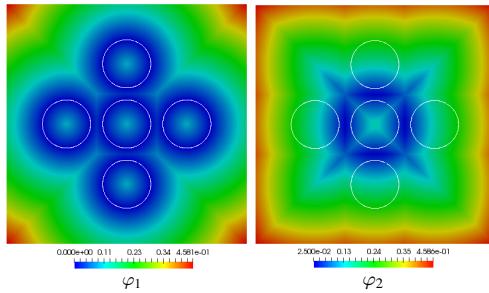


Figure 2: Illustration of the two distance functions for a configuration of five cells. The white contour represents the boundary of the cells.

2.3 Evolution of the label maps and distance functions

The evolution of the label maps and distance functions is based on the transport of one level set function ϕ by the fluid velocity, this one captures the union of all interfaces. Then a local fast-marching method is performed to update the label and distance functions.

Redefinition of ϕ_1 and L_0

The distance function φ_1 is directly given by the absolute value of ϕ . In order to evolve the label function L_0 , we use the level set function ϕ . At each time step t , we change the label value L_0^t , near the interface. Namely, if the level set function ϕ^t is positive, we set L_0^t to the label of the fluid ($L_0^t = N_f$). Then, at each point x where the level set function is negative and the label $L_0^{t-1}(x)$ is still the label of the fluid, we assign to $L_0^t(x)$ the value of its neighbours which are different from N_f .

Multiple label fast marching method

To evolve the functions φ_1 , φ_2 and the label maps L_1 and L_2 , we perform a multiple label fast marching procedure. This local fast marching is an extension of the fast marching method [7] that was introduced in [4].

To achieve this, we solve the following eikonal equation in the entire computational domain Ω :

$$|\nabla d| = 1$$

At initialization, the function d is equal to φ_1 on the interfaces. At each point x of the domain, an integer $\text{lab}(x)$ provides the number of the interface that spreads and the distance value $d(x)$ gives the distance from x to the interface $\Gamma_{\text{lab}(x)}$.

There are three sets of points: alive(A), narrow-band (NB) and far away (F). At initialization, the narrow-band contains the closest points to the interfaces. The algorithm computes the new values only at the nodes

belonging to the narrow-band and accepts just one of them, the one corresponding to the minimum value. If the point x_m has not been visited yet, this minimum value corresponds to the distance $\varphi_1(x_m)$ and the label associated is $L_1(x_m)$. Thus, this point has to be suppressed from the narrow-band. The second boundary that reaches x_m gives us $\varphi_2(x_m)$ and $L_2(x_m)$. The present algorithm stops propagation when the label function L_2 (and so the distance function φ_2) is defined for all points.

3 Collision model

In this work, we propose a collision model inspired by [5], that consists in a short range repulsive force which takes into account the interactions between the closest cells. To achieve this, we propose a collision model that only involves the two distance functions φ_1 and φ_2 as:

$$\forall x \in \Omega, F_{\text{col}}^{\text{lab}}(x) = \frac{k}{\varepsilon} \zeta \left(\frac{\varphi_1(x)}{\varepsilon} \right) \frac{\nabla \varphi_2(x)}{\varphi_2(x)} \exp \left(-\frac{\varphi_2(x)}{\varepsilon_b} \right) \quad (2)$$

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

We denote by ζ a one-dimensional cut off function that allows to localize the interfaces. The coefficient ε_b represents the rebound coefficient, in practice we set $\varepsilon_b = \varepsilon$.

As φ_2 is the distance between the closest cells at all points of the fluid/structures domain Ω , the direction of the force is given by $\nabla \varphi_2$. Moreover, if a cell is surrounded by other cells the interaction of the other cells are taken into account on different part of its interface. For instance, on Figure 1 the label L_2 indicates that the repulsive forces acting on the four surrounding cells comes from the blue body. On the contrary, the repulsive force acting on the blue cell is on one part of the interface coming from the light blue cell (top), on the left region coming from the green cell (bottom), on the right region coming from the purple cell and so forth.

This interaction force tends to zero out of a cut-off radius reducing the number of interacting neighbours. In the model introduced in [5] ,for N bodies captured by N level set functions, N^2 computations of the repulsive forces are required, which represents a huge computational effort. The advantage of our formulation is that there is only one repulsive force for an arbitrary number of structures, this leads to a considerable saving of the computational cost.

4 Eulerian model for fluid/rigid body coupling

We consider N cells 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 cell $\Omega_i, 1 \leq i \leq N$ represents a rigid body, ρ_f and μ denote the constant density and the viscosity of the fluid, U and p denote the flow velocity and the pressure.

The flow is governed by the viscous incompressible Navier-Stokes equations.

In order to get rid of the no-slip boundary conditions on the solid boundaries, we use a penalization method([8, 9, 5]). 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. We denote by F_{wall} the repulsive forces exerted by the walls.

The proposed penalization model

Using our method, we can define the following penalized model

$$\begin{cases} \rho(\partial_t U + (U \cdot \nabla) U) - \nabla \cdot (\mu \nabla U) + \nabla p = \rho g + \lambda(\chi_{L_0^\varepsilon}(u_{L_0^\varepsilon} - U)) + F_{\text{col}}^{\text{lab}} + F_{\text{wall}} & \text{in } \Omega_T \\ \nabla \cdot U = 0 & \text{in } \Omega_T \\ \partial_t \phi + u \cdot \nabla \phi = 0 & \text{in } \Omega_T \end{cases} \quad (3)$$

where $\lambda \gg 1$ is the penalization parameter, numerically λ is fixed to 10^{10} . The function $\chi_{L_0^\varepsilon}$ is the characteristic function of the region defined by the label map L_0^ε which is an extension to the distance ε of the label L_0 :

$$\forall x \in \Omega, \quad L_0^\varepsilon(x) = \begin{cases} L_0(x) & \text{if } (L_0(x) \neq N_f) \\ L_1(x) & \text{if } ((L_0(x) = N_f) \text{ and } (\phi \leq \varepsilon)) \\ N_f & \text{otherwise} \end{cases} \quad (4)$$

We can define a characteristic functions of the solid bodies:

$$\forall x \in \Omega, \forall y \in \Omega, \quad \chi_{L_0^\varepsilon(x)}(y) = 1 - H(\phi_{L_0^\varepsilon(x)(y)})$$

The level set functions ϕ_i are reconstituted as:

$$\forall x \in \Omega, \forall i \in \{1, \dots, N\}, \quad \phi_i(x) = \begin{cases} \phi(x) & \text{if } ((L_0(x) = i) \text{ or } (L_1(x) = i)) \\ \phi_2(x) & \text{otherwise} \end{cases}$$

For each $x \in \Omega$, $y \rightarrow u_{L_0^\varepsilon(x)}(y)$ is the rigid velocity of the solid body $\Omega_{L_0^\varepsilon(x)}$ obtained by averaging the translation and angular velocities over the solid. Denoting by $\rho_{L_0^\varepsilon(x)}$ the density of the body $\Omega_{L_0^\varepsilon(x)}$ we obtain the following density function:

$$\rho_x = \rho_f(1 - \chi_{L_0^\varepsilon(x)}) + \chi_{L_0^\varepsilon(x)}\rho_{L_0^\varepsilon(x)}$$

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. This is a very desirable model to simulate a large number of interacting cells.

5 Numerical illustrations

This section is devoted to the numerical results obtained with the proposed model. In [6], the numerical validation of the model is presented, a qualitative grid convergence in two and three dimensions is achieved and a comparison of our model to those available in the literature is presented. For all of the simulations presented in this paper, the computational domain Ω is a square of size $[0, 1]^2$ or a cube of size $[0, 1]^3$. 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 $\rho_s = 2$.

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.

5.1 Comparison of the method with multiple level sets decomposition

We give here a comparison of our penalization model (3) and the penalization model that uses multiple level sets decomposition in (see[5] and [6]).

The averaged CPU time of our algorithm is compared to the method using N level set functions , according to the number of cells. As noticed before, the collision model introduced in [5] computes N^2 repulsive forces which induced a high computational cost. The CPU time of the penalization model in [5] is larger because it depends on the number of cells. Indeed, N rigid velocities must be computed to get the right velocity of each cell. Finally, we can see that our method substantially reduces the computational time.

To highlight the difference between our collision model and the one introduced in [5] , we focus on a test case with six circular rigid bodies falling on each other. First, in the case of six rigid disk of radius $R = 0.1$ (see Figure (4)), and then for the same configuration with six smaller disks with radius $R = 0.03$ (see Figure (5)). The black line stands for the collision model in[5] and the white line for the collision model (2). In Figure (4), 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. The second test case shows a higher difference in the dynamics even if the final state is the same. In fact, the difference between

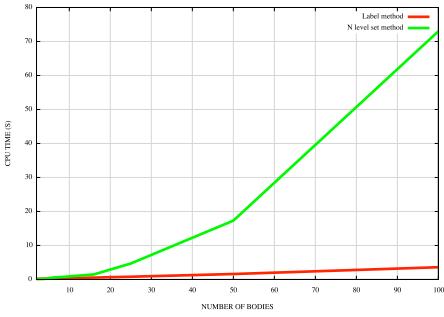


Figure 3: Comparison of the average CPU time

the two models is stronger when the number of body is larger or when the force coefficients are higher.

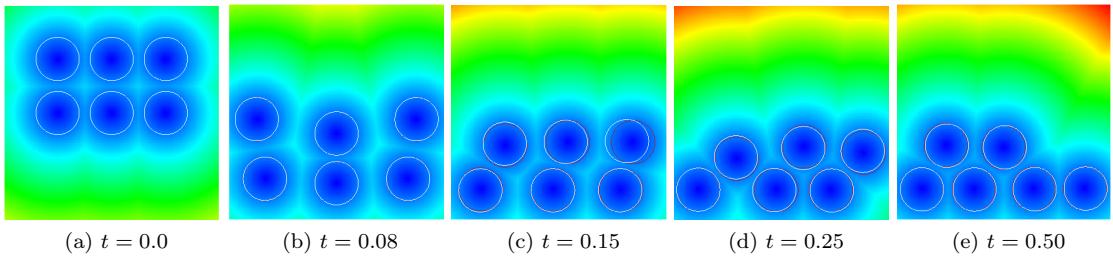


Figure 4: Comparison of the two collision models for six disks of radius $R = 0.1$. The background colors show the level set amplitude.

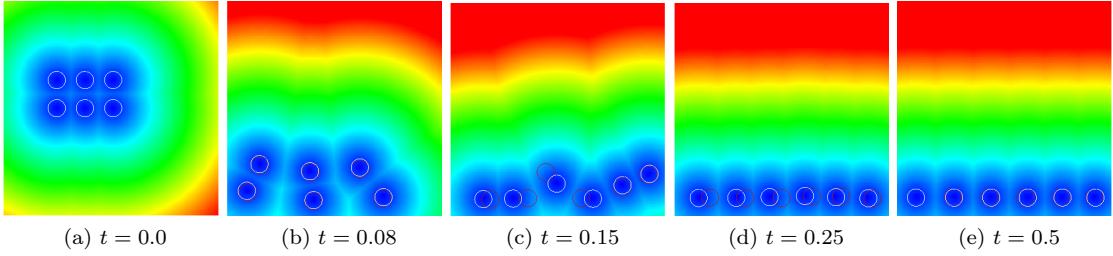


Figure 5: Comparison of the two collision models for six disks of radius $R = 0.03$. The background colors show the level set amplitude.

5.2 Dense suspensions of rigid bodies in 2D and 3D

In this part, we present some results of dense suspensions of rigid bodies subject to gravity which was performed using our numerical model. 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 repulsive coefficients are:

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

The first simulations deal with the sedimentation of 400 rigid bodies of radius $R = 0.01$ in the two dimensional case. The 400 bodies fall down symmetrically to reach a dense repartition at the bottom as can be seen in the Figure 6.

The second simulations deal with the sedimentation of 30 rigid bodies of various radii. The results are represented in Figure 7, the colors represent the different values of the label map L_0 . It appears that the repulsive forces are well taken into account even if the difference between the size of the bodies is important. Indeed, there is no merging of small and big bodies although, due to the distance function φ_2 , the force of a big cell on a small one is effective on the whole boundary ring of the small cell whereas the force of a small cell on a big one is effective only on a part of the boundary ring.

Finally, we address the 3D case. Figure 8 shows the simulation of 500 rigid spheres of radius $R = 0.01$ falling under gravity for two different grids of size 64^3 and 128^3 . The half thickness of the interface is $\varepsilon = 2\Delta x$. The coefficient of gravity g is set to -980. The repulsive coefficients are: $k_x = -g/10$, $k_y = -g/10$, $k_z = -g/10$, $k_x^{\text{wall}} = -g/40$, $k_y^{\text{wall}} = -g/40$, $k_z^{\text{wall}} = -g$.

At initial step, there are five slices of 100 bodies at a distance $d = 0.1$ (distance of two closest bodies' centers). The interactions between bodies occur at once after $t = 1.5$ and the equilibrium state is reached around $t = 2.5$.

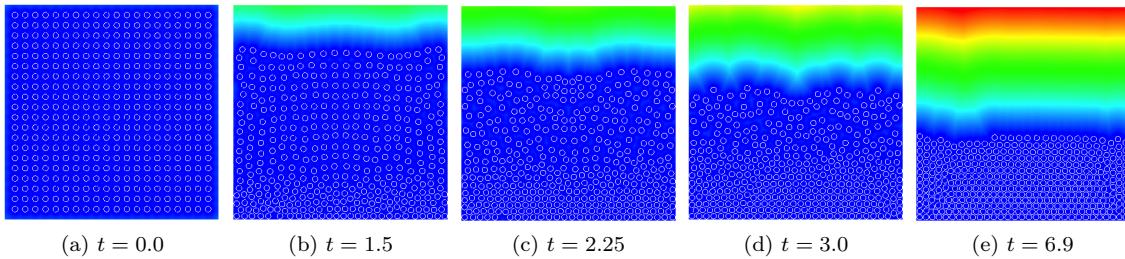


Figure 6: Simulation of 400 rigid disks submitted to gravity (the white line corresponds to the level line $\phi = \varepsilon$). The background colors show the level set amplitude.

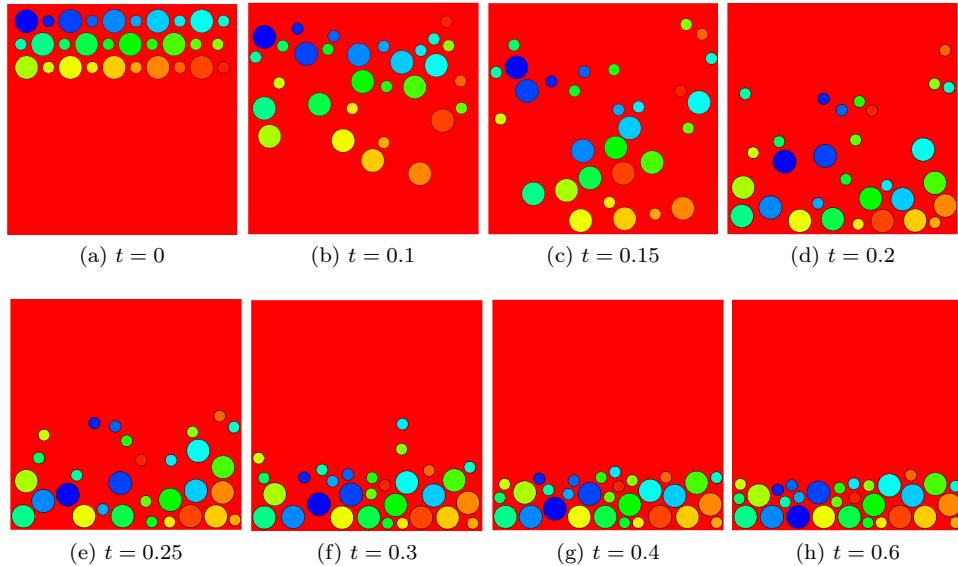


Figure 7: Simulation of 30 rigid bodies of different radii ($R = 0.05$ or $R = 0.025$) falling under gravity. The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 30th body and red for the fluid that is the 31th object.

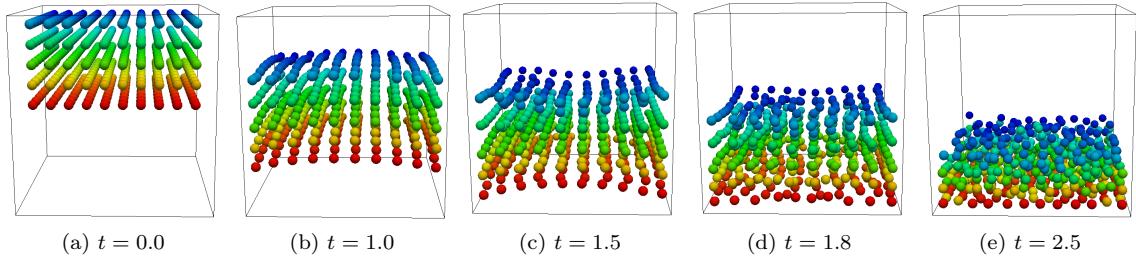


Figure 8: Simulation of 500 rigid spheres subject to gravity (grid resolution size 128^3). The colors indicate the values of the label map L_0 from dark blue for the first body to dark orange for the 500th body and red for the fluid that is the 501th object.

6 Conclusions and Future Work

In this work, we introduced a new model to simulate efficiently a large number of interacting cells 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 cells is proposed. This model which is totally independent on the number of bodies, is compared both theoretically and numerically to the model introduced in [5]. 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 cells, our model substantially reduces the CPU time. In future work, this model will be applied to elastic bodies.

References

- [1] J Donea, S Giuliani, and JP Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer methods in applied mechanics and engineering*, 33(1):689–723, 1982.
- [2] G.-H Cottet and E. Maitre. A level set method for fluid-structure interactions with immersed surfaces. *Mathematical models and methods in applied sciences*, 16(03):415–438, 2006.
- [3] M. Hillairet. Lack of collision between solid bodies in a 2d incompressible viscous flow. *Communications in Partial Differential Equations*, 32(9):1345–1371, 2007.
- [4] J. A Bogovic, J. L Prince, and P-L Bazin. A multiple object geometric deformable model for image segmentation. *Computer Vision and Image Understanding*, 117(2):145–157, 2013.
- [5] M. Coquerelle and G.-H. Cottet. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *Journal of Computational Physics*, 227(21):9121–9137, 2008.
- [6] M. Jedouaa, C.-H Bruneau, and E. Maitre. An efficient interface capturing method for a large collection of interacting cells immersed in a fluid. *submitted*, 2015.
- [7] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.
- [8] P. Angot, C.-H Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81(4):497–520, 1999.
- [9] NA Patankar. A formulation for fast computations of rigid particulate flows. *Center for Turbulence Research Annual Research Briefs*, 2001:185–196, 2001.