
Numerical methods for solid particles in particulate flow simulations

Patrice Laure* — Gregory Beaume**,*** — Olivier Basset**
Luisa Silva** — Thierry Coupez**

* *Laboratoire J. A. Dieudonné, UMR 6621 CNRS - Université de Nice
F-06108 Nice cedex 2*

Patrice.Laure@unice.fr

** *Centre de Mise en Forme des Matériaux, Ecole Nationale Supérieure
des Mines de Paris UMR 7635 CNRS, F-06904 Sophia-Antipolis*

{Gregory.Beaume, Luisa.Silva, Thierry.Coupez}@ensmp.fr

*** *Schneider Electric, Technopôle 38 TEC, F-38000 Grenoble*

ABSTRACT. The flow motion of solid particle suspensions is a fundamental issue in many problems of practical interest. The velocity field of a such system is computed by a finite element method with a multi-domain approach of two phases (namely a viscous fluid and rigid bodies), whereas the particle displacement is made by a particulate method. We focus our paper on a simple shear flow of Newtonian fluid.

RÉSUMÉ. Les écoulements de fluides chargés de particules interviennent dans de nombreux procédés industriels. Le champ de vitesse dans un tel système est donné par une méthode éléments finis associée à une formulation multidomaine (le fluide visqueux et les particules solides). Puis le déplacement des particules est effectué par une méthode particulaire. Cette approche est ici testée sur un écoulement de cisaillement simple.

KEYWORDS: particle suspensions, particulate flows, finite element method, direct numerical simulations.

MOTS-CLÉS : suspension, écoulements particuliers, méthode éléments finis, simulations numériques directes.

DOI:10.3166/REMN.16.365-383© 2007 Lavoisier, Paris. Tous droits réservés

1. Introduction

We present a new finite element method for the simulation of particulate Stokes flows. This formulation is based on the fictitious domain method (Glowinski *et al.*, 1998; Singh *et al.*, 2000), which consists in treating the entire fluid-particle domain as a fluid. The fluid in the particle domain is ensured to move accurately by adding the rigidity constraint in the weak formulation with a distributed Lagrange multiplier. The fluid flow problem is then formulated on a larger domain (the "fictitious domain") which is simpler and allows us to use a fixed regular mesh. In return, we have to project the particle domain in the fictitious one. The originality of our approach is the use of a characteristic function to describe the particle domain. We present two methods (Volume of Fluid and Level Set methods) to compute this function, and two approaches to move the domain (*i.e.* to update the characteristic function for particle domain at each time steps): the solution of advection equations and Lagrangian methods.

Most of these results are issued from (Megally, 2005), (Gruau, 2004) and (Bigot, 2001) PHDs. We use this method to simulate a complex fluid flow (Newtonian fluid matrix, charged with both elongated and spherical particles, at high solid concentration), supposed to describe industrial process of Bulk Molding Compounds (BMC) injection. In comparison with lubrication theory (Fan *et al.*, 1998) or slender body theory (Yamane *et al.*, 1994), the fictitious domain method is especially interesting for injection of fiber-reinforced composite, because we do not need the explicit form of drag forces or hydrodynamic interactions between particles, as forces on the particles boundary do not appear in the weak formulation.

In this framework, the numerical procedure is characterized by the way each of the following problems is solved: characterization of the solid part; computation of velocity field; particle displacement.

Let us introduce few notations which will be used in the sequel: the velocity field will be defined in the usual Sobolev space $\mathcal{H}^1(\Omega)$, the pressure in $\mathcal{L}^2(\Omega)$, the Lebesgue space of square-integrable measurable functions on Ω . The inner product on $\mathcal{L}^2(\Omega)$ is defined by

$$(f_1, f_2) = \int_{\Omega} f_1 f_2 d\Omega \quad [1]$$

Finally, the symbol h is used to express the variables on discrete space Ω_h built as the union of all simplexes (or elements) K that form the domain mesh.

2. Characterization of solid and fluid domains

The whole domain Ω is split into two sub-domains Ω_f and Ω_s which are associated respectively to fluid and solid areas (the subscripts f , s mean fluid and solid respectively). In fact, the solid domain Ω_s consists of several particles (namely $\Omega_s = \bigcup_i^n \Omega_{s_i}$ for n particles).

The weak formulation is written on the whole domain. In order to take into account each domain, the corresponding characteristic functions \mathbb{I}_j , $j \in \{f, s\}$ are introduced:

$$\mathbb{I}_j(x, t) = 1 \text{ if } x \in \Omega_j \text{ and } 0 \text{ if } x \notin \Omega_j \quad [2]$$

In the sequel, one only deals with the characteristic function of solid domain \mathbb{I}_s for legibility reasons. According to chosen approximations ($P0$ or $P1$), there are two different approaches: "Volume of Fluid" or "Level Set" methods.

2.1. Volume of fluid method

The characteristic function is approximated by a constant value on each element K , \mathbb{I}_s^h . This $P0$ approximation is simply the volume fraction occupied by the solid domain Ω_s on each element:

$$\mathbb{I}_s^h|_K = \mathbb{I}^e = \frac{|K| \cap |\Omega_s|}{|K|}. \quad [3]$$

The computation of characteristic function for a simple geometrical form, can be performed using a voxelization algorithm which is described in Figure 1. The main steps of this procedure are the following:

- an array of voxels (or pixel in 2D) is associated to the whole domain mesh (Figure 1a). The size of this array can be changed accordingly to the mesh refinement in order to get a correct description of the interface,
- the voxels or pixels that belong to solid domain are marked (Figure 1b). The solid domain can be defined by a mesh or an analytical formula. This step is made by checking each node of regular mesh contained into the simplex K . This mesh has the same refinement as the array of pixels,
- finally the characteristic function in each element K is simply the ratio between number of voxels which belong to both solid domain and simplex K and the number of voxels which are in K (Figure 1c).

The exact interpolation error is

$$\|\mathbb{I}_s^h - \mathbb{I}_s\| = \sum_K ((1 - \mathbb{I}^e)\mathbb{I}^e|K|)^{1/2} \quad [4]$$

Consequently, this errors can be decreased by reducing the volume of elements which are crossed by the interface. This may be done by a r-adaptation procedure (Bigot, 2001; Baines, 1998). In this way, a new mesh is obtained by local node movements, the topology of mesh remaining fixed.

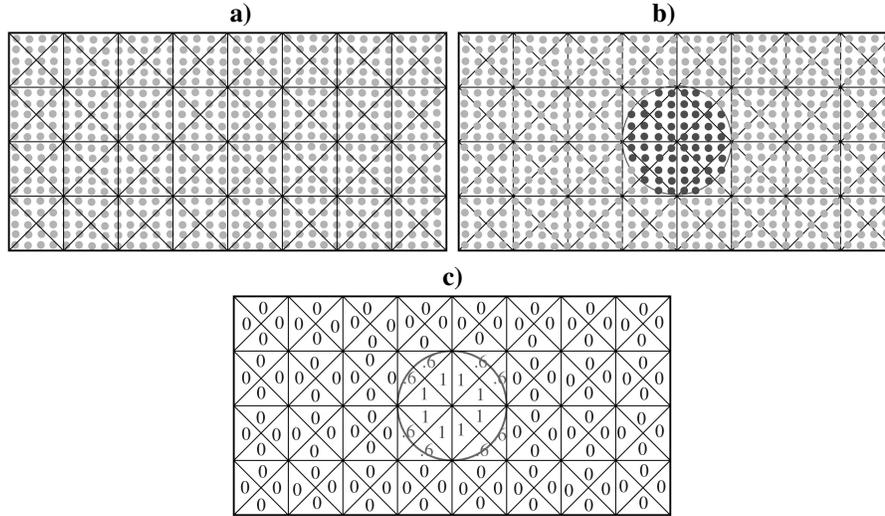


Figure 1. Voxelization algorithm and computation of $P0$ approximation of characteristic function: (a) voxelization of computational domain; (b) computation of voxels which belong to solid domain; (c) computation of characteristic function (ratio between black pixels over all pixels in each simplex K)

2.2. The "level set" method

The principle of the level set technique (Osher *et al.*, 1988) is to define an interface function in the computational domain from which the level zero trace is the interface that we wish to describe. This function is positive inside one sub-domain and negative elsewhere. Moreover, this function is continuous and more regular than the characteristic function defined in relation [2]. Therefore, a $P1$ interpolation of this function gives an accurate description of the interface. If the interface of solid domain is defined by a curve Γ_s , the level set function can be defined from a signed distance as:

$$\begin{cases} \alpha(x) = \|x - \Gamma_s\| & \text{in } \Omega_s \\ \alpha(x) = -\|x - \Gamma_s\| & \text{in } \Omega_f \end{cases} \quad [5]$$

where $\| \cdot \|$ is the Euclidian norm in \mathcal{R}^d . Figure 2a shows the isovalues of such function for Γ_s that has a shape of a disk with radius 0.25 and center (1.0, 0.5). As a consequence

$$\alpha = \pm \sqrt{|(0.25)^2 - (x - 1.0)^2 - (y - 0.5)^2|}.$$

A $P1$ approximation of a characteristic function \mathbb{I}_s is easily obtained by looking at the sign of $\alpha_h(x_i)$ at the mesh nodes x_i :

$$\mathbb{I}_s^h(x_i) = \begin{cases} = 1 & \text{s } \alpha(x_i) > 0 \\ = 0 & \text{if } \alpha(x_i) < 0 \end{cases} \quad [6]$$

Furthermore, a more regular transition is obtained by taking the following function

$$\mathbb{I}_s = \frac{1}{1 + e^{-\beta \alpha}} \quad , \quad \beta > 0 \quad [7]$$

where the transition area will depend on the value of β .

Finally a $P0$ approximation of \mathbb{I}_s can also be computed from the distance function α as follows:

$$\mathbb{I}_s^h|_K = \frac{\alpha_K^+}{|\alpha|_K} \quad [8]$$

where α_K^+ is the sum of all positive α determined at nodes of element K , whereas $|\alpha|_K$ is the sum of absolute values of α . Thus, $\mathbb{I}_s^h|_K$ equals 1 or 0 if all the nodal values of α in the element K are positive or negative, whereas the element crossed by the interface have a characteristic function lying between 1 and 0 (see Figure 2b).

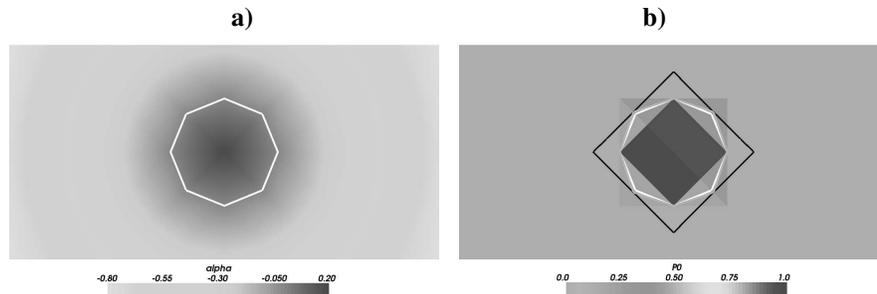


Figure 2. Level set function for a disc of radius 0.25: (a) isovalues of distance function (in white isoline $\alpha = 0$); (b) the various characteristic functions deduced from function α : $P0$ interpolation thanks to relation [8], isolines 1/2 of the characteristic functions given by relation [6] in black and the relation [7] with $\beta = 100$ in gray

In conclusion, one gets a "fuzzy" interface between solid and fluid domains. Even with a coarse mesh, the zero level gives a rather accurate description of interface. However, the method used to get the characteristic function from the level set function has an influence on the size of the transition zone between the two phases (see Figure 2). For example, according to the choice of relations [6] or [8], one has to refine more or less the area around the solid domain. This characteristic function is important as it intervenes in the mixing relations (see next Section) which give the values of viscosity and density on the whole domain.

3. Computation of the velocity field

The governing equations for Newtonian fluid motion are given by:

$$\left\{ \begin{array}{l} \rho_f \left[\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right] = \rho_f \vec{g} + \nabla \cdot \underline{\underline{\sigma}} \\ \nabla \cdot \vec{u} = 0 \\ \underline{\underline{\sigma}} = -p\mathbf{I} + 2\eta_f \underline{\underline{\dot{\epsilon}}}(\vec{u}) \\ [[\vec{u}]]_{\Gamma_s} = 0 \\ [[\underline{\underline{\sigma}} \cdot \vec{n}]]_{\Gamma_s} = 0 \\ \vec{u} = \vec{u}_\Gamma \quad \text{on the external boundary } \Gamma \end{array} \right. \quad [9]$$

where \vec{u} is the fluid velocity, p the pressure, η_f the fluid viscosity, ρ_f its density.

Patankar *et al.* (Patankar *et al.*, 2000) have proposed to extend the above Navier-Stokes equation to the solid domain thanks to a Lagrange multiplier by using the rigidity condition $\underline{\underline{\dot{\epsilon}}}(\vec{v}) = 0$ on Ω_s . In this way, the motion in solid domain Ω_s corresponds to a fluid motion with an additional stress field. That means

$$\left\{ \begin{array}{l} \rho_s \left[\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right] = \rho_s \vec{g} + \nabla \cdot \underline{\underline{\sigma}} \\ \nabla \cdot \vec{u} = 0 \\ \underline{\underline{\dot{\epsilon}}}(\vec{u}) = 0 \\ [[\vec{u}]]_{\Gamma_s} = 0 \\ [[\underline{\underline{\sigma}} \cdot \vec{n}]]_{\Gamma_s} = 0 \end{array} \right. \quad [10]$$

ρ_s being the density of solid particles.

This is equivalent to take the stress tensor σ inside the solid domain of the form

$$\underline{\underline{\sigma}} = 2\eta_s \underline{\underline{\dot{\epsilon}}}(\vec{u}) - p\mathbf{I} + \underline{\underline{\dot{\epsilon}}}(\vec{\lambda}) \quad [11]$$

Due to the rigid motion constraint the two first terms are zero; η_s can play the role of a penalization factor of the constraint $\underline{\underline{\dot{\epsilon}}}(\mathbf{u}) = 0$, and the symmetrical tensor $\underline{\underline{\dot{\epsilon}}}(\vec{\lambda})$ is the Lagrange multiplier associated to this constraint.

3.1. Weak formulation of fluid solid system

With Diriclet boundary conditions on external walls, the weak formulation for the whole domain Ω becomes:

find $(\vec{u}, p, \vec{\lambda})$ such that $\forall (\vec{v}, q, \vec{\mu}) \in \mathcal{H}^1(\Omega) \times \mathcal{L}_0^2(\Omega) \times \mathcal{H}^1(\Omega_s(t))$:

$$\left\{ \begin{array}{l} 0 = 2\eta \int_{\Omega} \underline{\dot{\epsilon}}(\vec{u}) : \underline{\dot{\epsilon}}(\vec{v}) d\Omega - \int_{\Omega} p \nabla \cdot \vec{v} d\Omega \\ \quad + \int_{\Omega} \rho \left(\frac{D\vec{u}}{Dt} - \vec{g} \right) \cdot \vec{v} d\Omega + \int_{\Omega_s} \underline{\dot{\epsilon}}(\vec{\lambda}) : \underline{\dot{\epsilon}}(\vec{v}) d\Omega \\ 0 = \int_{\Omega} q \nabla \cdot \vec{u} d\Omega \\ 0 = \int_{\Omega_s} \underline{\dot{\epsilon}}(\vec{\mu}) : \underline{\dot{\epsilon}}(\vec{v}) d\Omega \end{array} \right. \quad [12]$$

where ρ et η are defined on Ω_s thanks to mixing relations,

$$\rho = \mathbb{I}_s \rho_s + (\mathbb{I} - \mathbb{I}_s) \rho_f ; \eta = \mathbb{I}_s \eta_s + (\mathbb{I} - \mathbb{I}_s) \eta_f \quad [13]$$

3.2. Discrete formulation and Uzawa algorithm

We deal with a non linear problem, time integration and computation of Lagrange multiplier. An implicit time scheme is used and an iterative loop allows us to solve both nonlinearity and rigidity constraint.

At each time step t_n the procedure is the following:

1. initialization with values obtained at the previous time step:

$$\vec{u}_h^0 = \vec{u}_h(t_{n-1}), p_h^0 = p_h(t_{n-1}), \vec{\lambda}_h^0 = 0, \vec{u}_h^* = \vec{u}_h(t_{n-1})$$

2. at step k find \vec{u}_h^k et p_h^k with system

$$\left\{ \begin{array}{l} \left(\rho \frac{\vec{u}_h^k}{\Delta t}, \vec{v}_h \right) + \left(\rho \nabla \vec{u}_h^k \cdot \vec{u}_h^{k-1}, \vec{v}_h \right) + 2 \left(\eta \underline{\dot{\epsilon}}(\vec{u}_h) : \underline{\dot{\epsilon}}(\vec{v}_h) \right) \\ - \left(p_h^k, \nabla \cdot \vec{v}_h \right) = \left(\rho \vec{g}, \vec{v}_h \right) + \left(\rho \frac{\vec{u}_h^*}{\Delta t}, \vec{v}_h \right) \\ + \left(\rho \nabla \vec{u}_h^{k-1} \cdot \vec{u}_h^{k-1}, \vec{v}_h \right) - \left(\mathbb{I}_s \underline{\dot{\epsilon}}(\vec{\lambda}_h^{k-1}) : \underline{\dot{\epsilon}}(\vec{v}_h) \right) \\ \left(\nabla \cdot \vec{u}_h^k, q_h \right) = 0 \end{array} \right. \quad [14]$$

3. update $\vec{\lambda}$:

$$\vec{\lambda}_h^k = \vec{\lambda}_h^{k-1} + \eta_s \vec{u}_h^k$$

4. check $\|\vec{u}_h^{k-1} - \vec{u}_h^k\| < \epsilon_1$ and $\|\mathbb{I}_s \underline{\underline{\dot{u}}}_h^k\| < \epsilon_2$ to stop the loop on k .

One can remark that in system [12] the field $\vec{\lambda}$ is only defined on the solid domain Ω_s . For practical reasons this field is extended to the whole domain Ω because only the part inside Ω_s is took into account thanks to characteristic function.

The Navier-Stokes problem is solved using mixed finite element method. This method enters in the family of mini-element first introduced in (Arnold *et al.*, 1984). It is a tetrahedral first-order element with a linear continuous interpolation of both pressure and velocity and a bubble enrichment for the velocity. The bubble function ensuring the stability condition (Pichelin *et al.*, 1998), is built by four piecewise linear functions (Fortin *et al.*, 1991) called also pyramidal shape function. This particular choice enables to preserve the exact integration property of this element. Once the bubble term is condensed, it looks like a simple optimally stabilized equal order velocity and pressure interpolation $P1 + /P1$ (Franca *et al.*, 1992; Pierre, 1995).

If we choose a piecewise constant interpolation ($P0$ element) for the characteristic function, there is no loose of accuracy in the integration of term η as it is a factor of a term which constant by element. However as remarked in the previous Section, the method used to get this interpolation can modify the value of η in the transition area. The order of interpolation is more important for the density as the number of Gauss integration coordinates has to be increased for a $P1$ interpolation.

4. Particle displacements

The different approaches of the fictitious domain technique allow us to determine the velocity field in each point of the computational domain. However, each subdomain will evolve in time: for example, particles' displacement must be computed. In fact, once the whole problem has been discretized both in time and in space, two steps are performed at each time step: firstly, the velocity distribution is determined; then, particle position is updated. The displacement of the particles has been performed using two different techniques: through the solution of the advection equation on the distance function (Equation [16]) associated to particles, or by performing a Lagrangian displacement of the last ones.

4.1. Displacement of level set function (SUPG method)

In Section 2.2, the level set function has been defined from zero level of the signed distance function. The distance function can be used to give an initial value of the

level set function. In order to describe its evolution in time, one can define it in an alternative way:

$$\begin{cases} \|\nabla\alpha\| = 1 & \text{in } \Omega \\ \alpha = 0 & \text{on } \Gamma \end{cases} \quad [15]$$

This stationary problem used to determine the distance function is generally converted into a non-stationary one, for which α is the stationary solution. The following relationship is considered in the whole computational domain:

$$\frac{\partial\alpha}{\partial t} + \vec{u} \cdot \nabla\alpha = 0 \quad [16]$$

with the initial condition $\alpha(x, 0) = \alpha_0$ and $\alpha(x, t) = g$ in Γ^- , where

$$\Gamma^- = \{x \in \partial\Omega, \vec{u} \cdot \vec{n} < 0\}$$

represents the inlet boundary. Therefore we transport all the level sets, even if we loose the regularity of the initial distance function. The interface is always represented implicitly by its zero isovalue.

The weak formulation of this problem is:

$$\begin{aligned} \text{find } \alpha \in \mathcal{H}^1(\Omega) \text{ such that } \forall \phi \in \mathcal{H}_0^1(\Omega) = \{\phi \in \mathcal{H}^1(\Omega), \phi = 0 \text{ on } \Gamma^-\} \\ (\partial_t\alpha, \phi) + (\vec{u} \cdot \nabla\alpha, \phi) = 0 \end{aligned} \quad [17]$$

The temporal discretization of this equation is written

$$\left(\frac{\alpha_h}{\Delta t}, \phi_h\right) + \theta(\nabla\alpha_h \cdot \vec{u}, \phi_h) = \left(\frac{\alpha_h^-}{\Delta t}, \phi_h\right) + (1 - \theta)(\nabla\alpha_h^- \cdot \vec{u}, \phi_h) \quad [18]$$

where α_h^- is the value of the distance function at the previous time step. In the case where $\theta = 1$, we have an Euler implicit scheme; if $0 < \theta < 1$, the scheme is semi-implicit; when $\theta = 0$, it is explicit. In the examples shown, we have considered $\theta = 0.5$.

In what concerns spatial discretization, classical finite element methods are not very adequate to the resolution of pure convection problems. Centered differences give erroneous oscillations when the problem is of the hyperbolic type, and stabilization techniques are, for example, the Streamline Upwind Petrov Galerkin (SUPG) or Residual-Free Bubbles (RFB), built by adding an additional diffusion term in the flow direction. If we consider the SUPG method, the final variational formulation can be written by supposing that the test functions ϕ_h are built in another discretization space (Brooks *et al.*, 1982), to obtain $\tilde{\phi}_h$:

$$\tilde{\phi}_h = \phi_h + \tau_K \vec{u} \cdot \nabla\phi_h \quad [19]$$

where the stabilisation parameter τ_K (constant per element) depends on the local discretization, represented by the mesh size h_K . We may define

$$\tau_K = \frac{h_K}{2 |u_K|} \quad [20]$$

This method will perform well in the advection of scalar fields that do not present high gradients ($|\nabla\alpha| \sim 1$). Nevertheless, and as it has already been mentioned, the transport of the distance function does not guarantee that the gradient of α rests weak. If $|\nabla\alpha|$ becomes very important, we must replace α by another function, more regular, that has the same isovalue zero. This can be accomplished by solving an Hamilton-Jacobi equation:

$$\begin{aligned} & \text{At each time step} \\ & \phi(x, 0) = \alpha(x, t) \\ & \text{For } \tau = 0 \text{ to } \epsilon, \text{ solve} \\ & \frac{\partial \phi}{\partial \tau} = S(\phi)(1 - |\nabla \phi|) \\ & \alpha(x, t) = \phi(x, \epsilon) \end{aligned} \quad [21]$$

In this Hamilton-Jacobi equation, τ is a fictitious time, and S is a signed function, defined as:

$$S(\phi) = \begin{cases} -1 & \text{if } \phi < 0 \\ 0 & \text{if } \phi = 0 \\ 1 & \text{if } \phi > 0 \end{cases} \quad [22]$$

This function is not differentiable when $\phi = 0$, so other approximations may be preferred, for example (Peng *et al.*, 1999)

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\phi|^2 h_K^2}} \quad [23]$$

4.2. Lagrangian particle displacement

This approach has been used in Megally's thesis (Megally, 2005; Megally *et al.*, 2004) in order to move particles. As the particle has a solid motion, it is sufficient to move few points to get the new positions: one point for a sphere, two points for an axisymmetric solid and three points for an undefined solid. The new position of a point belonging to the solid is given by the relation

$$\vec{X}(t + \Delta t) = \vec{X}(t) + \Delta t \vec{u}(\vec{X}) \quad [24]$$

However, one need an efficient algorithm to interpolate the velocity at chosen points (that means in our case "find in which element belongs the center of sphere") because an important computational time is necessary for large meshes.

With this kind of approach, it is necessary to update the characteristic functions for the new positions of particles before computing again the velocity field. To do that, one can use one of methods presented in Section 3. In this way, there is no deformation and no loss of mass for the solid domain. This explicit time integration scheme can be improved by taking into account the velocity at previous time steps. For example, a second order Adams- Bashfort scheme (Hwang *et al.*, 2004) can be chosen.

5. Numerical results

Here, we want to discuss the performance of techniques presented in the previous Sections. A simple case of two spheres in a 2D shear rate flow is considered. Moreover inertial effect and gravity are neglected. This configuration has been already studied by (Hwang *et al.*, 2004) with other boundary conditions. A rectangular geometry of height 1 (y coordinate) and length 2 (x coordinate) is chosen. The shear rate flow is imposed by fixing the velocity on the upper and lower boundaries:

$$u_x(x, 1) = .5 ; u_x(x, 0) = -.5 \text{ and } u_y(x, 0) = u_y(x, 1) = 0 \quad [25]$$

Moreover the following conditions are added on the vertical walls

$$u_y(0, y) = u_y(2, y) = 0 \text{ and } \partial_x u_x(0, y) = \partial_x u_x(2, y) = 0 \quad [26]$$

in order to have a Couette flow.

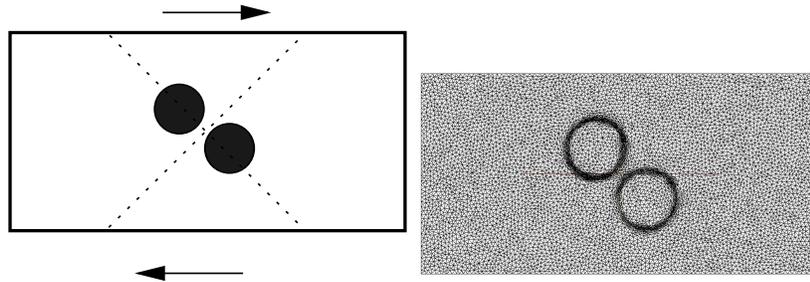


Figure 3. Computational geometry: (a) two particles in a shear flow; (b) the boundary-fitted mesh M_h

The two particles with the same radius (see Figure 3) are placed in the superior and inferior mid-plane and move toward the right and the left respectively. As they become closer, the hydrodynamic interactions between the two particles will prevent the collision.

5.1. Computation of rigidity constraint

First, a configuration with two symmetrically located particles is selected as a test problem. The two particles are close enough to check our ability to compute accurately hydrodynamic interaction and the influence of mesh and penalization factor on Uzawa algorithm. We consider a constant viscosity on each element, and therefore the mixing relation deals with a $P0$ interpolation for the characteristic function. This characteristic function can be obtained by voxelization procedure or a distance function (see Section 2).

Table 1. Mesh descriptions

Mesh	element number	node number	mean size
M_1	12 161	6 218	0.02
M_2	50 506	25 543	0.01
M_3	201 922	101 097	0.005
M_h	14 116	7 196	

The centers of two particles with the same radius 0.15 are located at point (0.873, 0.627) and (1.127, 0.373). First, the velocity is computed for mesh M_2 (see Table 1), $\eta_f = 1$ and various penalization factors η_s . The evolution of the rigidity constraint described by $\|\mathbb{I}_s \underline{\dot{e}}(\vec{u}_h^k)\|$ is plotted in Figure 4 for $\eta_s = 10, 10^2, 10^3, 10^4$ and up to 50 Uzawa iterations. It is found that:

- for each penalization factor, the norm of rigidity constraint decreases quickly until $k = 10$, then the reduction is slower. In fact, for $k \sim 10$ the norm of rigidity constraint is uniformly small inside the solid domain ($\mathbb{I}_s = 1$) whereas it remains more important in the transition area ($0 < \mathbb{I}_s < 1$). The next Uzawa iterations only decrease the norm inside this transition area,
- the rigidity constraint is more accurate as the penalization factor is bigger.

Finally, the velocity field is computed for all meshes described in Table 1. The three first meshes have a regular mesh size whereas the last one M_h is built from M_1 by increasing the number of elements between the two spheres (see Figure 3b). The method allowing to get this mesh is described in (Gruau *et al.*, 2005).

In Figures 5 and 6, we present comparison results for the velocity, the pressure and the normalized shear stress ($\|\underline{\dot{e}}(\vec{u}_h^k)\|$), respectively along two diagonals: the diagonal (/) without crossing particles (continuous phase) and diagonal (\) crossing the particles (discontinuous phase). The computations are made for a penalization factor $\eta_s = 1000\eta_f$ and 6 Uzawa iterations. The computations show uniform convergence with mesh refinement and the three meshes M_2, M_3 and M_h gives rather comparable results. More precisely, the Figures 5 point out:

- the rigidity condition is well satisfied inside the particles as $\|\underline{\dot{e}}(\vec{u})\| < 10^{-4}$,
- the pressure computation is less accurate between the two particles,

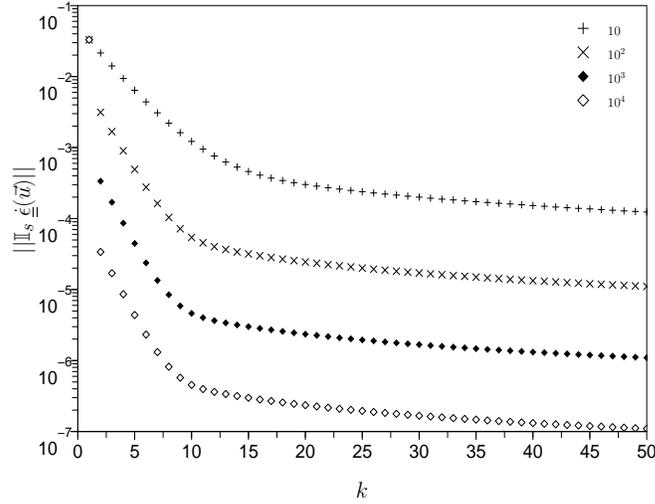


Figure 4. Influence of penalization factor η_s and number k of Uzawa loop on the rigidity condition $\|\mathbb{I}_s \dot{\underline{\epsilon}}(\bar{u}_h^k)\|$

- the computed velocity is less dependent on mesh size around particle center than near its boundaries,
- the size of the transition area between rigid and solid domains depends on the size of elements belonging to the fuzzy boundary.

Along the diagonal without crossing particles, one can see in Figures 6:

- the shear rate is more important in the area between the two particles,
- the vertical component of the velocity u_y depends on the mesh refinement.

5.2. Comparison of particle displacements

In this Section, we want to compare the Lagrangian displacement with the method using an advection equation. The first numerical procedure (which is similar to Particle-In-Cell method) is:

1. initialization $t = 0$: computation of the characteristic function \mathbb{I}_s with a voxelization method,
2. velocity solver: the velocity is computed (as detailed in Section 4),
3. explicit update of particle positions thanks to relation [24],
4. computation of new characteristic function \mathbb{I}_s by voxelization; return to step 2.

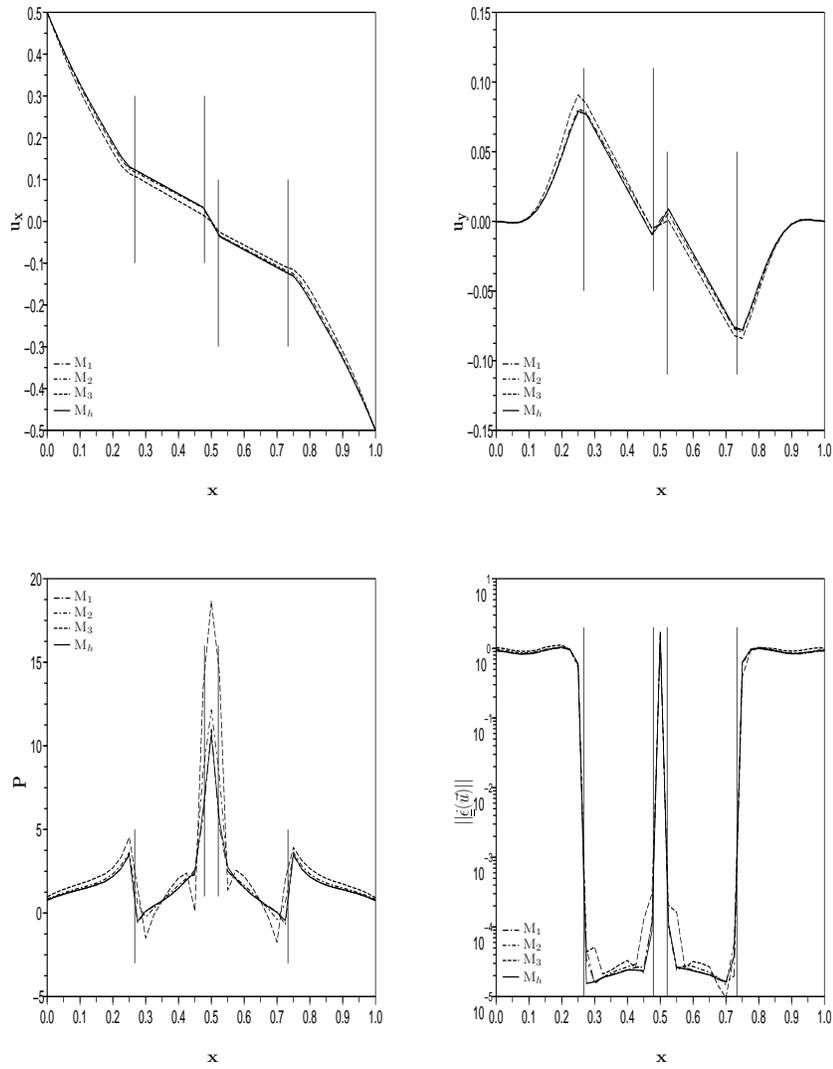


Figure 5. Comparison of the velocity, the pressure and total shear rate along the diagonal crossing the particles: influence of mesh refinement. The vertical line give the boundary of particles

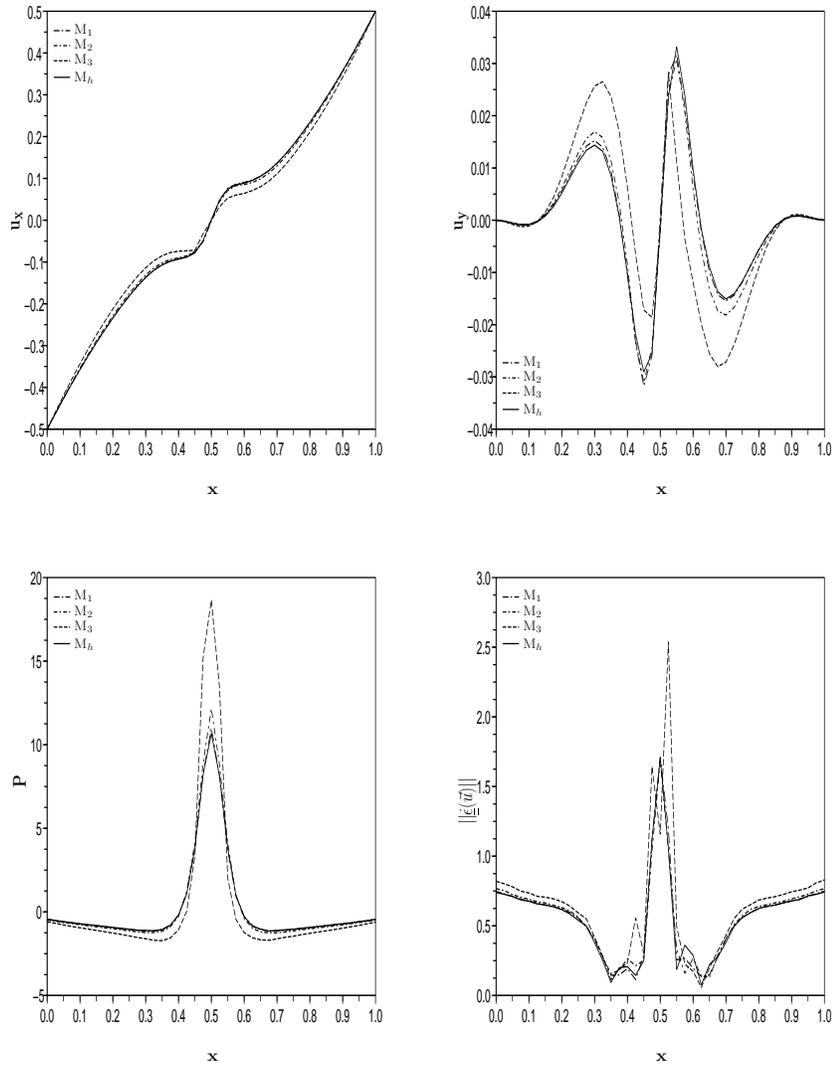


Figure 6. Comparison of the velocity, the pressure and total shear rate along the diagonal without crossing the particles: influence of mesh refinement

The second numerical procedure which uses only a finite element approach, reads:

1. initialization $t = 0$:
 computation of the level set function from the distance function,
 computation of $P0$ interpolation of \mathbb{I}_s from the level set function,
2. velocity solver: the velocity is computed as explained in Section 4,
3. displacement of the level set function with an advection equation: computation of $P0$ interpolation of \mathbb{I}_s from level set function; return to step 2.

The two particles with radius $r = 0.12$ are initially placed at points $(0.5, 0.6)$ and $(1.5, 0.4)$. The two particles move one towards the other, then they are pushed back to avoid a collision because of hydrodynamic repelling forces. Finally they come back to their initial vertical location. Orbits of the particles centers are plotted in Figure 7. An isolated sphere in a shear flow creates a small vortex because it turns on it (see Figure 8). However if they are close together, they move in a large movement of rotation to prevent a collision (see Figure 8).

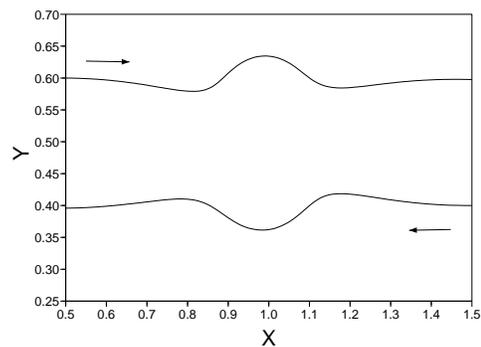


Figure 7. Orbits of particle centers: the particles are initially in $(.5,.6)$ and $(1.5, .4)$

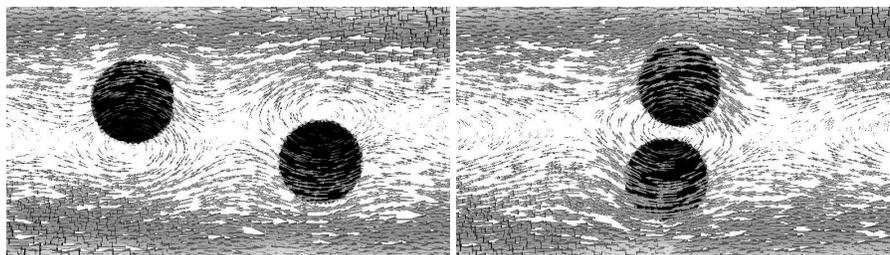


Figure 8. Focus on velocity field: $t = 2.5$; $t = 6.5$

One can compare the sphere displacements for the two methods by plotting the isovalues of level set function and characteristic function at various times. The time step used for both methods is $\Delta t = 0.05$. From Figures 9, one can see that:

- without interaction between particles, there is no difference between the two methods ($t = 2.5$),

- as the two particles become closer, the two methods do not coincide. In fact, the SUPG method needs an accurate computation of the velocity near the particle boundaries whereas with the Lagrangian method it is sufficient to compute accurately the velocity at particle center. Therefore without a better description of interface, the second method gives better results.

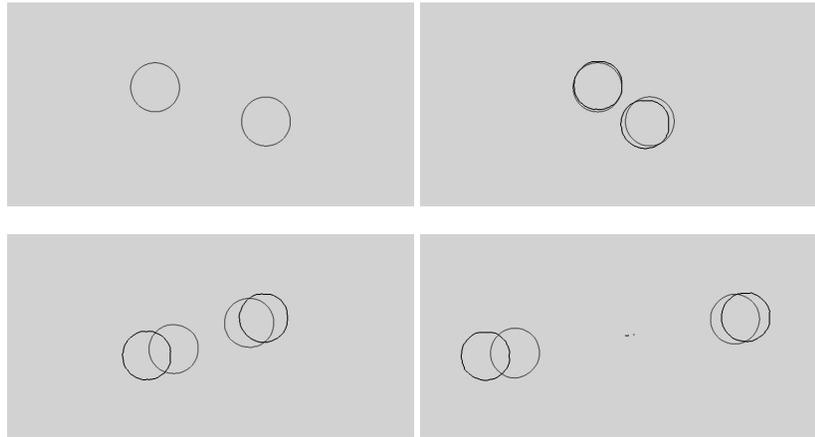


Figure 9. In gray zero isovalue of level set function moves with an SUPG method and in black .5 isovalue of characteristic function computed after an Lagrangian update of particle center: $t = 2.5$; $t = 5.$; $t = 10$; $t = 15$

6. Conclusion

We have proposed a strategy to simulate the motion of particles in a semi-dilute or concentrated suspension. The solid domain can be described by a characteristic function and the weak formulation uses Lagrangian multipliers to require a vanishing deformation rate inside particles. The Lagrangian multiplier is computed with an Uzawa algorithm and it is shown that this method gives rather good results.

Moreover, two ways for moving particles are compared. It is shown that the most efficient method consists to couple finite element computation of velocity to a Lagrangian method for particle displacement. This method is particularly well adapted

to compute the motion of several particles in lubrication regime of suspension flow (see Figure 10).

The possible improvements of this methodology could be:

- take a better integration scheme for Lagrangian update,
- use a $P1$ interpolation of characteristic function in the fluid motion solver. That can be do by updating the particle position with a distance function,
- propose a strategy to handle cases in which particles touch. A collision strategy is a method for preventing near collisions by defining a security zone around a particle. Instead of decreasing time step or refining mesh in the gap between particles, one can introduce an repelling force in the weak formulation or in the particulate method (Laure *et al.*, 2005).

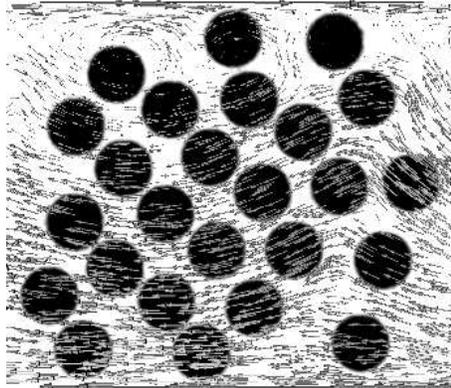


Figure 10. *The velocity for particles in a shear flow with horizontal boundary conditions*

Acknowledgements

The authors are grateful to H. Dignonnet and J. Bruchon for their helps in using the finite element library CIMlib.

7. References

- Arnold D., Brezzi F., Fortin M., “ Stable finite element for Stokes equations”, *Calcolo*, vol. 21, p. 337-344, 1984.
- Baines M., “ Grid adaptation via node movement”, *Applied Numerical Mathematics*, vol. 26, p. 77- 96, 1998.
- Bigot E., Simulation tridimensionnelle du remplissage de corps minces par injection, PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2001.

- Brooks A., Hughes T., "Streamline upwind/Petrov-Galerkin formulations for convectedominated flows with particular emphasis on incompressible Navier-Stokes equations", *Comp. Meth. in Appl. Mech. Engrg.*, vol. 32, p. 199-259, 1982.
- Fan X.-J., Phan-Thien N., Zheng R., "A direct simulation of fibre suspensions", *J. Non-Newtonian Fluid Mech.*, vol. 74, p. 113-1353, 1998.
- Fortin M., Brezzi F., *Mixed and Hybrid finite element method*, Springer Verlag, 1991.
- Franca L., Frey S., Hugues T., "Stabilized finite element methods: II The incompressible Navier-Stokes equations", *Comput. Methods Appl. Mech. Engrg.*, vol. 29, p. 209-233, 1992.
- Glowinski R., Pan T.-W., Helsa T., Joseph D., "A distributed Lagrange multiplier/fictitious domain method for particulate flows", *Int. J. Multiphase Flows*, vol. 25, p. 755-794, 1998.
- Gruau C., Génération de métriques pour adaptation anisotrope de maillages, applications à la mise en forme des matériaux, PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2004.
- Gruau C., Coupez T., "3D tetrahedral unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric", *Comput. Meth. Appl. Mech. Engrg.*, vol. 194, p. 4951-4976, 2005.
- Hwang W., Hulsen M., Meijer H., "Direct Simulation of particle suspensions in sliding bi-periodic frames", *Journal of Non-Newtonian Fluid Mechanics*, vol. 194, p. 742-772, 2004.
- Laure P., Megally A., Coupez T., "Collision strategy for the direct simulation of moving fibers in viscous fluid", *Int. Conf. on Computational Methods for coupled problems in science and engineering*, 2005.
- Megally A., Etude et modélisation de l'orientation de fibres longues dans des thermoplastiques renforcés, PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2005.
- Megally A., Laure P., Coupez T., "Direct Simulation of Rigid Fibers in Viscous Fluid", *3rd International Symposium on Two-phase flow modelling and Experimentation, Pisa*, 2004.
- Osher S., Sethian J., "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations", *J. of Computational Physics*, vol. 79, p. 12-69, 1988.
- Patankar N., Singh P., Joseph D., Glowinski R., Pan T.-W., "A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows", *Int. J. of Multiphase flow*, vol. 26, p. 1509-1524, 2000.
- Peng D., Mzerriman B., Osher S., Zhao H., Kang M., "A PDE-based fast local level set method", *J. Comp. Physics*, vol. 150, p. 410, 1999.
- Pichelin E., Coupez T., "Finite element solution of the 3D mold filling problem for viscous incompressible fluid", *Comput. Methods Appl. Mech. Engrg.*, vol. 163, p. 359-371, 1998.
- Pierre R., "Optimal selection of the bubble function in the stabilization of the P1-P1 element for the Stokes problem", *SIAM J. Numer. Anal.*, vol. 32, p. 1210-1224, 1995.
- Singh P., Joseph D., "Sedimentation of a sphere near a vertical wall in an Oldroyd-B fluid", *J. Non-Newtonian Fluid Mech.*, vol. 94, p. 179-203, 2000.
- Yamane Y., Kaneda Y., Dio M., "Numerical simulation of semi-dilute suspensions of rodlike particles in shear flow", *J. Non-Newtonian Fluid Mech.*, vol. 54, p. 405-421, 1994.

