A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies

Mathieu Coquerelle, Georges-Henri Cottet

A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies

M. Coquerelle[1,2] & G.-H. Cottet[1]

[1] Université de Grenoble and CNRS, Laboratoire Jean Kuntzmann, BP 53, 38041 Grenoble Cedex 9, France
[2]INRIA Grenoble - Rhône-Alpes, 38334 Saint Ismier Cedex, France

Abstract

We present a vortex method for the simulation of the interaction of an incompressible flow with rigid bodies. The method is based on a penalization technique where the system is considered as a single flow, subject to the Navier-Stokes equation with a penalization term that enforces continuity at the solid-fluid interface and rigid motion inside the solid. Level set functions are used to capture interfaces, compute rigid motions inside the solid bodies and model collisions between bodies. A vortex in cell algorithm is built on this method. Numerical comparisons with existing 3D methods on problems of sedimentation and collision of spheres are provided to illustrate the capabilities of the method.

1 Introduction

Vortex methods have long been used for the simulation of bluff-body flows. The success of these methods comes from the combination of several features, well illustrated in the pioneering work [18] for 2D cylinder wakes, as well in more recent 3D calculations [21, 11] : stability, localization of the computations in the regions of interest and natural treatment of far-field boundary conditions. Moreover when combined with fast grid solvers to compute velocity fields, and more generally all quantities that are not directly related to the transport of vorticity, these methods offer for external flows economical alternatives to more traditional grid-based Eulerian solvers.

In this paper we consider the extension of the Vortex In Cell method [7] to the case where several rigid bodies interact under the action of gravity with an incompressible flow. The difference with previously considered cases is that the rigid bodies move freely, and eventually collide, under the combined action of gravity and forces imparted by the flow. The numerical challenge in this problem lies in the coupling of stresses and velocities at the moving fluid/solid interfaces and in the modeling of collisions.

Let us first consider the fluid/solid interaction problem. This type of problem is often dealt with by ALE (for Arbitrary Lagrangian Eulerian) methods where the interface is tracked, flow and solid equations are solved separately, and continuity conditions for the velocity and stress are explicitly enforced at the interface (see [16] and the references therein). These methods are accurate but hard to implement in 3D and expensive, in particular if several objects interact in a non laminar flow. Here we consider a different approach. The fluid-solid is considered as a single flow. The interface is captured by a level set method and the rigid constraint together with the continuity conditions are implicitly recovered through a penalization method. When formulated in the primitive variables, this method

extends to the case of the two-way fluid-solid coupling the method of [1] which was introduced for the computation of bluff-body flows. The benefit that can be expected from this approach comes from the fact that fast grid solvers, based for instance on particle or finite-difference methods, can be used, leading to substantial computational savings, in particular for 3D cases, compared to body fitted methods.

Our method is related in particular to the fictitious domain approach of [14] and to the projection method of [20]. The fictitious domain approach of [14] was already derived as an alternative to ALE methods. The rigid motion inside the solid bodies is enforced through a Lagrange multiplier, which somehow plays the role of the pressure for the incompressibility constraint. The method is defined in a variational framework which is well suited for finite element discretization. A number of validations of this method have been performed and we will extensively use them to test our approach. A more recent work in the same spirit is the penalty method of [17]. In this work the rigid motion is also modeled in a variational framework leading to a minimization problem for the velocity. The functional to minimize in the space of rigid motions inside the body is approximated by a functional over all velocities with a penalization of the deformation inside the rigid bodies. In our method the interfaces are captured by advection equation which are connected to the flow equations only through the advection velocity. The added flexibility in the choice of the solvers makes the method fast and rather simple to implement, in particular when several bodies are considered.
The projection method of [20] consists of alternating Navier-Stokes solvers and projection steps where the rigid motion is recovered inside the body. This method is in the same spirit as the classical splitting method for the Navier-Stokes equations to enforce divergence free flows. As we will see it appears as a particular case of our penalization method, when a first order explicit time-discretization is chosen. Our approach is more general. In particular an implicit treatment of the penalization term allows larger penalization coefficients and thus more accurate results.

Besides providing a simple way to capture body boundaries and enforce rigid motions, an additional benefit of level set methods is to enable simple collision models to deal with contacts. Our starting point is a simple dynamical system with short range Hamiltonian forces to model single point collisions. The level set approach allows to generalize in a very straightforward way this system to more general collisions. Level set functions are used both to measure distance between interfaces and to localize the forces on the interfaces. The resulting collision forces appear just as an additional force in the single flow equation representing the fluid/solid system.

In this paper we focus on a vorticity formulation based on this approach which is appropriate for a particle discretization. We derive and illustrate particle-grid algorithms suitable for the simulation of the level set fluid-structure model.

This work is part of a series of papers where we systematically investigate multi-phase flow modeling and the associated algorithms for generic fluid-structure interaction problems. [2] is concerned with the numerical analysis of the underlying penalization method in primitive variables, and [6] deals with application of the present method in computer graphics. References [8, 9] deal with modeling and numerical issues of related methods for the case of elastic solids.

An outline of the paper is as follows. In section 2 we present our flow model. In section 3 we describe its particle discretization. In section 4, to validate our method and discuss its efficiency, we focus on test cases where quantitative comparisons with [14, 20] are possible, namely the sedimentation and collision of spheres. Finally section 5 is devoted to concluding remarks and perspectives.

2 Flow model

We first focus on the fluid/body interaction problem and then turn to the problem of collision between rigid bodies.

2.1 Interaction of an incompressible fluid with a rigid body

2.1.1 The physical problem

We consider a rigid solid $S(t)$ freely evolving in an incompressible flow contained in a domain K (the formulation readily extends to several bodies). We assume that the density is constant in the fluid and in the solid, with values respectively denoted by ρ_F and ρ_S. We denote by $F(t) = K - S(t)$ the fluid domain, by $\Sigma(t)$ the fluid-solid interface and by $\mathbf{n}$ its normal pointing towards the solid.

The fluid-solid interaction problem can be modeled by the Navier-Stokes equations in $F(t)$, continuity of velocity and stresses at the interface and the rigid motion of the solid $S(t)$:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u}\right) - \text{div}\,\mathbb{T}(\mathbf{u},p) = \rho\boldsymbol{g} \qquad \text{for } x \in F(t) \text{ and } t > 0 \tag{1}$$

$$\mathbf{u} = \mathbf{V} + \boldsymbol{\Omega}\times(\mathbf{x}-\mathbf{x}_G) \qquad \text{for } x \in S(t) \text{ and } t > 0 \tag{2}$$

$$M\dot{\mathbf{V}} = -\int_\Sigma \mathbb{T}(\mathbf{u},p)\mathbf{n}\,d\mathbf{x} + M\boldsymbol{g} \tag{3}$$

$$J\dot{\boldsymbol{\Omega}} = -\int_\Sigma \mathbb{T}(\mathbf{u},p)\times(\mathbf{x}-\mathbf{x}_G)\,d\mathbf{x}. \tag{4}$$

This system has to be complemented by initial conditions and boundary conditions on ∂K. In the above system $\boldsymbol{g}$ is the gravity, ρ is the variable density, $\mathbf{u}$ and p are the velocity and pressure, M is the solid mass, $\mathbf{x}_G$ its center of gravity and J its inertia matrix. The solid translation and angular velocities are respectively denoted by $\mathbf{V}$ and $\boldsymbol{\Omega}$. Equation (2) gives the general form of a rigid deformation. Equations (3) and (4) translate the solid acceleration as a result of gravity and fluid forces at the interface. The stress tensor $\mathbb{T}$ is defined as

$$\mathbb{T}_{ij}(\mathbf{u},p) = \nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - p\delta_{ij}$$

where ν is the fluid viscosity.

2.1.2 The penalization model

The idea is to extend the fluid velocity inside the solid body and to solve the flow equations with a penalization term to enforce rigid motion inside the solid. Given a penalization parameter $\lambda >> 1$, and denoting by χ_S the characteristic function of S, the model equation is the following :

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u}\right) - \text{div}\,\mathbb{T}(\mathbf{u},p) = \rho\,\boldsymbol{g} + \lambda\,\rho\,\chi_S(\overline{\mathbf{u}} - \mathbf{u}) \text{ for } x \in K \text{ and } t > 0 \tag{5}$$

coupled with the incompressibility condition

$$\text{div}\,\mathbf{u} = 0 \text{ for } x \in K \tag{6}$$

and the calculation of the rigid motion $\overline{\mathbf{u}}$, obtained by averaging translation and angular velocities over the solid :

$$\overline{\mathbf{u}} = \frac{1}{|S|}\int_K \chi_S \mathbf{u}\, d\mathbf{x} + \left(J^{-1} \int_K \chi_S \mathbf{u} \times (\mathbf{x} - \mathbf{x}_G)\, d\mathbf{x} \right) \times (\mathbf{x} - \mathbf{x}_G). \tag{7}$$

The rigid body follows the trajectories of the flow, with an advection velocity that can be chosen to be either the flow velocity $\mathbf{u}$ or the rigid motion $\overline{\mathbf{u}}$. As a result, its characteristic function can be obtained by solving the advection equation

$$\frac{\partial \chi_S}{\partial t} + (\overline{\mathbf{u}} \cdot \boldsymbol{\nabla})\chi_S = 0 \text{ for } x \in K \text{ and } t > 0. \tag{8}$$

Alternatively on may compute χ_S from a level set function ϕ_S satisfying the same advection equation

$$\frac{\partial \phi_S}{\partial t} + (\overline{\mathbf{u}} \cdot \boldsymbol{\nabla})\phi_S = 0 \text{ for } x \in K \text{ and } t > 0. \tag{9}$$

If ϕ is positive inside S and negative outside then $\chi_S = H(\phi_S)$ where H is the Heaviside function. It is customary to initialize ϕ_S as a signed distance function to the boundary of S. It is important to notice here that, because $\overline{\mathbf{u}}$ is a rigid body motion, one can guarantee that ϕ_S remains a signed distance for all time.

The density ρ is computed from the the fluid and solid densities and the function χ_S by

$$\rho = \rho_F(1 - \chi_S) + \rho_S \chi_S. \tag{10}$$

A few comments on the system (5)-(8) are in order. First in the case of a fixed obstacle with $\overline{\mathbf{u}}$ given, one can recognize the penalization method designed and analyzed in [1]. Next, the choice to follow the solid phase with the velocity $\overline{\mathbf{u}}$ instead of $\mathbf{u}$ (both choices are equivalent only in the limit $\lambda \to \infty$) maintains a strictly rigid motion of the solid, independently of numerical errors, a feature which is important in practice. Finally, a natural projection method for this system leads to the following algorithm : given a time step Δt, if we set $t_n = n\Delta t$, one goes from $\overline{\mathbf{u}}^n \approx \overline{\mathbf{u}}(\cdot, t_n)$ to $\overline{\mathbf{u}}^{n+1}$ through the following steps :

- solve the Navier-Stokes equation (5) with $\lambda = 0$ for a time step Δt; denote by $\widetilde{\mathbf{u}}^n$ the result of this iteration,
- compute $\overline{\mathbf{u}}^n$ from (7) after replacing $\mathbf{u}$ by the result $\widetilde{\mathbf{u}}^n$ of the previous step,
- solve

$$\frac{\partial \mathbf{u}}{\partial t} = \lambda \chi_S (\overline{\mathbf{u}} - \mathbf{u}) \tag{11}$$

  for a time step Δt.

Next we observe that for the particular choice $\lambda = 1/\Delta t$ and a Euler scheme to discretize (11), this step returns

$$\mathbf{u}^{n+1} = \begin{cases} \widetilde{\mathbf{u}}^n & \text{in } F(t) \\ \overline{\mathbf{u}}^n & \text{in } S(t). \end{cases} \tag{12}$$

This is essentially the method proposed by Patankar [20] to compute flow around rigid particles. There are however several advantages in considering penalization formulations. First this approach enables to consider penalization term in the diffusion term, in order to enforce more strongly rigid motion inside the solid. This option has not been considered in our numerical illustrations below but

it can prove useful to accelerate convergence. The penalization approach is also more tractable for numerical analysis (see [2]) and it allows to distinguish between modeling and discretization errors. A third and more important advantage is that the penalization equation can be discretized in time in an implicit fashion allowing larger penalization coefficients and therefore better accuracy in satisfying the interface boundary conditions. Indeed, explicit time discretization of (11) requires taking λ smaller than $1/\Delta t$. A consequence that we have observed in practice is that an accurate treatment of wall boundary conditions may thus require to take time-step values for this part of the algorithm much smaller than necessary for the particle treatment of the advection-diffusion. Alternatively, the implicit Euler time discretization of (11) reads:

$$\mathbf{u}^{n+1} = \frac{\mathbf{u}^n + \lambda \Delta t \chi_S \overline{\mathbf{u}}^n}{1 + \lambda \Delta t \chi_S} \tag{13}$$

This method is unconditionally stable. We will show in section 3.2 how this implicit formulation is implemented in our vortex method.

Finally, and this is important in the context of the present work, the penalization approach provides a natural vorticity formulation which would be less straightforward to derive in a projection approach. As a matter of fact, taking the vorticity formulation of (5) yields

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})\boldsymbol{\omega} = (\boldsymbol{\omega}\cdot\boldsymbol{\nabla})\mathbf{u} + \nu\Delta\boldsymbol{\omega} - \boldsymbol{\nabla}p\times\boldsymbol{\nabla}(\frac{1}{\rho}) + \lambda\boldsymbol{\nabla}\times\chi_S(\overline{\mathbf{u}}-\mathbf{u}). \tag{14}$$

This system has to be complemented by the usual system giving the velocity in terms of the vorticity :

$$\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \text{ in } K; \boldsymbol{\nabla}\times\mathbf{u} = \boldsymbol{\omega} \text{ in } K \tag{15}$$

and appropriate boundary conditions on ∂K. The rigid velocity $\overline{\mathbf{u}}$ is computed from (7). Ignoring for a moment the barotropic term and the rigid motion coupling, when combined with a vortex method, one may recognize in the above equation an immersed boundary method in the spirit of the methods designed in [21, 11, 12]. We will comment later on this point.

We now continue with the derivation of the model. Developing the term $\boldsymbol{\nabla}\times\chi_S(\overline{\mathbf{u}}-\mathbf{u})$ one obtains

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})\boldsymbol{\omega} - (\boldsymbol{\omega}\cdot\boldsymbol{\nabla})\mathbf{u} - \nu\Delta\boldsymbol{\omega} = -\boldsymbol{\nabla}p\times\boldsymbol{\nabla}(\frac{1}{\rho}) + \lambda\chi_S(\overline{\boldsymbol{\omega}}-\boldsymbol{\omega}) + \lambda\delta_\Sigma\,\mathbf{n}\times(\overline{\mathbf{u}}-\mathbf{u}). \tag{16}$$

In the above equation we have set $\overline{\boldsymbol{\omega}} = \boldsymbol{\nabla}\times\overline{\mathbf{u}}$ and we recall that $\mathbf{n}$ is the normal to the interface Σ oriented towards the solid. It is interesting to note that the right hand side of this equation contains two vorticity generators : one in the barotropic term related to the underlying variable density flow, and one coming from the no-slip condition at the fluid-solid interface. Both terms are singular and localized at the interface. Note that the second vorticity production term, which is reminiscent to vorticity creation algorithms often used in conjunction with vortex methods, would not directly result from a projection method on the vorticity flow equation.

Both (14) or (16) require to compute the pressure, something which is in general not necessary in vorticity formulation of the Navier-Stokes equations. To avoid explicitly computing the pressure one may alternatively rewrite the vorticity equation in the following way :

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})\boldsymbol{\omega} = (\boldsymbol{\omega}\cdot\boldsymbol{\nabla})\mathbf{u} + \nu\Delta\boldsymbol{\omega} - \frac{\boldsymbol{\nabla}\rho}{\rho}\times\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u} - \boldsymbol{g}\right) + \lambda\boldsymbol{\nabla}\times\left(\chi_S(\overline{\mathbf{u}}-\mathbf{u})\right) \tag{17}$$

The system to solve in our model consists finally of the equations (14) or (16) or (17) together with (15), (7), (8) (or (9)). In all our calculations we will follow the above formulation. However one may

also consider the implementing a simplified model based on a Boussinesq approximation of variable density flows. In this approximation, the density is supposed to be almost constant. If one still denotes by ρ the density variations around a constant state, and $\mathbf{u}, \boldsymbol{\omega}$ the velocity and vorticity created by this density variation, an expansion around a steady-state of the system gives, up to lower order terms, the following vorticity equation:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \boldsymbol{\nabla})\boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \boldsymbol{\nabla})\mathbf{u} + \nu \Delta \boldsymbol{\omega} + \boldsymbol{\nabla}\rho \times \boldsymbol{g} + \lambda \boldsymbol{\nabla} \times \chi_S(\overline{\mathbf{u}} - \mathbf{u}). \tag{18}$$

2.2 Collision model

We now consider the case of N rigid bodies S_i with densities ρ_i interacting in an incompressible fluid and we design a method to handle collisions between the bodies. Each body is followed through its level set function ϕ_i. Each level set function is moved with the rigid motion associated to the corresponding rigid body. In other words we consider N copies of equations (7), (8), (9), with S replaced by S_i. The density is given by

$$\rho = \rho_F(1 - \sum_{i=1}^{N} \chi_i) + \sum_{i=1}^{N} \rho_i \chi_i, \text{ where } \chi_i = H(\phi_i). \tag{19}$$

When dealing with collision of solids in incompressible fluids, one faces a dilemma. On the one hand, experiments show that collisions occur between solids in a fluid and lead to rebounds. On the other hand, as noted in [14] and confirmed by recent theoretical works, the Navier-Stokes system with no-slip boundary conditions is not compatible with contact between rigid bodies. In any case, collision models appear to be necessary at least to prevent numerical error to result in overlapping of solid bodies. Such collision models should be energy conserving. For the case of bodies imbedded in a viscous flow, energy dissipation at collision time should only result from viscous dissipation in the boundary layer. Here we consider a direct approach that will be again rely on interface capturing by means of level set functions. Using level set functions to deal with collisions is actually not new. These techniques were proposed under the names of implicit surfaces in physical modeling for the animation of elastic or rigid bodies (see for instance [4]). In these methods, body surfaces are captured by implicit surfaces which are used to detect penetration. Distance functions serve to compute the amount of overlapping. This information is used to derive forces on the surfaces that are subsequently propagated throughout the bodies in some *ad-hoc* fashion in order to remove this overlapping. Our method differs from these implicit surfaces techniques in particular in that we compute a single force which is essentially parameter free and simply supplements the flow equation. The level set collision formulation is only a way to distribute in a clear-cut way repulsive point forces over the surface of the bodies.

2.2.1 One-dimensional model

Like in [14], we start from a dynamical system for the one-dimensional collision of a material point located at $x(t) > 0$ with a wall assumed to be located at $x = 0$. We choose a dynamical system which is both short range and singular at the contact point. This dynamical system reads

$$\ddot{x} = \frac{\kappa}{x} \exp\left(-x/\epsilon\right) \tag{20}$$

where κ is a coefficient which has the dimension of the square of a velocity. It is an Hamiltonian model acting on a width of the order of ϵ around $x = 0$. As a result, it produces an energy preserving

collision at a distance $O(\epsilon)$ of the wall. More precisely, its potential energy is, up to an additive constant, given by

$$E(x) = \int_1^x \frac{\kappa}{s} \exp(-s/\epsilon), ds = \int_{1/\epsilon}^{x/\epsilon} \frac{\kappa}{y} \exp(-y)\, dy.$$

Consider a point initially located at $x(0) = 1$ and with velocity $v(0) = v_0 < 0$. If we denote by $x^\star$ its minimal location (a point at which the velocity is zero), by conservation of energy we have

$$x^\star = \epsilon\, G_\epsilon(v_0^2/2\kappa) \simeq \epsilon\, G(v_0^2/2\kappa) \text{ for } \epsilon \ll 1, \tag{21}$$

where $G\epsilon$ and G denote the inverse functions respectively of the functions $F\epsilon$ and F defined as

$$F_\epsilon(x) = \int_x^{1/\epsilon} \frac{1}{y} \exp(-y)\, dy\ ,\ F(x) = \int_x^{+\infty} \frac{1}{y} \exp(-y)\, dy.$$

Equation (21) states that the rebound width scales as ϵ.

2.2.2 Level set model

The level set formulation allows to translate the simple dynamical system above into a body force distributed on the surface. Level set functions indeed give simultaneously access to distances, normals and surface integrals on the interfaces. If the rigid bodies are labelled by indices i, to compute the rebound of S_i and S_j, imbedded in a fluid with fluid interfaces respectively captured by level set functions ϕ_i and ϕ_j, point forces similar to the right hand side of (20) are implemented on each point of the interface of S_i, localized through the level set function ϕ_i. The direction of the force and the distance of the boundary points of S_i to the body S_j are obtained directly from the level set function ϕ_j. From these observations, one readily obtains the following force resulting from all possible body interactions :

$$\mathbf{f}_{\text{col}}(\mathbf{x}) = -\rho \sum_{ij} \frac{\kappa_{ij}}{\epsilon} \zeta\left(\frac{\phi_i(\mathbf{x})}{\epsilon}\right) \frac{\boldsymbol{\nabla}\phi_j(\mathbf{x})}{\phi_j(\mathbf{x})} \exp\left(-\phi_j(\mathbf{x})/\epsilon\right). \tag{22}$$

The minus sign above comes from the fact that, by the definition of the level set functions, $\boldsymbol{\nabla}\phi_j$ is oriented inwards with respect to the body S_j. The function ζ in the above expression is a one-dimensional cut-off function. In view of (21), the coefficients κ_{ij} are to be chosen proportional to the square of the relative velocities of the corresponding bodies just before the collision. Note that in the above equation, we have used the fact already mentioned that since the body/fluid interface level set functions are transported by rigid body velocity, they remain distance functions at all time (in the discretization described below, this property will also be satisfied at the discrete level). The force given by (22) supplements density driven and penalization forces in the right hand side of (5). In a vorticity formulation, the right hand side of equation (14) is complemented by the term $(\boldsymbol{\nabla} \times \mathbf{f}_{\text{col}})/\rho$. If we denote by $\overline{\mathbf{u}}_i$ the rigid motion velocity of the solid S_i, obtained by replacing S by S_i in (7), and set $\chi_i = \chi_{S_i}$ we obtain :

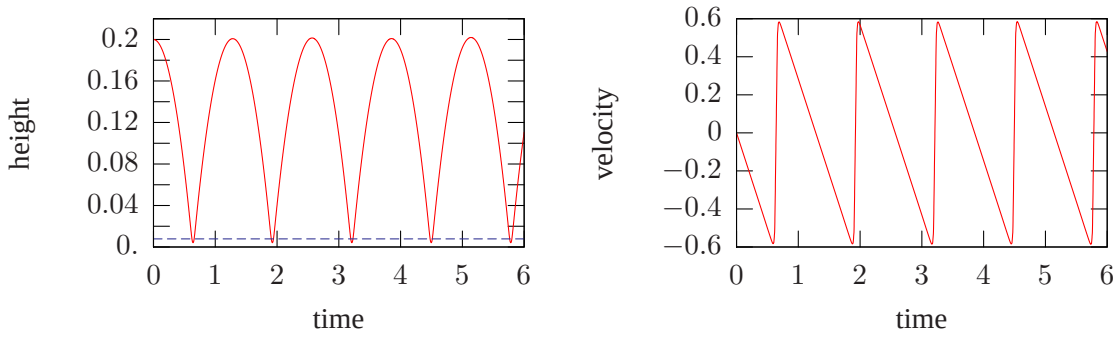

Figure 1: Level set collision model (22) for a 2D cylinder falling in vacuum on a flat plane. Positions (left picture) and velocity (right picture) as a function of time. Sampling is done at time intervals equal to $\epsilon/10$ and $\epsilon = h = 1/128$. The dotted horizontal line in the right picture is at $y = y_{\texttt{wall}} + h$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})\boldsymbol{\omega} - (\boldsymbol{\omega}\cdot\boldsymbol{\nabla})\mathbf{u} = \tag{23}$$
$$\nu\Delta\boldsymbol{\omega} - \frac{\boldsymbol{\nabla}\rho}{\rho}\times\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u} - \boldsymbol{g}\right) + \lambda\boldsymbol{\nabla}\times\left(\sum_i \chi_i(\overline{\mathbf{u}}_i - \mathbf{u})\right) + \frac{\boldsymbol{\nabla}\times\mathbf{f}_{\text{col}}}{\rho}.$$

To illustrate this model, we consider in Figure 1 the case of a 2D cylinder falling on a flat plane. The solid body has initially its center at $y = 0.3$ and is falling under the action of gravity, with $g = -1$. The action of the flow is neglected so the only forces are gravity force and the collision force (22). In this equation the level set function associated to the flat plane is the y-coordinate. The numerical parameters are $\epsilon = h = 1/128$ and $\Delta t = h/10$, where h is the grid size on which level set functions are captured. One can observe that the balls rebounds without dissipation. As a matter of fact, the ball elevation after the first rebound is slightly higher than the initial elevation. This is due to the fact that we used an Euler time explicit discretization that does not guarantee energy control. A more involved implicit discretization, which would prevent energy increase, has not been considered, as in the flow/body interaction problems below the Euler discretization was found to give satisfactory results. A first conclusion is that the equation (22) provides a reasonable energy preserving collision model which is essentially parameter free if we seek to prevent contact beyond a scale of the order of the grid spacing.

3 Discretization method

To discretize the system (23), (15), (7), (8) we combine a level set method with an hybrid particle-grid method. The fluid-solid system is assumed to be located inside a computational box Ω in which we consider a uniform fixed grid with grid size h. This grid is used in particular to initialize and remesh vorticity particles and to compute velocity and level set functions. We denote by Δt the time-step and we set $t_n = n\Delta t$ for $n \geq 0$ and ϕ_i^n, $\mathbf{u}^n$, $\boldsymbol{\omega}^n$ grid values of the level set functions, velocity and vorticity fields at time t_n. We assume that these quantities are known up to time t_n and we describe how to compute them at time level t_{n+1}. We first focus on the level set equation and then turn to the vorticity equation.

3.1 Level set equations

Each body-fluid interface is captured by a level set function. These level set functions are advected by a rigid body motion, associated to the rigid body under consideration and computed from the known values of ϕ_i et $\mathbf{u}$ on the grid. This enables an analytic calculation of the the level set functions. For simplicity let us focus on an individual level set function ϕ satisfying the following equation

$$\frac{\partial \phi}{\partial t} + (\overline{\mathbf{u}} \cdot \boldsymbol{\nabla})\phi = 0 \text{ for } t > 0;\ \phi(\cdot, 0) = \phi_0. \tag{24}$$

Let us denote by $\mathbf{X}(t; \mathbf{x}, 0)$ and $\mathbf{X}(0; \mathbf{x}, t)$ respectively the forward and backward characteristics associated to the flow field $\overline{\mathbf{u}}$. One has

$$\phi(\mathbf{x}, t) = \phi_0(\mathbf{X}(0, \mathbf{x}, t)). \tag{25}$$

Let us fix $\mathbf{x}$ and set $\mathbf{X}_n = \mathbf{X}(t_n; \mathbf{x}, 0)$. Assuming that the rigid motion velocity $\overline{\mathbf{u}}$ takes a constant value (with respect to time) in the time interval $[t_n, t_{n+1}]$ one may deduce $\mathbf{X}_{n+1}$ from $\mathbf{X}_n$ by rotation and translation. Given the following parameters

$$\theta^n = |\overline{\boldsymbol{\omega}}^n| \Delta t\ ,\ \frac{\overline{\boldsymbol{\omega}}^n}{|\overline{\boldsymbol{\omega}}^n|} = (\alpha, \beta, \gamma)$$

we consider the rotation around the mass center $\mathbf{c}_n$ of S at time t_n and matrix

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

where

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

With these notations, the rigid motion of S is given by

$$\mathbf{X}^{n+1} = \mathbf{c}^n + \overline{\mathbf{u}}^n \Delta t + \mathcal{R}^n(\mathbf{X}^n - \mathbf{c}^n).$$

By induction we can thus compute matrices $\mathcal{M}^n$ and vectors $\mathbf{V}^n$, independent of $\mathbf{x}$, such that

$$\mathbf{X}^{n+1} = \mathcal{M}^{n+1}\mathbf{X}^0 + \mathbf{V}^{n+1}.$$

From the above formula one can therefore analytically compute $\mathbf{X}^0 = \mathbf{x}$ from $\mathbf{X}^{n+1}$. The numerical computation of ϕ^{n+1} on the grid thus goes along the following lines :

- for all grid points $\mathbf{x}_i$, compute analytically $\mathbf{X}(0; \mathbf{x}_i, t^{n+1})$ with the above procedure,
- interpolate $\phi_0(\mathbf{X}(0; \mathbf{x}_i, t^{n+1})$ from grid values of ϕ_0 by bilinear interpolation.

Note that this algorithm, unlike a solution based on the numerical discretization of the underlying PDE, avoids to accumulate numerical errors which might eventually deform the rigid body. It also involves negligible computational effort, since it amounts to computing at every time step a single 3 by 3 matrix. This is a very desirable feature if one wishes to simulate a large number of interacting bodies.

3.2 Vorticity equation

We now come to the vorticity equation (23). To simplify the notations we consider the case of a single body. We split this equation and successively solve in the time interval $[t_n, t_{n+1}]$:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \frac{\boldsymbol{\nabla} \times \mathbf{f}_{\text{col}}}{\rho} \tag{26}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \lambda \boldsymbol{\nabla} \times (\chi_S(\overline{\mathbf{u}} - \mathbf{u})) \tag{27}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = -\frac{\boldsymbol{\nabla}\rho}{\rho} \times \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u} - \boldsymbol{g} \right) \tag{28}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \boldsymbol{\nabla})\, \boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \boldsymbol{\nabla})\mathbf{u} + \nu \Delta \boldsymbol{\omega} \tag{29}$$

The velocity $\overline{\mathbf{u}}$ needed in equation (27) is evaluated with the formula (7). In principle the velocity $\mathbf{u}$ in the right hand side of (7) should be computed from the vorticity resulting from one time-step of the equation (26). In practice we directly compute it by adding to the average velocity of the previous time-step the average velocity of the collision force in its primitive form. This means that we actually by-pass the resolution of (26) and directly average the collision force (22) on the body. This has the advantage of avoiding the numerical differentiation of a singular force, and the associated possible discretization error. Note that this also means that we omit the possible contribution of the collision force in the flow outside the body, or, equivalently, that we use a one-sided cut-off function in (22). To solve (27) we use a vorticity form of the implicit scheme (13) and we set

$$\widetilde{\boldsymbol{\omega}}^{n+1} = \boldsymbol{\nabla} \times \left[\frac{u^n + \lambda \Delta t \chi_S \overline{\mathbf{u}}^n}{1 + \lambda \Delta t \chi_S} \right].$$

The right hand side above is evaluated by centered finite differences.
To solve the equation (28), density values are obtained from (19) and differentiated on the grid ; we use grid values of $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$ to compute $\partial \mathbf{u}/\partial t$ and centered finite differences to estimate $(\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u}$, $\boldsymbol{\nabla} \times (\chi_S(\overline{\mathbf{u}} - \mathbf{u}))$ at all grid points at time t_n. Vorticity grid values are updated from $\widetilde{\boldsymbol{\omega}}^{n+1}$ using these approximations.

At this point, grid vorticity above a certain cut-off is used to create particle at grid point locations and equation (29) is solved by a classical vortex-in-cell method [10]. Particles are pushed with a RK2 time-stepping. Particles are then remeshed on the original grid using the following third order interpolation kernel

$$M_4'(x) = \begin{cases} 0 & \text{if } |x| > 2 \\ \frac{1}{2}(2 - |x|)^2(1 - |x|) & \text{if } 1 \le |x| \le 2 \\ 1 - \frac{5x^2}{2} + \frac{3|x|^3}{2} & \text{if } |x| \le 1 \end{cases} \tag{30}$$

Finally diffusion is solved through a implicit solver on the grid, with a classical 7-points second order scheme. Note that the same kernel is used to interpolate grid velocity values onto particles in the RK2 particle pusher. It remains now to indicate how we compute grid velocity values. In all the examples below we will consider internal flows in domains that can be put in a rectangular box. Since we use an immersed boundary approach, including for the external walls, we can choose any type of boundary conditions for the velocity at the boundary of this computational box. The wall boundary conditions will eventually be corrected by the penalization method. For maximum efficiency it is natural to use periodic boundary conditions and FFT-based velocity evaluations on the grid. For external flows it

would be more appropriate to use fast Poisson solvers with Dirichlet boundary conditions related to the flow value at infinity for the stream function at the outer boundary. Grid values for vorticity, velocity and level set functions are now available for time t_{n+1} and a new cycle of iterations can start.

It is worthwhile to point out that, in case we consider a rigid solid with a prescribed motion, we obtain an immersed boundary vortex method. This type of method has been for instance considered in [21, 11, 12]. Note however that, unlike in these references, the vorticity penalization approach avoids to deal separately with the vorticity forcing and the boundary potential (or vortex sheet) in the calculations of the velocity. The penalization approach allows to cancel at once, up to truncation errors, both normal and tangential velocity components. Boundary conditions for both the normal and tangential components can be simultaneously imposed because the penalization method tends to cancel the vorticity inside the solid. This clearly simplifies the implementation and reduces the computational cost. Another important difference is that the method is by definition conservative with respect to the circulation, whereas in previous works, vorticity inside the obstacle is generally discarded which may require ad hoc circulation corrections in the vicinity of the obstacles [12]. A validation study of the present method in the case of a flow past a fixed cylinder is reported in [19].

4 Numerical illustrations

In this section we consider numerical illustrations of the method just described on test cases that have been considered in [14] and [20]. These cases deal with the sedimentation and interaction of spheres. Throughout this section we denote by h the size of the grid on which particles are initialized and remeshed and on which velocity fields and level set functions are evaluated. In all cases we have taken the collision parameter ϵ equal to h and we have set the penalization parameter λ to 10^6.

4.1 Sedimentation and rebound of a 2D cylinder on a flat plane

We consider the case of a 2D cylinder in a square cavity, falling under gravity on a flat plane. This test serves as a way to validate both the force calculations and the contact model. We use parameters that have been considered in [14]. The dimension of the cavity is $[0, 2] \times [0, 6]$. The viscosity is 0.01. The density inside and outside the cylinder is 1.5 and 1. The cylinder has radius 0.125 and is initially located at the point $(1, 4)$. It is accelerating under gravity, set to $g = -980$, then settles to a steady velocity, due to the friction forces, and eventually hit the bottom of the cavity and stops. We show in Figure 2 the values of the vertical velocities, for $h = 1/256$ and $h = 1/512$, in an horizontal cut of the computational domain though the center of the cylinder at time $t = 1$. This figure shows that the rigid motion is not exactly satisfied inside the cylinder. There is actually for $h = 1/512$ a maximum relative discrepancy of about 2% in the vertical velocity. This discrepancy is small but would be enough to eventually deform the body if the level set function was advected by $\mathbf{u}$ instead of $\overline{\mathbf{u}}$. This figure also illustrates that we do not discard vorticity inside the body : in the present case, a small dipole survives inside the body. Figure 3 demonstrates that the deviation of the flow velocity from a rigid body motion has negligible impact on the dynamic of the body. For the same grid-resolution and time-steps our results match almost perfectly those of [14]. Although in principle, the collision parameters κ_{ij} should be automatically adjusted to the velocities before the contact, in this experiment as in the following ones, we have chosen to specify it by hand. The collision parameters $\kappa_{12} = \kappa_{21}$ were set to 30. We actually observed that the results, in particular the amplitude of the rebound when the cylinder hits the wall, are not very sensitive to this value (for instance multiplying this coefficient by a factor 2 has almost no effect). However taking a too small value (in this experiment a value smaller than 10) can lead to undesirable oscillations after the contact. This experiment both validates

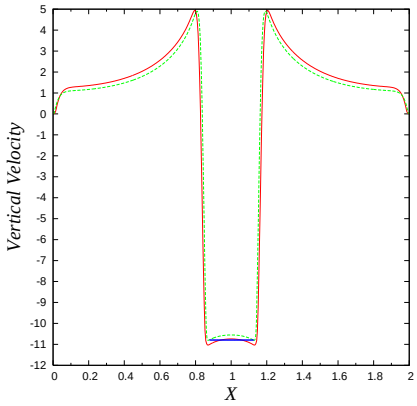

Figure 2: 2D cylinder falling in a square cavity (parameters given in section 4.1) : vertical velocity on an horizontal cut through the center of the cylinder at time $t = 1$. Solid line : $h = 1/512$; dashed line : $h = 1/256$; dotted line : value of $\overline{\mathbf{u}}$ obtained for $h = 1/512$.

the penalization approach for the two-way fluid-body coupling and shows that our collision model behaves like the one in [14]. We will come back to this point in section 4.2.3 below.

4.2 Three-dimensional cases

We consider here three-dimensional cases which are well documented in the literature, namely the sedimentation of a sphere and the tumbling and kissing of two spheres. To illustrate our collision model in absence of symmetry we also consider the less conventional case of an ellipsoid colliding with a sphere.

4.2.1 Sedimentation of a sphere

We consider here the 3D counterpart of the problem seen in section 4.1. We study the settling velocity of a sphere in an infinitely deep channel. This problem has been studied both experimentally and numerically. Experiments have been done to measure the terminal velocity of settling spheres falling in cylindrical channels and particularly to study the effects of boundaries on it. Brown and Lawler [3] give the corrections to be applied to account for wall effects. The smaller the ratio $\alpha = d/L$ between the channel size L and the sphere diameter d is, the more the walls affect the sphere and slow it down. Glowinski et al. [14] have computed terminal velocities with varying fluid viscosity, sphere diameter and density. We use the same geometry and physical parameters. The computational box is $[0, 1] \times [0, 1] \times [0, 4]$. No slip boundary conditions are imposed, using the penalization method, on the surface of a vertical cylinder of radius 1 inside this box. The fluid density is 1 and $g = -980$. We first perform a convergence study for our method in the case when the sphere density is 1.02. Figure 4 shows the terminal velocities obtained at $t = 1.2$ for various grid sizes. This experiment indicates that our method is roughly first order as far as the rigid motion velocity is concerned.
We show in table 1 a comparison between our results (in the limit when $h \to 0$) and experimental results for a sphere diameter $d = 0.2, 0.3$, fluid kinematic viscosity of $\nu = 0.02, 0.05, 0.1$. Our results agree with the experiments within an accuracy of 5 to 11% depending on the cases. A possible reason for the discrepancy between the experimental results and our numerical results may be that our simulation runs in a closed box where no fluid is escaping at the top nor the bottom. For the same

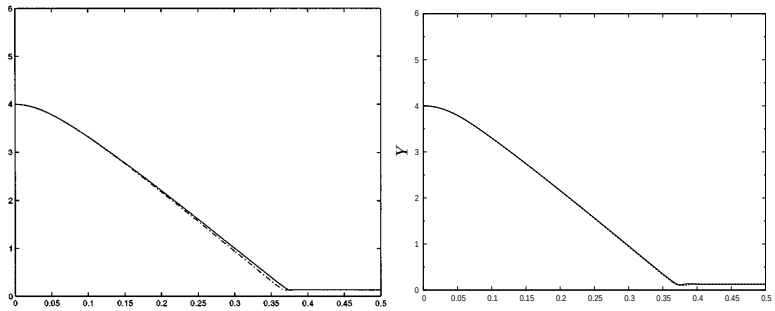

Figure 3: Sedimentation of a 2D cylinder falling in a square cavity under gravity and hitting the bottom wall. Comparison of our results (right pictures) with those in [14] (left pictures). Top pictures : height of the center of the cylinder. Bottom pictures : vertical velocity. In all pictures the solid lines are for ($h = 1/256$, $\Delta t = 7.5\,10^{-4}$), and the dashed lines for ($h = 1/384$, $\Delta t = 5\,10^{-4}$). Physical parameters are given in section 4.1.

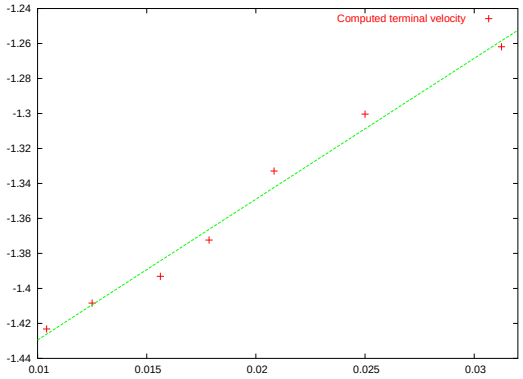

Figure 4: Terminal velocities as a function of grid spacing of a settling sphere of diameter 0.3 and density 1.02 falling in a fluid of density 1 and viscosity 0.02.

diameter	viscosity	U_E	$U_{h\to 0}$	Relative error	$U_{h=1/64}$	U_G
0.2	0.10	0.2571	0.275	8%	0.256	0.2567
0.2	0.05	0.4603	0.513	11%	0.475	0.4844
0.2	0.02	0.9129	1.016	11%	0.937	0.9480
0.3	0.10	0.4047	0.435	7%	0.401	0.4072
0.3	0.05	0.7493	0.795	6%	0.748	0.7599
0.3	0.02	1.4359	1.51	5%	1.39	1.392

Table 1: Comparison between terminal velocities U for spheres falling in a fluid for various diameters and viscosities. U_E are the experimental results. The relative error is computed between experimental results and our asymptotic results. U_G and $U_{h=1/64}$ are respectively the results in [14] and our results for $h = 1/64$.

experiments [14] report values obtained with a grid size $h = 1/64$. For a fair comparison we have indicated these results together with our (non converged) results for the same grid resolution. The results agree very well.

4.2.2 Kissing and tumbling of spheres

We next turn to more complex 3D dynamics : we study the sedimentation of two balls in a Newtonian fluid. The balls are aligned on the axis of gravity, one being at a distance of the order of its diameter above the other. While falling, the sphere on top experiences less friction with the fluid as it stands in the wake of the bottom one. Its velocity becomes greater than the second one, and at some point the balls are bound to collide with each other leading to the "kissing". The aligned configuration of the balls falling is known to be unstable and they eventually leave the axis and tumble aside from each other. This phenomena called "Drafting, kissing and tumbling" has been observed and studied experimentally in [13]. Numerical simulations are reported in [14, 20] an we use the same parameters as in these references.

The computational box is a rectangular channel of size $[0,1] \times [0,1] \times [0,4]$. The fluid has density 1 and a kinematic viscosity $\nu = 0.01$; the balls have radius 0.083 and density 1.14. They are placed initially at the center of the (x,y) cross section at $z = 3.5$ and $z = 3.16$ respectively. In order to initiate the instability in the x and y directions, we slightly shifted (by one grid size) the location of

the bottom ball. The spatial discretization is $h = 1/60$. For that case, the time-step value is adjusted in function of the maximum vorticity value, as it is commonly done in vortex methods. Our time-step value was initialized at 0.07 and then was given by $\Delta t = 0.8/|\boldsymbol{\omega}|_\infty$ As a result of vorticity production in the wake of the cylinders it decreased to reach a value 0.0054 at $t = 0.5$ and settle around this value later on. As usual the collision width ϵ was equal to h and the corresponding amplitude coefficients were $\kappa_{12} = \kappa_{21} = 25$. The gravity was set to $g = -980$.
The velocity of the solids are plotted in figure 5. They are in good agreement with the results obtained by Glowinski et al. [14]. The fact that in our case the top ball accelerates a bit more just before $t = 0.3$, may be due to the fact that the instability, which causes the balls to separate in a diagonal plane, is triggered faster in [14]. This is visible in the horizontal velocity plot in figure 6. In our case the instability is triggered by the contact, while in [14] it happens earlier. As a result, in our simulation the ball on the top is sucked in the wake of the bottom one for a longer time and slips more slowly along its edge. For the same reason, the contact between the balls between $t = 0.3$ and $t = 0.4$, when their velocities are almost equal, lasts longer in our case. It is interesting to note that our results as well as the results of [14], depart quite a bit just before and after the contact from the results of Sharma et Patankar [20], although our penalization method is closer to their method than to the method of Glowinski et al. In figure 7 we show the distance between the two balls against time : from $t = 0.3$ to $t = 0.4$ the minimum distance using in our method is approximately equal to h.
Let us now comment on the computational complexity of the methods. For $h = 1/60$, [14] reports a CPU time of 120 seconds per iteration on a DEC 500 Mhz workstation (there is no reference to computational time in [20]). In our case the cost of one time-step, on a Dell Precision M70 running an Intel Centrino 2GHz processor with 1GB of RAM memory, varies, due to the increasing number of particles from 5,000 to about 400,000, from 9 seconds, at the early stage of the simulation, to 18 seconds at the end. Given the relative performance of these processors one may deduce that the complexity of the methods are similar, at least for the flow under consideration. A difference in the computational cost of the methods comes however from the robustness of the vortex methods and the large time-steps that they allow. The time-step used in [14] was 10^{-3}, a value from 5 to 7 times smaller than our time-step.

4.2.3 Collision model in cases with or without symmetry

We come back to our level set collision model. We have already observed in section 4.1 that it performed like that of [14] in the case of a 2D cylinder hitting a wall. To confirm this observation in the more complex case of kissing spheres just considered, we show in Figure 8 a comparison of the velocity of the spheres obtained by our vortex method complement either by our collision model or the collision model of [14]. To implement the collision model of [14] in our vortex penalization method , we took, with the notations of this reference, $\rho = 2/60$, $\epsilon = 10^{-4}$ and $c_{ij} = 3.33\ 10^{-6}$. The two models give roughly the same response.

The reason why the two collision models give results which are so close lies in the symmetry of the problem. The model used in [14], and which is similar to other models currently used in the literature, looks at the distance between the centers of the objects. Based on this distance it computes a short range repelling force parallel to the centerline (with obvious modifications in the case of a sphere interacting with a flat plane). This force is well adapted to small particles or objects which have spherical symmetry, but not in more general cases. In particular it is unable to produce rotation that should result from a collision taking place on a point off the line linking the centers of the objects. By contrast, our model yields a force which is distributed along the interface. In presence of symmetry, the resulting global force is along the centerline, like the model in [14], but in more general

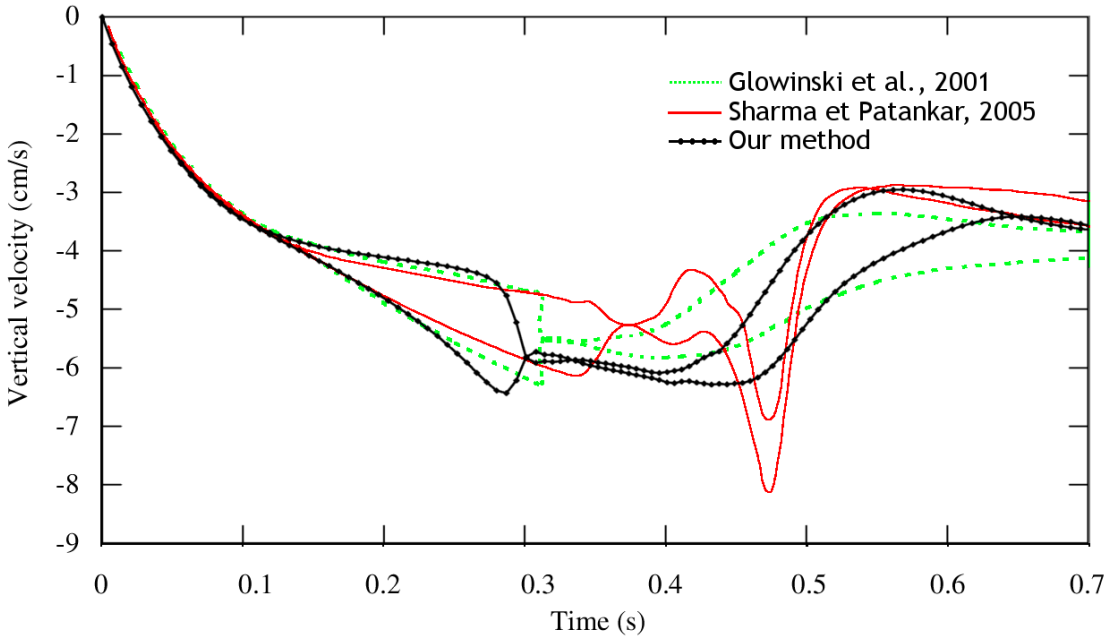

Figure 5: Drafting, kissing and tumbling of two balls falling in a Newtonian fluid : comparison of the balls velocity obtained with different methods. Physical and numerical parameters of section 4.2.2.

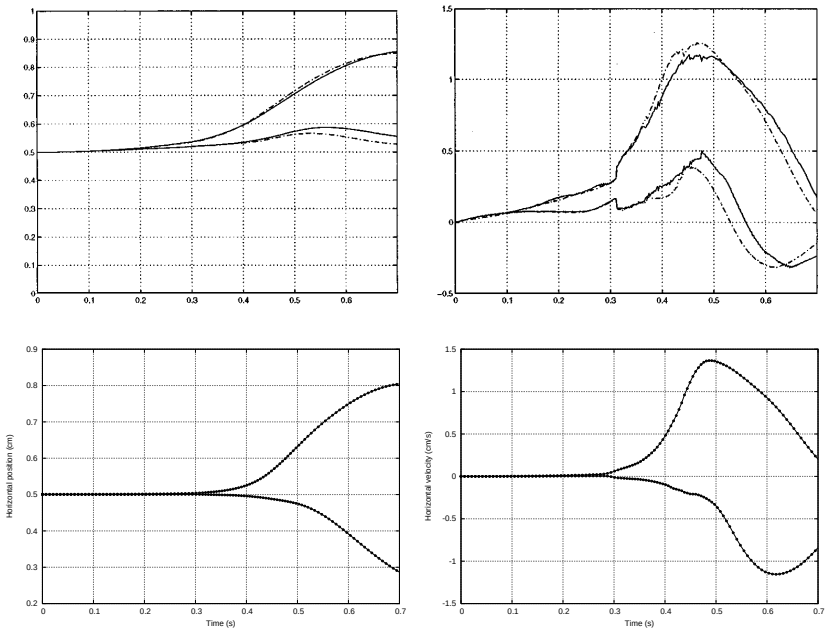

Figure 6: Drafting, kissing and tumbling of two balls falling in a Newtonian fluid: horizontal components of the position (left) and velocity (right) of the center of the two spheres . Top pictures : results from [14], bottom pictures : our results. Physical and numerical parameters of section 4.2.2.

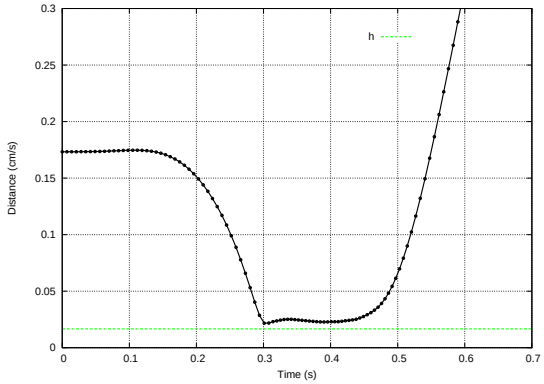

Figure 7: Drafting, kissing and tumbling of two balls falling in a Newtonian fluid : distance between the two balls. Physical and numerical parameters of section 4.2.2.

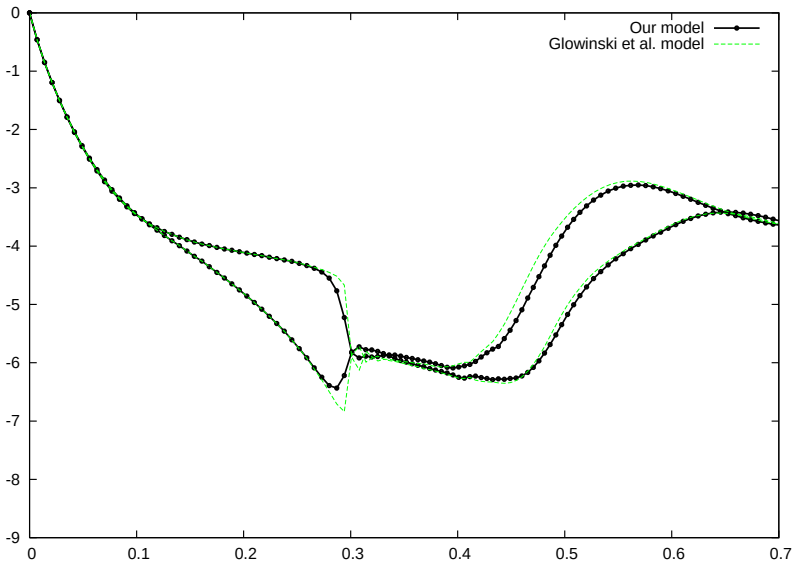

Figure 8: Drafting, kissing and tumbling of two balls falling in a Newtonian fluid : comparison between the collision model of [14] and our level set model (22). Physical and numerical parameters of section 4.2.2.

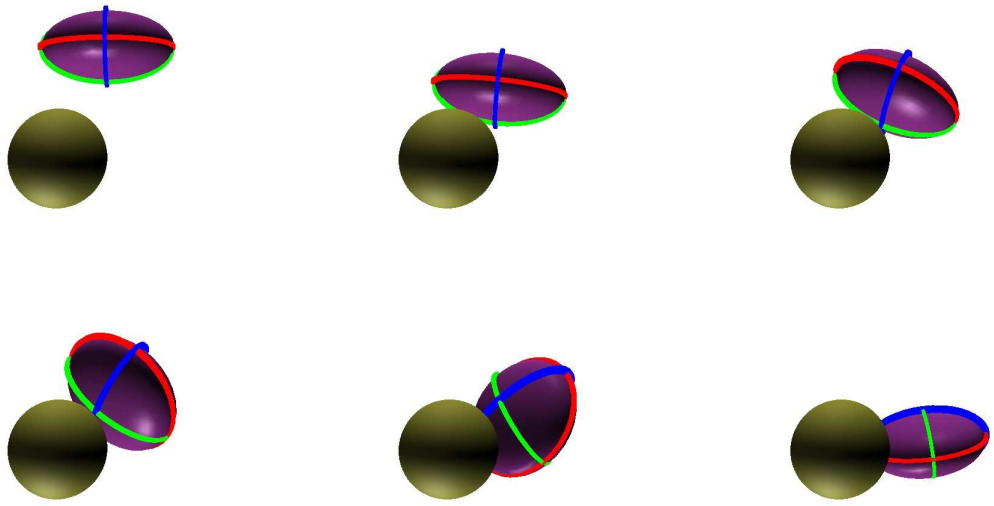

Figure 9: An ellipsoid colliding a still sphere : 3D views at time $t = 0, 0.15, 0.175, 0.2, 0.25, 0.3$ from left to right and top to bottom. The x, y and z axis of the ellipsoid are shown in color red, green and blue respectively. Physical and numerical parameters of section 4.2.3.

cases it has the ability to produce rotation. This is illustrated in Figure 9 which shows the collision of an ellipsoid with a fixed sphere. For this simulation we have taken $h = 1/64$ in a computational box $[0, 1]^3$, $\Delta t = 0.005$, $\epsilon = h$ and collision coefficients κ_{ij} equal to 5. The center of the sphere is initially located slightly on the right and in the back of the sphere so that the contact will produce rotations along the three axis. In this example, to make sure that the rotation is an effect of the contact and not of the vorticity shed in the flow by the objects, we have turned off the flow solver, like in the experiment in Figure 1.

5 Conclusion

We have presented a vortex level set method for the two-way coupling of a fluid with rigid bodies. The method is based on a penalization fluid model to enforce rigid motion inside the solid, coupled with a level set method to capture the interfaces. The level set method is also used to distribute collision forces to handle the contact between rigid bodies. A particle discretization of this model was implemented and tested in 2D and 3D flows against experiments and other fictitious domain methods. For fixed bodies the method appears as a simplified versions of vorticity creation algorithms that are commonly used for the simulation of bluff-body flows by vortex methods. For the more general case of fluid-solid interaction it essentially combines a classical vortex in cell method with an advection equation that can be solved analytically for the rigid bodies. Although the rigid motion constraint is not exactly satisfied inside the bodies, the accuracy of the method was found to be satisfactory. In addition the method enjoys the usual robustness of vortex methods which allows large time-steps. To further illustrate the capabilities of the method we show the example of a complex scene where three dimensional effects and strong interactions with the fluid are apparent. Figure 10 shows the fall of a cup on a water-air free surface. The density of the cup is 1.2. In this case the fluid has a variable density but the ideas described above readily apply in this case. An additional level set

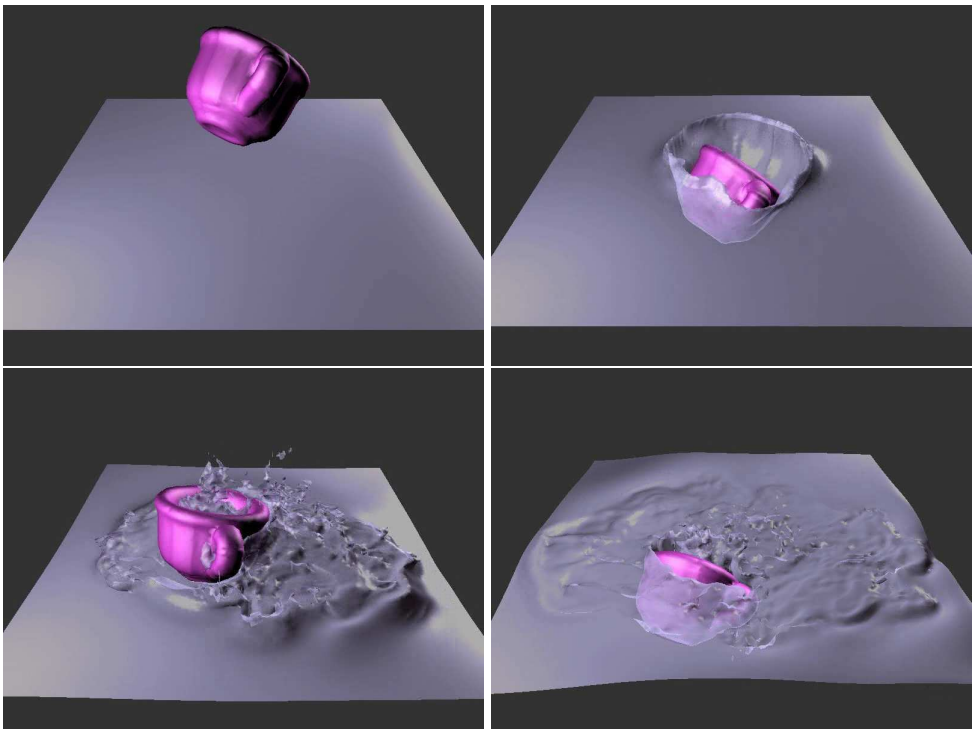

Figure 10: A cup falling into water. Level set functions are used for the air-water free surface and the cup-flow interface.

function was used to capture the air-water interface [5, 15]. No surface tension was assumed on this free surface. The viscosity of air and water was taken equal to 10^{-4}. The following successive events can be observed : the splash of the cup on the water, a slight rebound of the cup with a rotation effect, and finally the filling of the cup with water before it sinks. This calculation used 128^3 grid points. Although it is hard to validate in a quantitative way this type of calculations, these sequences do demonstrate that our model, although it essentially relies on a vortex code in a square box combined with a couple of advection equations, is able to handle the complex interaction of rigid bodies with free surface flows.

Acknowledgment

This work was partially supported by the ANR under grant # ANR-06-BLAN0306-01.

References

[1] P. Angot, C.-H. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numer. Mat.*, 81:497–520, 1999.

[2] C. Bost, G.-H. Cottet, and E. Maitre. Numerical analysis of a penalization method for the two-way coupling of an incompressible flow with a rigid body. in preparation.

[3] P. P. Brown and D. F. Lawler. Sphere drag and settling velocity revisited. *J. Environ. Engng.*, 129:222–231, 2003.

[4] M.-P. Cani and M. Desbrun. Animation of deformable models using implicit surfaces. *IEEE Trans. Visualization Comput. Graphics*, 3(1):39–50, 1997.

[5] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of eulerian interface capturing methods for incompressible fluid flows. *J. Comp. Phys.*, 124:449–464, 1996.

[6] M. Coquerelle, J. Allard, G.-H. Cottet, and M.-P. Cani. A vortex method for bi-phasic fluids interacting with rigid bodies. Arxiv preprint math. na/0607597, LMC-IMAG, 2006.

[7] G.-H. Cottet and P. Koumoutsakos. *Vortex Methods – Theory and Practice*. Cambridge University Press, New York, 2000.

[8] G.-H. Cottet and E. Maitre. A level set method for fluid-structure interactions with immersed surfaces. *M3AS*, 16(3):415–438, 2006.

[9] G.-H. Cottet, E. Maitre, and T. Milcent. Eulerian formulation and level set models for incompressible fluid-structure interaction. to appear in M2AN, 2008.

[10] G.-H. Cottet, B. Michaux, S. Ossia, and G. VanderLinden. A comparison of spectral and vortex methods in three-dimensional incompressible flows. *J. Comput. Phys.*, 175:702–712, 2002.

[11] G.-H. Cottet and P. Poncet. Advances in direct numerical simulation of 3D wall-bounded flows by vortex-in-cell methods. *J. Comput. Phys.*, 193:136–158, 2004.

[12] J. D. Eldredge, T. Colonius, and A. Leonard. Numerical simulation of the fluid dynamics of 2d rigid body motion with the vortex particle method. *J. Comput. Phys.*, 221:628–648, 2007.

[13] A. F. Fortes, D. D. Joseph, and T. S. Lundgren. Nonlinear mechanics of fluidization of beds of spherical particles. *J. Fluid Mech.*, 177:467–483, 1987.

[14] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, and J. Periaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. *J. Comput. Phys.*, 169:363–426, 2001.

[15] S. E. Hieber and P. Koumoutsakos. A lagrangian particle level set method. *J. Comput. Phys.*, 210:342–367, 2005.

[16] H. H. Hu, N. A. Patankar, and M. Y. Zhu. Direct numerical simulations of fluid–solid systems using the arbitrary lagrangian–eulerian technique. *J. Comput. Phys.*, 169:427–462, 2001.

[17] J. Janela, A. Lefebvre, and B. Maury. A penalty method for the simulation of fluid-rigid body interaction. In E. Cances and F. Gerbeau, editors, *ESAIM Proceedings*, volume 14, pages 115–123, 2005.

[18] P. Koumoutsakos and A. Leonard. High-resolution simulation of the flow around an impulsively started cylinder using vortex methods. *J. Fluid Mech.*, 296:1–38, 1995.

[19] A. Magni. Méthodes particulaires avec pénalisation pour la simulation d'écoulements autour d'obstacles. Master's thesis, Université Bordeaux 1, June 2007.

[20] N. Nitin Sharma and N. A. Patankar. A fast computation technique for the direct numerical simulation of rigid particulate flows. *J. Comput. Phys.*, 205:439–457, 2005.

[21] P. Ploumhans, G. S. Winckelmans, J. K. Salmon, A. Leonard, and M. S. Warren. Vortex methods for direct numerical simulation of three-dimensional bluff body flows: Applications to the sphere at $Re = 300$, 500 and 1000. *J. Comput. Phys.*, 178:427–463, 2002.