FLUID-PARTICLE SIMULATIONS WITH FREEFEM++

Aline Lefebvre[1]

Abstract. We present here a method to simulate the motion of rigid particles in a Newtonian fluid. The rigid motion is enforced by penalizing the strain tensor on the rigid domain and the time discretization is performed using the method of characteristics. This leads to a generalized Stokes variational formulation on the whole domain which can easily be implemented from any Finite Element Stokes/Navier-Stokes solver. In order to ensure robustness, we describe a strategy to take collisions into account. To validate the method, simulations implemented with FreeFem++ are presented.

Résumé. Nous présentons ici une méthode pour simuler des écoulements de particules rigides dans un fluide Newtonien. Le mouvement rigide est imposé grâce à une pénalisation du tenseur des déformations sur le domaine rigide et la discrétisation en temps est effectuée en utilisant la méthode des caractéristiques. Cela nous permet d'obtenir une formulation variationnelle sur le domaine tout entier de type Stokes généralisé qui peut être programmée facilement à partir de tout solveur éléments finis pour Stokes/Navier-Stokes. Afin d'assurer la robustesse du code, on décrit une stratégie pour prendre en compte les collisions. Pour valider la méthode, on présente des simulations implémentées sous FreeFem++.

Introduction

In this article, we consider rigid particles embedded in a Newtonian fluid. Our goal is to describe a method for their direct numerical simulation that can be straightforwardly implemented on a general Finite Element solver like FreeFem++ (see [1]) which we use to make numerical experiments. Numerical simulation of particulate flows raises two main issues. The first one is to impose the rigid motion of the particles and to deal with the fact that the domain filled with the fluid varies in time. The second one relates to the collisions between particles.

The methods to handle the rigid motion of the particles can be devided in two classes. The first one relies on a moving mesh following the fluid domain (see [7, 8, 10, 11]). The second approach is the fictitious domain methods also called domain embedding methods: the idea is to extend a problem defined on a time-dependent, complex domain (the fluid domain) to a larger one (fixed) called the fictitious domain. Most of these methods involve a cartesian mesh covering the whole domain (fluid and particles) on which local meshes are moving (following the particles) and the rigid motion is enforced using a Lagrange multiplier. For more details, see [5] where the velocity is constrained to be a rigid motion and [14] where the strain tensor is constrained to be zero. Penalty methods are another class of fictitious domain strategies: in [2] the authors take obstacles into account and, in [16], particulate flows are simulated using an augmented Lagrangian approach. We propose here to enforce the rigid motion penalizing the strain tensor. This method, associated to the method of characteristics for the time discretization leads to a generalized Stokes variational formulation on the whole domain. Therefore, it can be easily implemented from any finite element Stokes/Navier-Stokes solver. See [9] for an exemple of a rigid body attached at one of his points.

[1] Laboratoire de Mathématiques, Université Paris-Sud, Orsay, France; e-mail: `Aline.Lefebvre@math.u-psud.fr`

Collisions also present severe difficulties in direct numerical simulations and an efficient strategy has to be implemented to deal with this problem. Note that, "collision" is a somewhat unsuited term since, in the case of spherical particles in a Newtonian fluid, lubrication forces do not allow particles to get into contact in finite time (see [6]). However, these particles can approach arbitrarily close and, because of the discretization, numerical collisions may occur. The first idea is to find a strategy that makes it possible to approximate with accuracy the lubrication forces: in [8], a method based on local mesh refinements is implemented and gives good results. However, the number of refinements and the smallness of the time steps that are necessary to avoid collisions are not known a priori and the method can therefore present a heavy computational cost. Therefore, less time consuming strategies have been developed to avoid collisions. Some of them consist in adding a short range repulsive force (see [5,14]). In [11] a minimization algorithm is used to impose a minimal distance between the particles. In [17], particles are allowed to slightly overlap each other and an elastic repulsive force is applied when such overlapping occurs. Those methods ensure numerical stability but do not respect the physics. Another approach is to implement a collision strategy based on inelastic collisions. This idea has been used in [10] to impose a minimal distance between particles but the method considers separately each couple of neighbouring particles and therefore can not be used for a large number of particles. We propose here to implement the scheme described in [12] for inelastic collisions. It allows us to impose a minimal distance between particles and, since it globally handles all the possible contacts, we can consider mixtures with many particles.

1. Continuous Problem

1.1. Modelling particulate flows

We consider (see Fig. 1) $\Omega \subset \mathbb{R}^2$ a connected, bounded, regular domain. We denote by $(B_i)_{i=1\ldots N}$ N rigid inclusions (particles) in Ω (subsets of Ω, disjoint and strongly contained in Ω) and by B the whole rigid domain: $B = \cup_{i=1}^{N} B_i$. We suppose that $\Omega \setminus \bar{B}$ is filled with a Newtonian fluid governed by the Navier-Stokes equations. For the sake of simplicity, we will consider spherical particles with homogeneous Dirichlet boundary conditions on $\partial\Omega$.
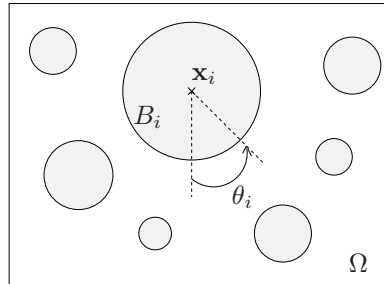

Figure 1. Notations

We respectively denote by $\mathbf{f}_f$ and $\mathbf{f}_i$ the external forces exerted on the fluid and the i-th particle and by ρ_f and ρ_i their respective densities. μ is the viscosity of the fluid. The mass of the i-th particle is denoted by m_i. The position of its center of mass and its angular orientation are respectively denoted by $\mathbf{x}_i$ and θ_i. Finally, $\mathbf{V}_i = \dot{\mathbf{x}}_i$ and $\omega_i = \dot{\theta}_i$ are its translational and angular velocities and J_i is the kinematic momentum about its center of mass: $J_i = \int_{B_i} \rho_i |\mathbf{x} - \mathbf{x}_i|^2$. We will also use the following classical notations:

$$\sigma = 2\mu \mathbf{D}(\mathbf{u}) - p\mathrm{Id}, \quad \mathbf{D}(\mathbf{u}) = \frac{\nabla\mathbf{u} + (\nabla\mathbf{u})^T}{2} \quad \text{and} \quad \frac{D\mathbf{u}}{Dt} = \frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}.$$

where σ is the Cauchy stress tensor and $D\mathbf{u}/Dt$ is the total derivative of $\mathbf{u}$. Finally, $\mathbf{x}^\perp$ denotes $(-x_2, x_1)$ and $\mathbf{n}$ is the external normal to $\Omega \setminus \bar{B}$.

We have to find the velocity $\mathbf{u} = (u_1, u_2)$ and the pressure field p defined in $\Omega \setminus \bar{B}$ as well as the velocities of the particles $\mathbf{V} \in \mathbb{R}^{2N}$ and $\omega \in \mathbb{R}^N$. At each instant of time t, the fluid obeys the Navier-Stokes equations in $\Omega \setminus \bar{B} = \Omega \setminus \bar{B}(t)$ with homogeneous Dirichlet boundary conditions:

$$\begin{cases} \rho_f \dfrac{D\mathbf{u}}{Dt} - \mu \triangle \mathbf{u} + \nabla \mathrm{p} & = & \mathbf{f}_f & \text{in } \Omega \setminus \bar{B} \\ \nabla \cdot \mathbf{u} & = & 0 & \text{in } \Omega \setminus \bar{B} \\ \mathbf{u} & = & 0 & \text{on } \partial\Omega. \end{cases} \tag{1}$$

The viscosity imposes a no-slip boundary condition on ∂B:

$$\mathbf{u} = \mathbf{V}_i + \omega_i(\mathbf{x} - \mathbf{x}_i)^\perp \text{ on } \partial B_i \ \forall i. \tag{2}$$

Finally, since the fluid exerts hydrodynamic forces on the particles, Newton's second law couples these equations:

$$\begin{cases} m_i \dfrac{d\mathbf{V}_i}{dt} & = & \int_{B_i} \mathbf{f}_i - \int_{\partial B_i} \sigma \mathbf{n} \, ds & \forall i \\ J_i \dfrac{d\omega_i}{dt} & = & \int_{B_i} (\mathbf{x} - \mathbf{x}_i)^\perp \cdot \mathbf{f}_i - \int_{\partial B_i} (\mathbf{x} - \mathbf{x}_i)^\perp \cdot \sigma \mathbf{n} \, ds & \forall i. \end{cases} \tag{3}$$

1.2. **The corresponding variational formulation over a constrained domain**

In order to avoid remeshing, we look for a weak formulation involving functions defined on the whole domain Ω. As it is shown in [9], this can be done by introducing, at each time step, rigid constraints into the functionnal spaces considered:

$$\begin{aligned} K_\nabla & = & \left\{ \mathbf{u} \in H_0^1(\Omega), \nabla \cdot \mathbf{u} = 0 \right\}, \\ K_B & = & \{ \mathbf{u} \in H_0^1(\Omega), \forall i \ \exists (\mathbf{V}_i, \omega_i) \in \mathbb{R}^2 \times \mathbb{R}; \\ & & \mathbf{u} = \mathbf{V}_i + \omega_i(\mathbf{x} - \mathbf{x}_i)^\perp \text{ a.e. in } B_i \} \\ & = & \{ \mathbf{u} \in H_0^1(\Omega), \mathbf{D}(\mathbf{u}) = 0 \text{ a.e. in } B \}. \end{aligned}$$

K_∇ is the space of divergence free functions on Ω and K_B is the space of functions on Ω that do not deform B. Note that, since K_B depends on B, it may vary on time. At each instant of time $t > 0$, the solution to (1),(2),(3) is extended on the whole domain Ω by a function in K_B:

$$\mathbf{u}(t, \mathbf{x}) = \mathbf{V}_i(t) + \omega(t)(\mathbf{x} - \mathbf{x}_i(t))^\perp \text{ in } B_i(t) \ \forall i$$

and we still denote by $\mathbf{u}$ this extension. Let $(\mathbf{u}, \mathrm{p})$ be an extended solution of the problem for a certain time $t > 0$ and choose $\tilde{\mathbf{u}} \in K_B$. By multiplying the Navier-Stokes equation by $\tilde{\mathbf{u}}$, and integrate it by parts over $\Omega \setminus \bar{B}$, we obtain:

$$\int_{\Omega \setminus \bar{B}} \rho_f \frac{D\mathbf{u}}{Dt} \cdot \tilde{\mathbf{u}} + 2\mu \int_{\Omega \setminus \bar{B}} \mathbf{D}(\mathbf{u}) : \mathbf{D}(\tilde{\mathbf{u}}) - \int_{\Omega \setminus \bar{B}} \mathrm{p} \nabla \cdot \tilde{\mathbf{u}} - \int_{\partial(\Omega \setminus \bar{B})} \sigma \mathbf{n} \cdot \tilde{\mathbf{u}} = \int_{\Omega \setminus \bar{B}} \mathbf{f}_f \cdot \tilde{\mathbf{u}}.$$

The intergrals over $\Omega \setminus \bar{B}$ can be extended over Ω using the fact that $\mathbf{D}(\tilde{\mathbf{u}}) = 0$ and $\nabla \cdot \tilde{\mathbf{u}} = 0$ in B. Moreover, since $\tilde{\mathbf{u}}$ is in K_B, we have:

$$\forall i, \quad \exists \tilde{\mathbf{V}}_i, \tilde{\omega}_i \text{ such that } \tilde{\mathbf{u}}(\mathbf{x}) = \tilde{\mathbf{V}}_i + \tilde{\omega}_i(\mathbf{x} - \mathbf{x}_i)^\perp \text{ in } B_i.$$

The relation above and the boundary condition on ∂B allow us to cancel the hydrodynamic forces:

$$\int_{\Omega\setminus\bar{B}} \rho_f \frac{\mathbf{Du}}{\mathbf{D}t} \cdot \tilde{\mathbf{u}} + \sum_{i=1}^{N} m_i \frac{d\mathbf{V}_i}{dt}\tilde{\mathbf{V}}_i + \sum_{i=1}^{N} \mathrm{J}_i \frac{d\omega_i}{dt}\tilde{\omega}_i + 2\mu \int_{\Omega} \mathbf{D}(\mathbf{u}) : \mathbf{D}(\tilde{\mathbf{u}}) - \int_{\Omega} \mathrm{p}\nabla\cdot\tilde{\mathbf{u}} = \int_{\Omega} \mathbf{f}\cdot\tilde{\mathbf{u}}$$

where $\mathbf{f} = \mathbf{f}_f\mathbf{1}_{\Omega\setminus\bar{B}} + \sum_{i=1}^{N} \mathbf{f}_i\mathbf{1}_{B_i}$. Finally, $(\mathbf{V}_i, \omega_i)$ can be considered as auxiliary variables since using

$$\forall i, \quad m_i \frac{d\mathbf{V}_i}{dt}\tilde{\mathbf{V}}_i + \mathrm{J}_i\frac{d\omega_i}{dt}\tilde{\omega}_i = \int_{B_i} \rho_i \frac{\mathbf{Du}}{\mathbf{D}t} \cdot \tilde{\mathbf{u}},$$

we obtain the variational formulation:

$$\begin{cases} \displaystyle\int_{\Omega} \rho\frac{\mathbf{Du}}{\mathbf{D}t} \cdot \tilde{\mathbf{u}} + 2\mu \int_{\Omega} \mathbf{D}(\mathbf{u}) : \mathbf{D}(\tilde{\mathbf{u}}) - \int_{\Omega} \mathrm{p}\nabla\cdot\tilde{\mathbf{u}} = \int_{\Omega} \mathbf{f}\cdot\tilde{\mathbf{u}}, \quad \forall\tilde{\mathbf{u}}\in K_B, \\ \displaystyle\int_{\Omega} \mathrm{q}\nabla\cdot\mathbf{u} = 0, \quad \forall\mathrm{q}\in L^2(\Omega), \end{cases} \tag{4}$$

where $\rho = \rho_f\mathbf{1}_{\Omega\setminus\bar{B}} + \sum_{i=1}^{N} \rho_i\mathbf{1}_{B_i}$.

2. Numerical Strategy: Towards a Classical Weak Formulation

2.1. Time discretization

We denote by $\Delta t > 0$ the time step and for any function f we define $f^n(\mathbf{x}) = f(\mathbf{x}, t_n)$ where $t_n = n\Delta t$. The time discretization is performed using the method of characteristics in order to obtain a generalized Stokes variational formulation. If we define the characteristic trajectory that is passing through $\mathbf{x}$ at time t as

$$\begin{cases} \dfrac{\partial\mathbf{X}}{\partial\tau}(\mathbf{x}, t, \tau) = \mathbf{u}(\mathbf{X}(\mathbf{x}, t, \tau), \tau), \\ \mathbf{X}(\mathbf{x}, t, t) = \mathbf{x}, \end{cases}$$

it is clear that, for any function $\Phi(t, \mathbf{x})$ we have

$$\frac{D\Phi}{Dt}(\mathbf{x}, t) = \left(\frac{\partial\Phi}{\partial t} + \mathbf{u}\cdot\nabla\Phi\right)(\mathbf{x}, t) = \frac{\partial}{\partial t}\left(\Phi(\mathbf{X}(\mathbf{x}, t, \tau), \tau)\right)|_{\tau=t}.$$

Therefore, using the fact that $\mathbf{X}(\mathbf{x}, t_{n+1}, t_{n+1}) = \mathbf{x}$, we can discretize the total derivative of Φ writing:

$$\left(\frac{D\Phi}{Dt}\right)^{n+1}(\mathbf{x}) \approx \frac{\Phi^{n+1}(\mathbf{x}) - \Phi^n(\mathbf{X}^n(\mathbf{x}))}{\Delta t}$$

where $\mathbf{X}^n(\mathbf{x})$ is an approximation of $\mathbf{X}(\mathbf{x}, t_{n+1}, t_n)$. For further details about this dicretization see [15].

To apply this method to our problem, we note that since ρ is constant along the characteristics, we have $\rho\dfrac{\mathbf{Du}}{\mathbf{D}t} = \dfrac{\mathbf{D}(\rho\mathbf{u})}{\mathbf{D}t}$, and we obtain the following discretized scheme: for each $n > 0$,

(i) compute ρ^{n+1} from $\mathbf{u}^n$ and $(B_i^n)_i$:

$$\forall i, \quad \mathbf{V}_i^n = \frac{1}{\pi r_i^2} \int_{B_i^n} \mathbf{u}^n, \quad \mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t \mathbf{V}_i^n \text{ and } \rho^{n+1} = \rho_f \mathbf{1}_{\Omega \setminus \bar{B}^{n+1}} + \sum_{i=1}^{N} \rho_i \mathbf{1}_{B_i^{n+1}} \tag{5}$$

(ii) solve the following discretized weak formulation for (4):

$$\begin{cases} \text{Find } \mathbf{u}^{n+1} \in K_{B^{n+1}} \text{ and } \mathrm{p}^{n+1} \in L^2(\Omega) \text{ such that:} \\[2mm] \frac{1}{\Delta t} \int_{\Omega} \rho^{n+1} \mathbf{u}^{n+1} \cdot \tilde{\mathbf{u}} + 2\mu \int_{\Omega} \mathbf{D}(\mathbf{u}^{n+1}) : \mathbf{D}(\tilde{\mathbf{u}}) - \int_{\Omega} \mathrm{p}^{n+1} \nabla \cdot \tilde{\mathbf{u}} \\[3mm] \qquad\qquad = \frac{1}{\Delta t} \int_{\Omega} (\rho^n \mathbf{u}^n) \circ \mathbf{X}^n \cdot \tilde{\mathbf{u}} + \int_{\Omega} \mathbf{f}^{n+1} \cdot \tilde{\mathbf{u}}, \quad \forall \tilde{\mathbf{u}} \in K_{B^{n+1}}, \\[3mm] \int_{\Omega} \mathrm{q} \nabla \cdot \mathbf{u}^{n+1} = 0, \quad \forall \mathrm{q} \in L^2(\Omega), \end{cases} \tag{6}$$

where $\mathbf{X}^n$ in step (ii) is computed using the characteristics associated to $\mathbf{u}^n$. Note that, in order not to deform the rigid domain, we use the real degrees of freedom of the particules to convect ρ in step (i).

2.2. A penalty method to enforce the rigid motion

Our aim is to obtain a variational formulation adapted to a finite element discretization in space. In order to do so, we use a penalty method to take the constraint into account. This method is presented in [9] and consists in considering the minimization problem over a constrained domain associated to (6) and relaxing the constraint by introducing a penalty term in the minimized functional. The added term is the following:

$$\frac{1}{\varepsilon} \int_{B^{n+1}} \mathbf{D}(\mathbf{u}^{n+1}) : \mathbf{D}(\mathbf{u}^{n+1}),$$

so that $\mathbf{D}(\mathbf{u}^{n+1})|_{B^{n+1}}$ goes to zero when ε goes to zero and $\mathbf{u}^{n+1}$ tends to be a rigid motion in B^{n+1}.

The variational formulation obtained is:

$$\begin{cases} \text{Find } \mathbf{u}^{n+1} \in H_0^1(\Omega) \text{ and } \mathrm{p}^{n+1} \in L^2(\Omega) \text{ such that:} \\[2mm] \frac{1}{\Delta t} \int_{\Omega} \rho^{n+1} \mathbf{u}^{n+1} \cdot \tilde{\mathbf{u}} + 2\mu \int_{\Omega} \mathbf{D}(\mathbf{u}^{n+1}) : \mathbf{D}(\tilde{\mathbf{u}}) + \frac{2}{\varepsilon} \int_{B^{n+1}} \mathbf{D}(\mathbf{u}^{n+1}) : \mathbf{D}(\tilde{\mathbf{u}}) - \int_{\Omega} \mathrm{p}^{n+1} \nabla \cdot \tilde{\mathbf{u}} \\[3mm] \qquad\qquad = \frac{1}{\Delta t} \int_{\Omega} (\rho^n \mathbf{u}^n) \circ \mathbf{X}^n \cdot \tilde{\mathbf{u}} + \int_{\Omega} \mathbf{f}^{n+1} \cdot \tilde{\mathbf{u}}, \quad \forall \tilde{\mathbf{u}} \in H_0^1(\Omega), \\[3mm] \int_{\Omega} \mathrm{q} \nabla \cdot \mathbf{u}^{n+1} = 0, \quad \forall \mathrm{q} \in L^2(\Omega). \end{cases} \tag{7}$$

We replace in the previous algorithm (6) by (7) which can be solved using any Stokes finite element solver. Note that, similarly to what is done in [16], the variational formulation (7) shows that the physics behind this method is to consider the rigid domain as a fluid with infinite viscosity.

3. Taking the Collisions into Account

In [12], B. Maury proposed a numerical scheme to compute inelastic collisions between rigid particles. It consists in imposing a new constraint to the solution: the velocities of the particles are required to be in a set of *admissible velocities*.

We denote by $D_{ij}(\mathbf{x}^n) = |\mathbf{x}_i^n - \mathbf{x}_j^n| - r_i - r_j$ the signed distance between particles i and j and by $\mathbf{G}_{ij}(\mathbf{x}^n) \in \mathbb{R}^{2N}$ the gradient of this distance:

$$\mathbf{G}_{ij}(\mathbf{x}^n) = \underset{i}{(\ldots, 0, \quad -\mathbf{e}_{ij}^n}, 0, \ldots, 0, \quad \underset{j}{\mathbf{e}_{ij}^n}, 0, \ldots, 0)$$

where $\mathbf{e}_{ij}^n = \dfrac{\mathbf{x}_j^n - \mathbf{x}_i^n}{|\mathbf{x}_j^n - \mathbf{x}_i^n|}$. At each time step, $\mathbf{V} \in \mathbb{R}^{2N}$ is an *admissible* vector if the particles with velocity $\mathbf{V}$ do not overlap at the next time step:

$$E(\mathbf{x}^n) = \left\{ \mathbf{V} \in \mathbb{R}^{2N}, D_{ij}(\mathbf{x}^n + \Delta t \mathbf{V}) \geq 0 \quad \forall i < j \right\}.$$

We linearize the constraint and define the set of *admissible velocities* as the following closed and convex set:

$$K(\mathbf{x}^n) = \left\{ \mathbf{V} \in \mathbb{R}^{2N}, D_{ij}(\mathbf{x}^n) + \Delta t \mathbf{G}_{ij}(\mathbf{x}^n) \cdot \mathbf{V} \geq 0 \quad \forall i < j \right\}.$$

Note that, since the distance between two circular particles is convex with respect to $\mathbf{x}$, we have $K(\mathbf{x}^n) \subset E(\mathbf{x}^n)$.

For each $n > 0$, B^{n+1} is therefore computed at step (i) of the algorithm and step (ii) is now a variational problem similar to (7) where we take into account the new constraint:

$$\mathbf{V}^{n+1} \in K(\mathbf{x}^{n+1}) \text{ with } \forall i, \quad \mathbf{V}_i^{n+1} = \frac{1}{\pi r_i^2} \int_{B_i^{n+1}} \mathbf{u}^{n+1}.$$

We write this problem as a saddle-point problem and we denote by $\lambda_{ij}^{n+1} \geq 0$ the Lagrange multiplier associated to the constraint $D_{ij}(\mathbf{x}^{n+1}) + \Delta t \mathbf{G}_{ij}(\mathbf{x}^{n+1}) \cdot \mathbf{V} \geq 0$. We have to find $\mathbf{u}^{n+1} \in H_0^1(\Omega)$, $\mathrm{p}^{n+1} \in L^2(\Omega)$ and $\boldsymbol{\lambda}^{n+1} \in \mathbb{R}_+^{N(N-1)/2}$ such that:

$$\frac{1}{\Delta t} \int_\Omega \rho^{n+1} \mathbf{u}^{n+1} \cdot \tilde{\mathbf{u}} + 2\mu \int_\Omega \mathbf{D}(\mathbf{u}^{n+1}) : \mathbf{D}(\tilde{\mathbf{u}}) + \frac{2}{\varepsilon} \int_{B^{n+1}} \mathbf{D}(\mathbf{u}^{n+1}) : \mathbf{D}(\tilde{\mathbf{u}}) - \int_\Omega \mathrm{p}^{n+1} \nabla \cdot \tilde{\mathbf{u}}$$
$$= \frac{1}{\Delta t} \int_\Omega (\rho^n \mathbf{u}^n) \circ \mathrm{X}^n \cdot \tilde{\mathbf{u}} + \int_\Omega \mathbf{f}^{n+1} \cdot \tilde{\mathbf{u}} + \sum_{i<j} \lambda_{ij}^{n+1} \Delta t \mathbf{G}_{ij}(\mathbf{x}^{n+1}) \cdot \tilde{\mathbf{V}}, \quad \forall \tilde{\mathbf{u}} \in H_0^1(\Omega), \tag{8}$$

$$\int_\Omega \mathrm{q} \nabla \cdot \mathbf{u}^{n+1} = 0 \quad \forall \mathrm{q} \in L^2(\Omega), \tag{9}$$

$$D_{ij}(\mathbf{x}^{n+1}) + \Delta t \mathbf{G}_{ij}(\mathbf{x}^{n+1}) \cdot \mathbf{V}^{n+1} \geq 0, \quad \forall i < j, \tag{10}$$

$$\lambda_{ij}^{n+1} \left(D_{ij}(\mathbf{x}^{n+1}) + \Delta t \mathbf{G}_{ij}(\mathbf{x}^{n+1}) \cdot \mathbf{V}^{n+1} \right) = 0, \quad \forall i < j \tag{11}$$

where $\mathbf{V}_i^{n+1} = \dfrac{1}{\pi r_i^2} \int_{B_i^{n+1}} \mathbf{u}^{n+1}$ and $\tilde{\mathbf{V}}_i = \dfrac{1}{\pi r_i^2} \int_{B_i^{n+1}} \tilde{\mathbf{u}}$. Equation (11) is a compatibility equation imposing that a Lagrange multiplier is *active* (non zero) only if their is a contact between the associated particles. Equation (8) shows that λ_{ij}^{n+1} is proportional to a force that must be exerted on particles i and j in order to avoid their contact.

In [12], the saddle-point problem is numerically solved by a Uzawa algorithm on $\boldsymbol{\lambda}^{n+1}$. In our case, since the particles are embedded in a fluid, each iteration of this algorithm needs to solve a Stokes-like problem which is very time consuming. In order to avoid this problem, we propose a splitting strategy: for each time step we first compute ρ^{n+1} with (5), we solve the penalized problem (7), then compute the velocities of the particles

$\hat{\mathbf{V}}^{n+1}$ associated to $\mathbf{u}^{n+1}$ and finally project it onto the set of admissible velocities. This last step consists in finding $\mathbf{V}^{n+1}$ solution to:

$$\left|\mathbf{V}^{n+1} - \hat{\mathbf{V}}^{n+1}\right|^2 = \min_{\mathbf{V} \in K(\mathbf{x}^{n+1})} \left|\mathbf{V} - \hat{\mathbf{V}}^{n+1}\right|^2$$

and is performed using a Uzawa algorithm. The obstacle/particle collisions are treated in a similar way.

4. NUMERICAL TESTS

4.1. **Actual implementation with FreeFem++**

We chose to implement this algorithm with the finite element solver FreeFem++ (see [1]). The space discretization is carried out using the so-called mini-element (see [3]).

To compute $(\rho^n \mathbf{u}^n) \circ \mathbf{X}^n$, we use the function `convect` of FreeFem++. If $\mathbf{X}$ is the characteristic trajectory associated to the finite element velocity field $\boldsymbol{\alpha}$, $\mathbf{X}^n(\mathbf{x})$ is given by FreeFem as $\mathbf{x} - \Delta t \boldsymbol{\alpha}^n(\mathbf{x})$. The `convect` function is defined by $\texttt{convect}(\boldsymbol{\alpha}^n, -\Delta t, \mathbf{v}^n) = \mathbf{v}^n \circ \mathbf{X}^n(x)$ where $\mathbf{v}$ is a vector-valued finite element function. Consequently, in our case, we can compute

$$(\rho^n \mathbf{u}^n) \circ \mathbf{X}^n = \texttt{convect}(\mathbf{u}^n, -\Delta t, \rho^n \mathbf{u}^n).$$

In order to make the system inversible, in the variational formulation we add the term $\varepsilon_0 \mathrm{p}$ with $\varepsilon_0 << 1$ and we use the Crout solver to solve the system. Finally, the variational formulation implemented in FreeFem++ is the following:

```
problem NStokes([u1,u2,p],[v1,v2,q],solver=Crout)=
    //Navier-Stokes formulation
    int2d(Th)(rho*u1*v1-convect([uold1,uold2],-dt,rhoold*uold1)*v1
            +rho*u2*v2-convect([uold1,uold2],-dt,rhoold*uold2)*v2)
    +int2d(Th)(mu*dt*(2*dx(u1)*dx(v1)+dy(u1)*dy(v1)+dx(u2)*dx(v2)
            +2*dy(u2)*dy(v2)+dy(u1)*dx(v2)+dx(u2)*dy(v1)))
    +int2d(Th)(eps0*p*q - dt*p*dx(v1) - dt*p*dy(v2) + q*dx(u1) + q*dy(u2))
    -int2d(Th)(dt*f1*v1 + dt*f2*v2)
    //Penalty term
    +int2d(Th)(mu*(2*dx(u1)*dx(v1)+dy(u1)*dy(v1)+dx(u2)*dx(v2)
            +2*dy(u2)*dy(v2)+dy(u1)*dx(v2)+dx(u2)*dy(v1))*chi/eps)
    //Boundary conditions
    +on(1,2,3,4,u1=g1,u2=g2);
```

where `Th` is the mesh of Ω, `(u1,u2,p)` are the unknowns, `(v1,v2,q)` are their respective test functions, `(uold1,uold2)` is the velocity field computed at the previous time step, `rho` and `rhoold` are respectively the densities at the current and previous time steps , `chi` is the characteristic function of B, `(f1,f2)` is the source term and `(g1,g2)` is the boundary condition.

Finally, in order to have at least one mesh element between two neighbouring particles (which is necessary to compute acuratly the fluid remaining between these particles) we reduce the set of admissible velocities: if h is the mesh size, we choose

$$K(\mathbf{x}^n) = \left\{ \mathbf{V} \in \mathbb{R}^{2N}, D_{ij}(\mathbf{x}^n) + \Delta t \mathbf{G}_{ij}(\mathbf{x}^n) \cdot \mathbf{V} \geq \eta \right\} \quad \text{with} \quad \eta \approx h.$$

4.2. **Sheared particle**

To validate the penalty method, we consider the instantaneous problem of a particle embedded in a Stokes fluid. The computational domain is a square 1cm wide and a particle of radius 0.1cm is situated at its center. The right and left walls of the domain impose a shearing motion to the system (see Fig. 2), the viscosity of the fluid is equal to 1 and there is no external force.
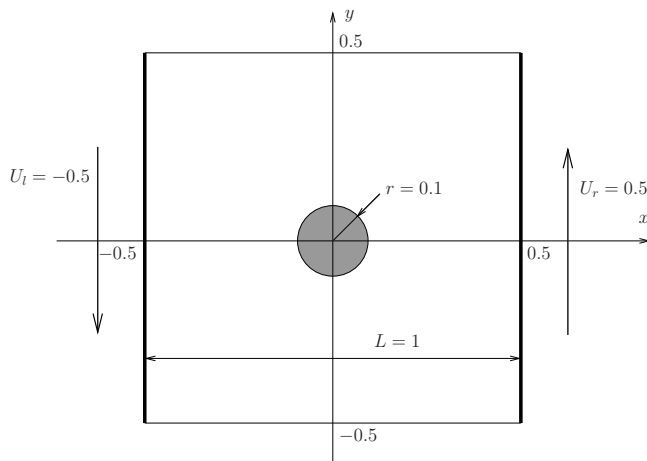
FIGURE 2. Sheared particle problem.

The simulations are run for two kinds of meshes: meshes that fit the particle and cartesian meshes (see Fig. 3). We denote by n the number of elements on each side of the square. Note that, to simulate the motion of particles, we shall use non fitted meshes in order to avoid remeshing at each time step.
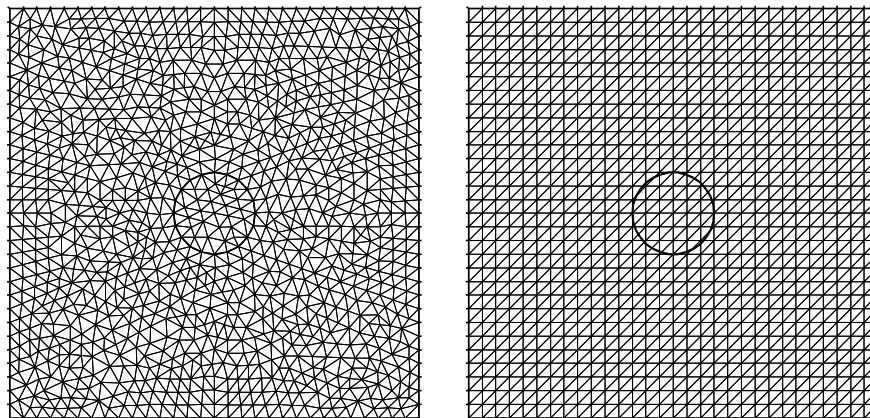

FIGURE 3. Sheared particle: Fitted (left) and cartesian (right) meshes for $n = 30$

Since we want to see if the rigid constraint is well taken into account, we plot in Fig. 4 the $L^2(B)$ norm of $\mathbf{D}(\mathbf{u})$ against ε for different meshes. We can observe that this norm is of order 10^{-5} when ε is of order 10^{-2}, which may allow us to choose ε not too small. We can also note that the rigid constraint is well taken into account even for relatively small values of n and that the cartesian mesh gives good results even if the fitted one is slightly better.
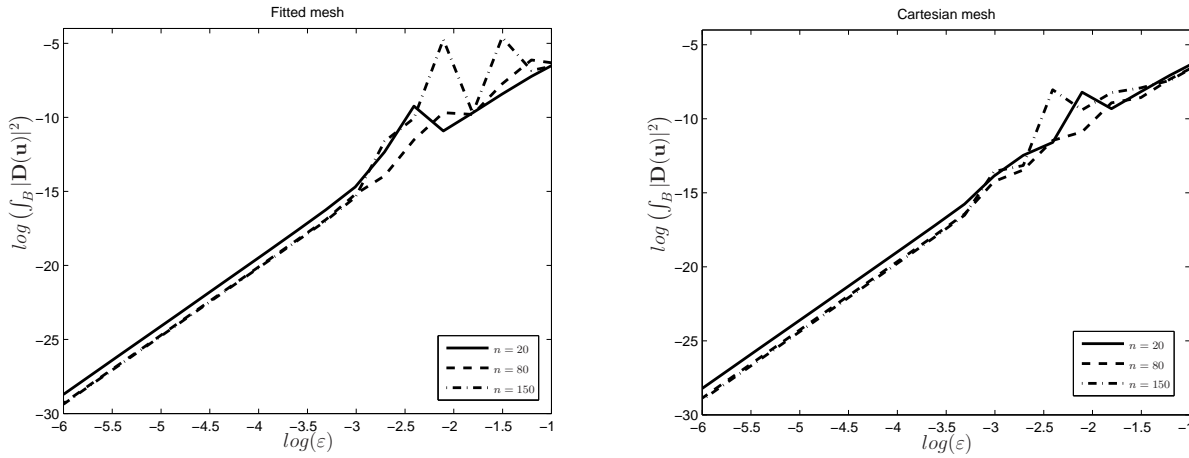
FIGURE 4. Sheared particle: Rigid movement in B, influence of n and ε.

It is known that the angular velocity of the particule converges to $\dfrac{\dot{\gamma}}{2} = \dfrac{U_r - U_l}{2L} = 0.5$ as its radius goes to zero. This value is recovered in our tests and, in Fig. 5 we can observe the streamlines of this rotational motion for $n = 150$ and $\varepsilon = 10^{-8}$.
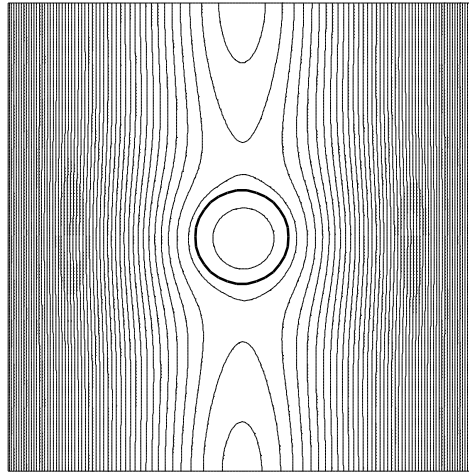

FIGURE 5. Sheared particle: Streamlines for a cartesian mesh with $n = 150$ and $\varepsilon = 10^{-8}$.

4.3. **Sedimentation of a particle : convergence of the penalty method**

In order to better understand the convergence of the penalty method when $h = 1/n$ and ε go to zero, we consider the instantaneous problem of the sedimentation of a single particle in a box, for which we are able to numerically compute an accurate reference solution. The particle is embedded in a Stokes fluid. As before (see Fig. 6), the computational domain is a square 1cm wide, the particle of radius 0.1cm is situated at its center and the viscosity is equal to 1. We impose homogeneous boundary conditions on the walls. The force acting on the particle is $\mathbf{f}_B = (0, -500)$.
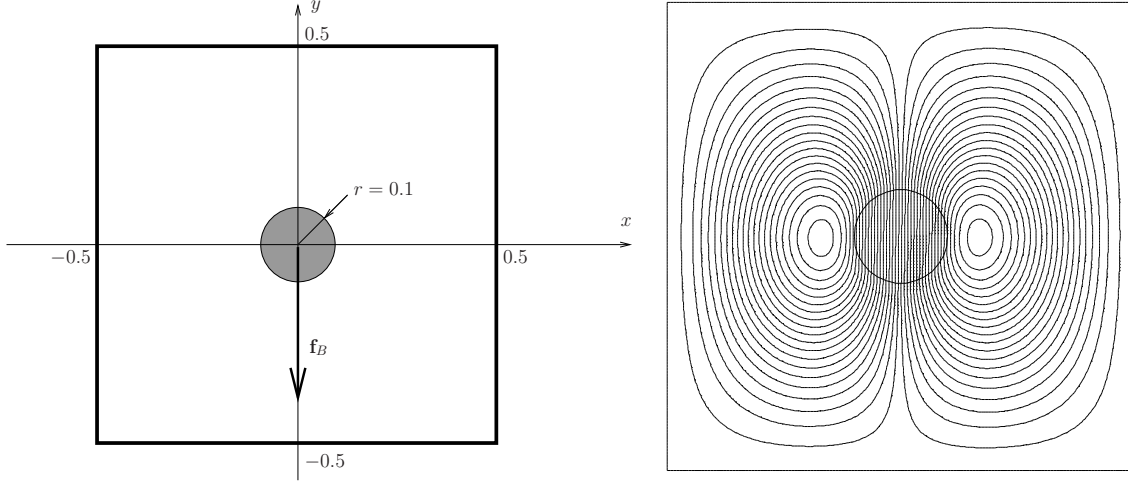
FIGURE 6. Sedimentation of a particule: description of the problem (left) and streamlines for a cartesian mesh with $n = 150$ and $\varepsilon = 10^{-8}$ (right).

To study the convergence of the method, we have to compute the solution to this problem (denoted by $\mathbf{u}$). To do so, we first define $(\mathbf{u}_1, \mathrm{p}_1)$ solution to the following problem:

$$\begin{cases} -\triangle \mathbf{u}_1 + \nabla \mathrm{p}_1 & = & 0 & \text{in } \Omega \setminus \bar{B} \\ \nabla \cdot \mathbf{u}_1 & = & 0 & \text{in } \Omega \setminus \bar{B} \\ \mathbf{u}_1 & = & 0 & \text{on } \partial\Omega \\ \mathbf{u}_1 & = & -1 & \text{on } \partial B \end{cases} \tag{12}$$

Symmetry considerations give that $\mathbf{u} = \alpha \mathbf{u}_1$ and $\mathrm{p} = \alpha \mathrm{p}_1$ where α is such that the balance of forces is zero:

$$\int_{\partial B} \sigma(\mathbf{u}) \cdot n = \int_B \mathbf{f}_B$$

therefore, α is given by

$$\alpha = \frac{\int_B \mathbf{f}_B}{\displaystyle\int_{\partial B} \sigma(\mathbf{u}_1) \cdot n}$$

and finally, an integration by part gives:

$$\alpha = \frac{\int_B \mathbf{f}_B}{\displaystyle\int_{\Omega \setminus \bar{B}} \mathrm{p}_1 \nabla \cdot \mathbf{u}_1 - 2\mu \int_{\Omega \setminus \bar{B}} \mathbf{D}(\mathbf{u}_1) : \mathbf{D}(\mathbf{u}_1)}.$$

$\mathbf{u}$ can be extended over Ω by $\mathbf{u} = -\alpha$ in B. We denote by $\mathbf{u}_{ref}$ the solution obtained by implementing this method in FreeFem++ over a boundary fitted fine mesh of $\Omega \setminus \bar{B}$ ($h = 1/150$). We are now interested in the evolution of the following three quantities when ε and h go to zero:

$$e_K = \|\mathbf{D}(\mathbf{u}_h^\varepsilon)\|_{L^2(B)}$$
$$e_{L^2} = \|\mathbf{u}_h^\varepsilon - \mathbf{u}_{ref}\|_{L^2(\Omega)}$$
$$e_{H^1} = \|\mathbf{u}_h^\varepsilon - \mathbf{u}_{ref}\|_{H^1(\Omega)}$$

where $\mathbf{u}_h^\varepsilon$ denotes the solution to the sedimentation problem given by the FreeFem++ implementation described in section 4.1 for a penalty parameter equal to ε and a mesh size equal to h.
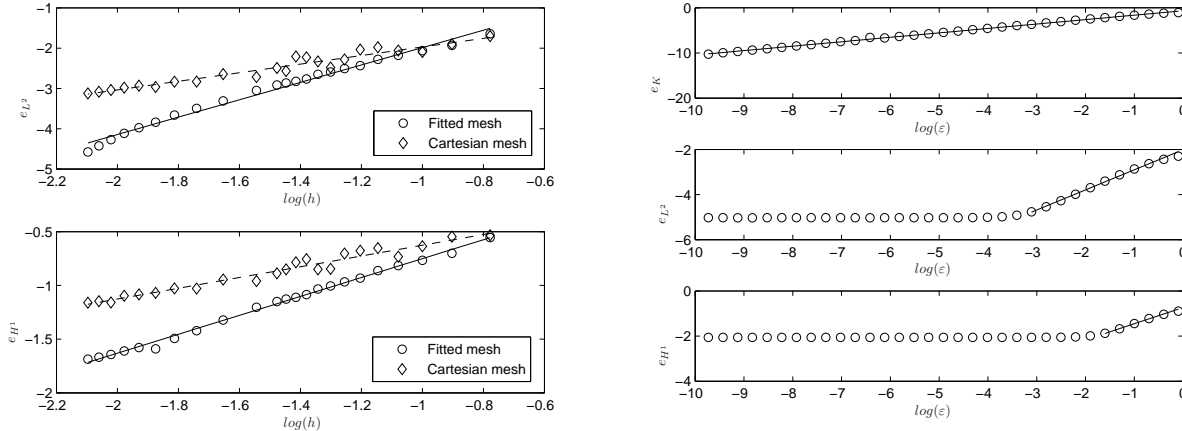

FIGURE 7. Sedimentation of a particle: convergence of the penalty method.

On the left side of Fig. 7, ε is fixed to 10^{-8}. We plot $log(e_{L^2})$ and $log(e_{H^1})$ versus $log(h)$. The following convergence rates are observed:

	Fitted mesh	Cartesian mesh
e_{L^2}	2.1577	0.8798
e_{H^1}	1.0658	0.5007

TABLE 1: Convergence rates when h goes to zero

On its right side, $\mathbf{u}_h^\varepsilon$ is computed over a fitted mesh and h is fixed to $1/150$. We plot $log(e_K)$, $log(e_{L^2})$ and $log(e_{H^1})$ versus $log(\varepsilon)$. As expected, a saturation due to the space discretization error is observed for e_{L^2} and e_{H^1}. We observe the following convergence rates:

	Fitted mesh
e_K	0.9806
e_{L^2}	0.8909
e_{H^1}	0.7177

TABLE 2: Convergence rates when ε goes to zero

To conclude, we can say that the convergence rate in space for a fitted mesh is the one given by the Finite Element theory. Even though it is divided by 2 for cartesian meshes we must recall that these meshes are of great interest for non-stationnary problems. This study also confirms that the convergence versus ε is fast enough to allow us to use not too small ε: for ε lower than 10^{-2}, the space discretization error becomes the leader term of the error in $H^1(\Omega)$. However, the convergence rate of e_{H^1} to zero before this saturation effect is somewhat surprising: for very fine meshes, one would have expected the global error to behave like the error for the continuous penalized problem which is of order ε (see [9]). A more precise investigation should be undergone to understand that phenomenon.

4.4. Sedimentation of two particles

To show that our scheme reproduces the behaviour of physical non-stationary systems, we now present the sedimentation of two neighbouring particles in a closed channel. We consider a channel 2cm wide and 5cm tall: $\Omega = [0,2] \times [0,5]$. This channel is filled with a Navier-Stokes fluid and the simulation is started by dropping

two particles of diameter $d = 0.25$cm from points $(1, 4.5)$ and $(1 + 0.2r, 4)$ where r is the radius of the particles. The viscosity of the fluid is $\mu = 0.01$, the densities of the fluid and of the particles are $\rho_f = 1$ and $\rho_B = 2$. Fig. 8 shows the configurations obtained at different time steps and reproduces the well-known phenomenon of drafting, kissing and tumbling (see [4]).
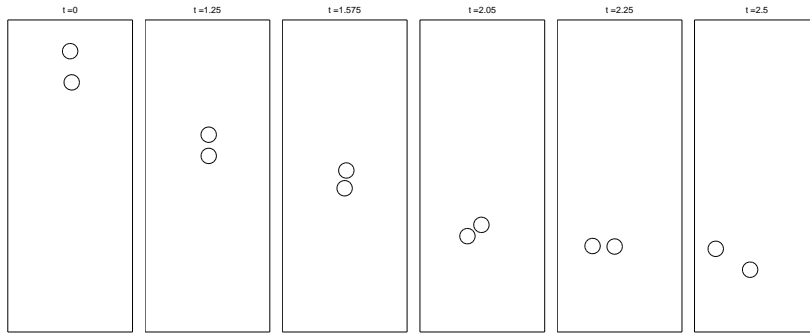

FIGURE 8. Sedimentation of two particles: configurations at different time steps for $n = 50$ and $dt = 0.005$.

4.5. **Sedimentation of 228 particles**

In Fig. 9, we plot the results of the sedimentation of 228 particles in a closed box filled with a Navier-Stokes fluid. The box is a square 2cm wide, the radius of the particles is $r = 0.04$, the viscosity of the fluid is $\mu = 0.01$ and the densities of the fluid and of the particles are $\rho_f = 1$ and $\rho_B = 1.5$. The simulation is run for $n = 50$ and $dt = 0.01$.
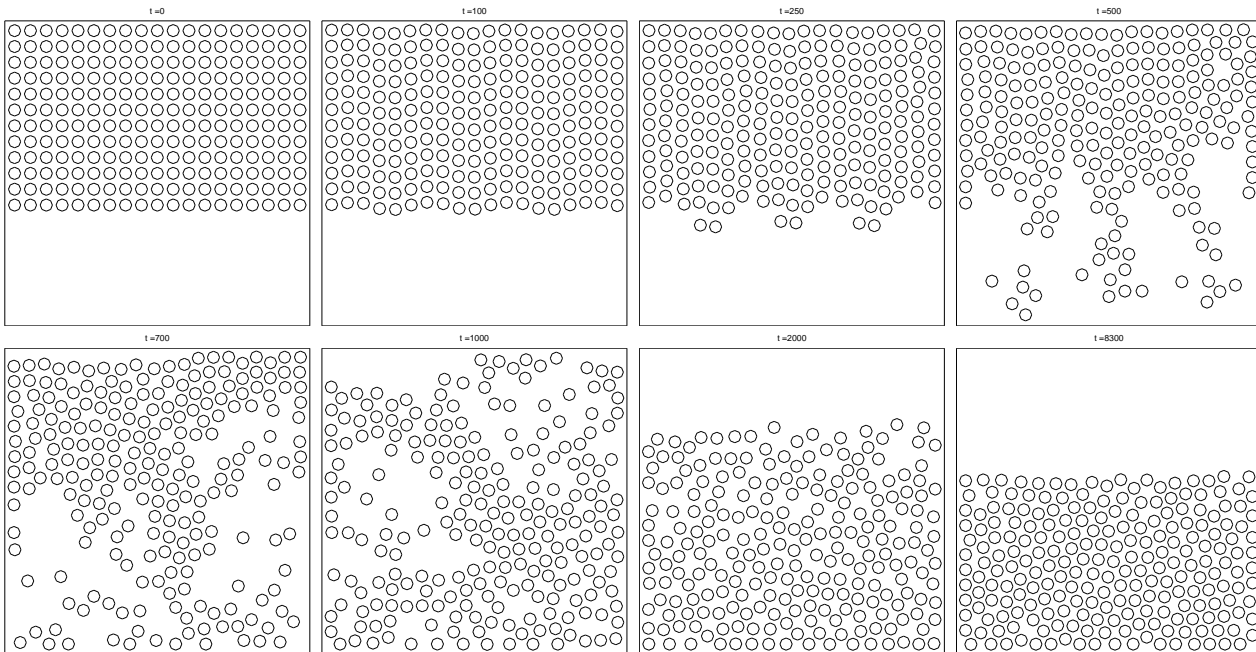

FIGURE 9. Sedimentation of 228 particles: configurations at different time steps.

Conclusion

We have presented in this paper a method to simulate the motion of rigid particles in Newtonian fluids: the rigid motion is imposed by penalizing the strain tensor, the time discretization is performed using the method of characteristics and the collisions are taken into account thanks to a projection method.

This algorithm allows us to use fixed cartesian meshes even for non-stationnary problems and is easy to implement from any Stokes/Navier-Stokes solver. Moreover, it is straightforward to extend it to the three-dimensional case provided we have a 3D-Stokes/Navier-Stokes solver.

Whereas the systems usually obtained by using penalty methods are ill conditioned, a numerical investigation of the convergence of our scheme showed that we can choose ε not too small in order to deal with reasonably conditioned systems. However, the interactions between parameters h and ε seem to be important and shall be better understood. In order to do so, we would like to extend to the fluid case the numerical analysis presented in [13] of the penalty method in the finite element context for the Poisson's problem.

References

[1] http://www.freefem.org

[2] P. Angot, C-H. Bruneau, P. Fabrie, *A penalization method to take into account obstacles in incompressible viscous flows*, Numerische Mathmatik, Vol. 81(4), pp. 497-520, 1999

[3] D. Arnold, F. Brezzi, M. Fortin, *A stable finite element for the Stokes equations*, Calcolo 21(4) 337-344,1984

[4] A.F. Fortes, D.D. Joseph, T.S. Lundgren, *Nonlinear mechanics of fluidization of beds of spherical particles*, J. Fluid Mech. (1987), vol. 177, pp. 467-483

[5] 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 (1999) 755-794

[6] M. Hillairet, *Lack of collision between solid bodies in a 2D constant density incompressible viscous flow*, Submitted

[7] H.H. Hu, D.D.Joseph, M.J. Crochet, *Direct simulation of fluid particle motion*, Theoret. Comput. Fluid Dynamics (1992) 3: 285-306

[8] H.H. Hu, *Direct simulation of flows of solid-liquid mixtures*, Int. J. Multiphase Flow Vol.22, No. 2, pp. 335-352, 1996

[9] J. Janela, A. Lefebvre, B. Maury, *A penalty method for the simulation of fluid - rigid body interaction*, ESAIM: Proc., September 2005, Vol. 14, pp. 115-123

[10] A.A. Johnson, T.E. Tezduyar, *Simulation of multiple spheres falling in a liquid-filled tube*, Comput. Methods Appl. Mech. Engrg. 134 (1996) 351-373

[11] B. Maury, *Direct simulation of 2D fluid-particle flows in biperiodic domains*, Journal of Computational physics 156, 325-351 (1999)

[12] B. Maury, *A time-stepping scheme for inelastic collisions*, Numerische Mathematik, Volume 102, Number 4, pp. 649 - 679, 2006

[13] B. Maury, *Numerical Analysis of a Finite Element / Volume Penalty Method*, Submitted

[14] N.A. Patankar, P. Singh, D.D. Joseph, R. Glowinski, T-W. Pan, *A new formulations for the distributed lagrange multiplier/fictitious domain method for particulate flows*, International Journal of Multiphase Flow 26 (2000) 1509-1524

[15] O. Pironneau, J. Liou, T. Tezduyar, *Characteristic-Galerkin and Galerkin/least-squares space-time formulations for the advection-diffusion equations with time-dependent domains*, Comput. Methods Appl. Mech. Engrg. 100 (1992), no. 1, 117–141.

[16] T.N. Randrianarivelo, G. Pianet, S. Vincent, J.P. Caltagirone, *Numerical modelling of solid particle motion using a new penalty method*, Int. J. Numer. Meth. Fluids 2005; 47:1245-1251

[17] P. Singh, T.I. Hesla, D.D. Joseph, *Distributed lagrange multiplier method for paticulate flows with collisions*, International Journal of Multiphase Flow 29 (2003) 495-509