Annals of Mathematical Sciences and Applications
Volume 9, Number 1, 59–89, 2024

Mathematical and computational framework for moving and colliding rigid bodies in a Newtonian fluid

Céline Van Landeghem, Luca Berti, Vincent Chabannes, Agathe Chouippe, Laetitia Giraldi, Yannick Hoarau, and Christophe Prud'homme

We studied numerically the dynamics of colliding rigid bodies in a Newtonian fluid. The finite element method is used to solve the fluid-body interaction and the fluid motion is described in the Arbitrary-Lagrangian-Eulerian framework. To model the interactions between bodies, we consider a repulsive collision-avoidance model, defined by R. Glowinski in [8]. The main emphasis in this work is the generalization of this collision model to multiple rigid bodies of arbitrary shape. Our model first uses a narrow-band fast marching method to detect the set of colliding bodies. Then, collision forces and torques are computed for these bodies via a general expression, which does not depend on their shape. Numerical experiments examining the performance of the narrow-band fast marching method and the parallel execution of the collision algorithm are discussed. We validate our model with literature results and show various applications of colliding bodies in two and three dimensions. In these applications, the bodies move due to forces such as gravity, a fluid flow, or their own actuation. Finally, we present a tool to create arbitrarily shaped bodies in discretized fluid domains, enabling conforming body-fluid interface and allowing to perform simulations of fluid-body interactions with collision treatment in these realistic environments. All simulations are conducted with the **Feel++** open source library.

MSC 2020 subject classifications: 65M60, 74F10, 76M10, 70E99.
Keywords and phrases: Fluid-structure interaction, rigid body motion, collision simulation, **Feel++**.

Introduction

Fluid flows laden with particles are common in industrial and biological processes, such as fluidization, the cell transport in arteries, or the simulated

motion of articulated micro-swimmers in the human body. Due to the particle volume and the confined environments, these processes are characterized by inter-particle interactions. The modeling of these interactions, based on collision detection algorithms, and the computation of lubrication and collision forces, is challenging for arbitrarily shaped particles and add further complexity to the coupled fluid-solid interaction problem.

In recent years, a variety of approaches have been developed to simulate solid-solid and solid-wall interactions. These approaches are based on collision and lubrication forces, added to the system when solving the fluid-solid interactions [26]. When the distance between solid surfaces is small, approximately the mesh size of the computational grid, the fluid placed between the solids is squeezed out and the hydrodynamics are not resolved. The resulting underestimation of the hydrodynamic forces is compensated by a short-range analytical repulsive correction called lubrication force. In [9], this lubrication force is defined by a constant expression and in [23] its magnitude is inversely proportional to the distance between the solid surfaces. In some approaches, the lubrication model allows avoiding the direct contact between the solid surfaces [8]. Such contact-avoidance schemes are frequently used in literature to simulate the interactions between spherical bodies. In this paper, we will present the generalization of these schemes to complex-shaped and articulated bodies. The contact-avoidance models often require the adjustment of stiffness parameters, for which the optimal values are not known. Over- or underestimation of these values can falsify the results. To overcome this issue, the authors in [4] solve a minimization problem to determine the minimal force magnitude ensuring a fixed separation distance between the solids at each time step and in [12] the authors introduce the gluey particle method.

A collision force is applied, when the lubrication correction does not avoid the direct contact between the solids. Usually, collision forces are based on the soft-sphere or hard-sphere approach. When two solids are in direct contact or overlapping, the soft-sphere approach defines the collision force according to the geometrical characteristics of this contact [1]. The soft-sphere approach requires a small time step to resolve the collision process. In contrast, the hard-sphere approach [9] assumes that the collision force is impulsive. During the collision process, the velocities of the interacting solids change instantaneously at the time of contact. The magnitude of the new velocities depends on the pre-collision velocities and some physical parameters.

For large fluid systems, efficient collision detection algorithms are mandatory. These algorithms identify the pairs of solids that are actually interacting, before computing the collision forces only for the respective pairs,

which reduces the computational costs. Most collision detection algorithms consist of two phases, the broad-phase and the narrow-phase. First, the broad-phase uses spatial partitioning or sorting methods to determine the smallest possible set of neighbouring pairs that are likely to interact. In spatial partitioning, the fluid system is divided into regions and all solids assigned to the same region are considered as neighbours. The data structures employed for spatial partitioning and neighbour detection are lists [15], trees [25], adapted for large number of solids, or hash tables [19], efficient for solids with a wide range of sizes. The spatial sorting methods, as the Sort and Sweep algorithm [5], sort the solids in space to determine overlapping. Then, the narrow-phase detects whether these pairs are actually in contact by computing the distance between the solid surfaces. Although the distance between spherical solids is self-evident, it is often challenging and expensive to compute the distance between complex-shaped or articulated bodies. In the literature, multiple analytical methods for specific geometries as well as general algorithms for arbitrary shapes are proposed [26]. Our model uses the fast marching method [21] to track the distance between any pair of arbitrarily shaped bodies. When focusing on fluid systems containing up to one hundred solids, the implementation of a broad-phase algorithm is not necessary. To mitigate the high computational costs of the fast marching method, we have adapted a parallel-executable narrow-band approach.

The paper is organized as follows. Section 1 details the coupled problem describing the fluid-solid interaction. The collision model, including the collision detection algorithm based on the narrow-band fast marching method and the computation of the contact-avoidance lubrication force, is presented in section 2. The performance verification and the validation of the model are done in sections 3 and 4. Section 5 shows some applications of the collision model, including simulations of single complex-shaped and articulated solids as well as spherical multi-body interactions. A summary and conclusion are given in section 6.

1. Rigid body moving in a fluid

This section briefly describes the coupled fluid-rigid body interaction problem. A detailed description is given in [3].

The fluid model. The fluid model considered throughout the paper is incompressible and Newtonian, and the motion of the fluid domain is modelled using the Arbitrary-Lagrangian-Eulerian (ALE) formalism [7]. $\mathcal{F}_t \subseteq \mathbb{R}^d, d \in \{2, 3\}$, denotes the domain occupied by the fluid at time t, where $t \in [0, T]$

and T is the final time of the simulation. Let $\mathcal{A}_t : \mathcal{F}_0 \to \mathcal{F}_t$ be the ALE map mapping the reference fluid domain $\mathcal{F}_0$ to the current domain $\mathcal{F}_t$, defined as $\mathcal{A}_t(X) = \mathcal{A}(t, X) = X + x(t, X)$ with $x(t, X)$ the displacement of the domain. Let $u : \mathcal{F}_t \times [0, T] \to \mathbb{R}^d$ and $p : \mathcal{F}_t \times [0, T] \to \mathbb{R}$ be the fluid velocity and the hydrostatic pressure. Let ρ_f and μ be the fluid density and dynamic viscosity, constant for the fluid model under consideration. Finally, let $\sigma \in \mathcal{M}_d(\mathbb{R})$ be the fluid stress tensor and $g \in \mathbb{R}^d$ the gravity acceleration. The ALE formulation of the Navier-Stokes equations partially decouples the geometric evolution of the fluid domain from that of the fluid continuum. Due to the change of frame, the ALE time derivative substitutes the Eulerian one $\partial_t u$ as $\partial_t u = \partial_t u|_{\mathcal{A}} - (\partial_t x \cdot \nabla)u = \partial_t u|_x - (u_{\mathcal{A}} \cdot \nabla)u$ in the momentum equation. The first term of the ALE derivative corresponds to the time variation of the fluid velocity as seen in the arbitrary frame, while the second contains the relative velocity between the fluid continuum and the new reference frame. The Navier-Stokes equations in the ALE frame are:

$$\begin{aligned} \rho_f \partial_t u|_{\mathcal{A}} + \rho_f \Big((u - u_{\mathcal{A}}) \cdot \nabla\Big) u &= -\nabla \cdot \sigma + \rho_f g, && \text{in } \mathcal{F}_t, \\ \nabla \cdot u &= 0, && \text{in } \mathcal{F}_t. \end{aligned} \tag{1}$$

The rigid body equations. A rigid body moving in a fluid is described by the motion of its center of mass, given by the Newton equation, and the rotation matrix between its local frame and the laboratory frame, which evolves according to the Euler equation. In this paper, the motion of a body is influenced by fluid stresses and gravity, contact and lubrication forces that act as external forces F_e and torques T_e. Let $\mathcal{S} \subset \mathbb{R}^d$ be the domain occupied by the body, $\rho_{\mathcal{S}} \in \mathbb{R}_{>0}$ its density and $m = \int_{\mathcal{S}} \rho_{\mathcal{S}}$ its mass. Let $U : [0, T] \to \mathbb{R}^d$ and $\omega : [0, T] \to \mathbb{R}^{d^*}$, where $d^* = 1$ if $d = 2$ or $d^* = 3$ if $d = 3$, be the linear and angular velocity of the rigid body as seen from the laboratory frame. Let $x^{CM} = m^{-1} \int_{\mathcal{S}} \rho_{\mathcal{S}} x$ be its center of mass, $x^{CM} \in \mathbb{R}^d$, and $I = \int_{\mathcal{S}} \rho_{\mathcal{S}} (x - x^{CM}) \otimes (x - x^{CM})$ its inertia tensor, $I \in S^{d^*}_{++}$ is positive definite and symmetric. We chose to describe the rotation matrix $R(\theta) : \Theta \to SO(d)$ using Euler angles $\theta \in \Theta$, where $\Theta = [-\pi, \pi]$ if $d = 2$, or $\Theta = [-\pi, \pi] \times [0, \pi] \times [0, \pi/2]$ if $d = 3$. The Newton and Euler equations, describing the dynamics of a three-dimensional rigid body, are:

$$\begin{aligned} m \frac{d}{dt} U &= F_e - \int_{\partial \mathcal{S}} -pn + \mu(\nabla u + \nabla u^T) n, \\ \frac{d}{dt}(R I R^T \omega) &= T_e - \int_{\partial \mathcal{S}} [-pn + \mu(\nabla u + \nabla u^T) n] \times (x - x^{CM}), \end{aligned} \tag{2}$$

where n is the unit outward normal to $\partial\mathcal{S}$ and one has:

$$\frac{d}{dt}\theta_i = \omega_i, \quad \text{for } i \in \{x, y, z\},$$
$$R = R_z(\theta_z)R_y(\theta_y)R_x(\theta_x),$$

where $R(\theta_i)$ denotes the rotation matrix around axis $i \in \{x, y, z\}$ of angle θ_i. If $d = 2$, $R(\theta)$ has the form:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix},$$

while in three dimensions:

$$R_z(\theta_z) = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_y(\theta_y) = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix},$$
$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix}.$$

Fluid-solid interaction. The interaction between the rigid body and the fluid is ensured by the balance of stresses in the Newton and Euler equations (2) and a coupling condition at the interface, imposing the continuity of the velocities:

$$u = U + \omega \times (x - x^{CM}) \quad \text{on } \partial\mathcal{F}_t \cap \partial\mathcal{S}. \tag{3}$$

Finally, the definition of $\mathcal{A}_t(X)$ requires the computation of the displacement $x(t, X)$ in the reference domain. In order to find $x(t, X)$, we solve:

$$\begin{aligned} \nabla \cdot ([1 + \tau(X)]\nabla_X x(t, X)) &= 0 && \text{in } \mathcal{F}_0, \\ x(t, X) &= g(t, X) && \text{in } \partial\mathcal{F}_0, \end{aligned} \tag{4}$$

where $g(t, X) = \int_0^t U + \omega \times (X - X^{CM})$ is the rigid displacement of $\mathcal{S}$ and $\tau(X)$ is a space-dependent coefficient, related to the volume of the simplexes in the triangulation.

Numerical solution. The problem is discretized over a triangulation of the fluid domain. The discrete fluid velocity u_h and pressure p_h belong to the inf-sup stable Taylor-Hood space, where P_2 continuous finite elements are

chosen for the velocity and P_1 continuous finite elements for the pressure. The discrete ALE map $\mathcal{A}_h^t$ is also discretized using P_1 continuous finite elements. The coupling condition of the fluid-solid interaction is encoded in the finite element test spaces, as proposed in [14], which leads to the construction of an operator P that couples the velocities at the interface at the discrete level. In our computations, the interface is conforming and the solution of equation (4) handles small mesh deformations, with $\tau(X)$ a piecewise constant coefficient, defined on each element e of the mesh as $\tau\big|_e = (1 - V_{min}/V_{max})/(V_e/V_{max})$, where V_{max}, V_{min} and V_e are the volumes of the largest, smallest and current element of the domain discretization [11]. In case of larger domain deformations, we perform remeshing and preserve the discretization of the interface at the same time.

2. Collision model

In this section we will detail the two phases that characterize our collision model: a collision detection algorithm and a lubrication model. We do not add a scheme handling direct contact between body surfaces, since our lubrication model avoids this kind of interactions. We will first describe the three types of rigid bodies for which we have developed our collision model. Then we will detail the collision detection algorithm before defining the lubrication force.

2.1. Types of rigid bodies

We adjust the collision model for three different types of rigid bodies. First, we consider spherical bodies in two and three dimensions. As collision models for spherical bodies are common in literature, we use this case for validation. Then, we adapt the first model for complex-shaped bodies. In this paper, ellipses and ellipsoids are used for the two and three-dimensional simulations, respectively. At last, we consider articulated bodies, and the three-sphere swimmer [16], in particular. This type of swimmer consists of three identically-sized spheres connected by rods. To move, the swimmer alternates between extending and retracting these rods. The sequence begins with retracting the left sphere, followed by the right one. Then, it extends the left sphere and finally, the right sphere. This sequence of four movements result in a straight motion.

2.2. Collision detection

The first part of our model is a collision detection algorithm. Collision detection is an important task since it identifies the pairs of bodies that are

actually interacting. Two bodies are interacting when the distance between their surfaces is less than the width of the collision zone, denoted as ρ. In absence of a detection phase, collision forces are computed for each pair of bodies in the domain, independently of whether there will be an interaction. Most collision detection algorithms include two phases. First, the broad-phase identifies pairs of bodies in proximity. Then, the distance between the identified pairs is computed explicitly during the narrow-phase. Based on this computed distance, one concludes if collision forces are applied. Our collision detection algorithm solely relies on the narrow-phase, which involves computing the distance between each pair of bodies. Depending on the body shape, this is done in two different ways. We will outline the method for spherical bodies first, followed by the approach for complex-shaped or articulated bodies. Both approaches require some parameters, which are defined during a pre-processing step. This step will also be described.

Spherical bodies. The distance d_{ij} between the surfaces of two spherical bodies $\mathcal{S}_i$ and $\mathcal{S}_j$, $i \neq j$, is calculated by first computing the distance between their centers of mass x_i^{CM} and x_j^{CM} and then subtracting their radii r_i and r_j:

$$d_{ij} = ||x_j^{CM} - x_i^{CM}||_2 - r_i - r_j.$$

When this distance is smaller than a given parameter ρ, representing the width of the collision zone (detailed in subsection 2.3), then the identifiers i, j and the distance d_{ij} are stored in a collision map. Stored data is used during the phase where lubrication forces are computed.

The same formula can be used to obtain the distance d_i between the surface of body $\mathcal{S}_i$ and the boundary of the fluid domain $\partial\mathcal{F}_t$:

$$d_i = ||x_{i'}^{CM} - x_i^{CM}||_2 - 2r_i.$$

where $x_{i'}^{CM}$ is the center of mass of the nearest imaginary body placed on the outside of the fluid boundary [27]. This imaginary body has the same radius as the real body. Again, if d_i is smaller than the width ρ, then i and d_i are stored. The parameters used in these two equations, i.e., body identifiers, centers of mass and radii, are defined in a pre-processing step described in Algorithm 1.

Complex-shaped or articulated bodies. When considering complex-shaped or articulated bodies, then no explicit expression is provided to compute the distance between the surfaces. To address this, we use the fast marching method, based on a level set algorithm introduced by J. A. Sethian [21].

Algorithm 1 Pre-processing phase

Input: Computational domain: mesh M
Output: Set of bodies identifiers: bodyIds, set of centers of mass: massCenters, set of bodies markers: bodyMarkers, set of radii: radii, set of centers of mass of imaginary bodies: ImagmassCenters, and the fluid marker: fluidMarker

```
for 𝒮 in M.bodies do
    bodyIds.append(𝒮.id())
    massCenters.append(𝒮.massCenter())
    bodyMarkers.append(𝒮.boundary().name())

    if type = spherical then
        radii.append(𝒮.radius())
        ImagmassCenters.append(𝒮.ImagmassCenters())
    end if

end for

fluidMarker ← ℱ_t.boundary().name()
```

When applied to a body $\mathcal{S}_i$, this method yields the distance field D_i from that body to the rest of the domain. Thus, the distance field D_i is zero at the boundary of the body and positive elsewhere. Using the distance fields D_i and D_j, then the minimum distance between the surfaces $\partial\mathcal{S}_i$ and $\partial\mathcal{S}_j$ of the corresponding bodies is found by:

$$d_{ij} = ||\arg\min_{x\in\partial\mathcal{S}_j} D_i(x) - \arg\min_{x\in\partial\mathcal{S}_i} D_j(x)||_2.$$

The minima $X_i = \arg\min_{x\in\partial\mathcal{S}_i} D_j(x)$ and $X_j = \arg\min_{x\in\partial\mathcal{S}_j} D_i(x)$ represent the contact points of the bodies $\mathcal{S}_i$ and $\mathcal{S}_j$, i.e., the coordinates of the surface points where the bodies will interact. To get the minimal distance between a body surface and the domain boundary, we use:

$$d_i = ||\arg\min_{x\in\partial\mathcal{F}_t} D_i(x) - \arg\min_{x\in\partial\mathcal{S}_i} D_{\mathcal{F}_t}(x)||_2,$$

where $D_{\mathcal{F}_t}$ represents the distance field obtained by applying the fast marching method to the domain boundary $\partial\mathcal{F}_t\backslash(\cup_i\partial\mathcal{S}_i)$. The parameters that will be stored if $d_{ij} \leq \rho$ or $d_i \leq \rho$, are the identifiers, the minimal distance and the contact points.

When the fluid domain is large, the fast marching algorithm gets computationally expensive, especially in three dimensions. To accelerate the method, we develop the narrow-band approach, which computes the dis-

tance field only in a restricted neighborhood of the body boundary. The size of this neighborhood is set to a predefined threshold d_{max}. As collision forces are only applied on a collision zone of width ρ, the threshold d_{max} can be defined as function of ρ. Upon reaching the threshold distance d_{max}, the narrow-band approach assigns a default value δ to the distance field, corresponding to the maximum value reached:

$$D_i^{NB} = \begin{cases} 0\,, & \text{on} & \partial\mathcal{S}_i, \\ D_i\,, & \text{for} & D_i \leq d_{max}, \\ \delta\,, & \text{elsewhere} & . \end{cases}$$

The narrow-band algorithm is designed to run in parallel. We evaluate the performance of this approach in section 3. In our implementation, the narrow-band fast marching function takes as parameter the boundary marker of the targeted body. Therefore, all boundary markers are stored during the pre-processing phase, presented in Algorithm 1.

When considering articulated bodies, the same formulas as for the complex-shaped case are applied to each component of the body. So, for a pair of three-sphere swimmers, we calculate the distances between each sphere of the first swimmer and every sphere of the second. All steps of the collision detection algorithm for both methods are given by Algorithm 2 for

Algorithm 2 Collision detection algorithm for spherical bodies

Input: Output of the pre-processing phase
Output: Collision map

for i in bodyIds **do**

  for j in bodyIds, $j > i$ **do**
    $d_{ij} \leftarrow ||\text{massCenters}_i - \text{massCenters}_j||_2 - \text{radii}_i - \text{radii}_j$
    if $d_{ij} \leq \rho$ **then**
      Store i, j and d_{ij} in collision map
    end if
  end for

  for ImagCoord in ImagmassCenters$_i$ **do**
    $d_i \leftarrow ||\text{massCenters}_i - \text{ImagCoord}||_2 - 2\text{radii}_i$
    if $d_i \leq \rho$ **then**
      $j \leftarrow -1$
      Store i, j, ImagCoord and d_i in collision map
    end if
  end for

end for

Algorithm 3 Collision detection algorithm for complex-shaped bodies

Input: Output of the pre-processing phase
Output: Collision map

for i in bodyIds **do**
  $D_i \leftarrow$ narrow-band(bodyMarkers$_i$, $d_{max} = 1.5\rho$)
end for
$D_{\mathcal{F}_t} \leftarrow$ narrow-band(fluidMarker, $d_{max} = 1.5\rho$)

for i in bodyIds **do**

  for j in bodyIds, $j > i$ **do**
    $d_{ij} \leftarrow ||X_j - X_i||_2$
    if $d_{ij} \leq \rho$ **then**
      Store i, j, d_{ij}, X_i and X_j in collision map
    end if
  end for

  $d_i \leftarrow ||X_{\mathcal{F}_t} - X_i||_2$
  if $d_i \leq \rho$ **then**
    $j \leftarrow -1$
    Store i, j, d_i, X_i and $X_{\mathcal{F}_t}$ in collision map
  end if
end for

spherical and by Algorithm 3 for complex-shaped bodies. Computing the centers of the imaginary bodies is only possible when the dimensions of the fluid domain, which must be rectangular, are known. For this reason, we have developed a more generic algorithm for spherical bodies, where the distance between a body and the fluid boundary is determined using the narrow-band approach. Despite these advancements, the original algorithm remains viable for simulating spherical bodies in rectangular domains due to its lower computational costs.

2.3. Collision force

In the second part of our collision model, we focus on computing collision forces. The lubrication model is based on a short-range repulsive force introduced by R. Glowinski [8]. The force is activated when the distance between the surfaces of two bodies is smaller than the parameter ρ. This parameter represents the width of the collision zone, i.e., the zone where collision forces must be applied to prevent the bodies from overlapping and avoid direct contact. The definition of the force is quite simple: it is parallel to

the vector that connects the contact points of the bodies and its intensity increases as the distance decreases. We will first give the definition of this force for the simple case of spherical bodies. Then, we detail the required modifications, in particular the addition of its torque, for complex-shaped and articulated bodies.

Spherical bodies. For the definition of the repulsive force in the case of spherical bodies, we rely on articles [28, 24, 27]. The force that models the interaction between two bodies $\mathcal{S}_i$ and $\mathcal{S}_j$ is given by:

$$\overrightarrow{F_{ij}} = \frac{1}{\epsilon}\overrightarrow{x_j^{CM} x_i^{CM}}(\rho - d_{ij})^2 \mathbb{1}_{d_{ij}\leq\rho}.$$

In the same way, the interaction between a body $\mathcal{S}_i$ and the domain boundary $\partial\mathcal{F}_t$ is defined by:

$$\overrightarrow{F_{i\mathcal{F}_t}} = \frac{1}{\epsilon_{\mathcal{F}}}\overrightarrow{x_{i'}^{CM} x_i^{CM}}(\rho - d_i)^2 \mathbb{1}_{d_i\leq\rho}.$$

Both equations contain a quadratic activation term. The vector connecting the centers of mass gives the direction of the force, and the stiffness parameters ϵ and $\epsilon_{\mathcal{F}}$ describe its intensity for body-body and body-domain interactions. Finding the optimal values for these two parameters is not trivial, since their values depend on fluid and body properties. We refer the reader to the indications given in [27]: when the width of the collision zone is fixed to $\rho = 0.5h \sim 2.5h$, where h is the mesh size, and the ratio between the body and fluid density is equal to 1, then one can suppose $\epsilon \approx h^2$ and $\epsilon_{\mathcal{F}} \approx \frac{\epsilon}{2}$. As already mentioned, these forces are only computed for pairs stored during collision detection phase, i.e., for pairs whose distance is less than ρ, for which the indicator functions $\mathbb{1}_{d_{ij}\leq\rho} \equiv 1$ and $\mathbb{1}_{d_i\leq\rho} \equiv 1$. The total repulsion force applied on $\mathcal{S}_i$ is defined by:

$$\overrightarrow{F_i} = \sum_{(i,j)|d_{ij}\leq\rho} \overrightarrow{F_{ij}} + \sum_{i|d_i\leq\rho} \overrightarrow{F_{i\mathcal{F}_t}},$$

for all pairs $\mathcal{S}_i$ - $\mathcal{S}_j$ and $\mathcal{S}_i$ - $\partial\mathcal{F}_t$ present in the collision map that results from the detection algorithm. The total repulsion force $\overrightarrow{F_i}$ is added to the external forces $\overrightarrow{F_e}$ of the Newton equation in (2), describing the linear velocity of the body $\mathcal{S}_i$.

Complex-shaped or articulated bodies. For the case of non-spherical bodies, the direction of the vector connecting the mass centers x_i^{CM} and x_j^{CM} no

Algorithm 4 Collision force algorithm for spherical bodies

Input: Output of the pre-processing phase and collision map
Output: Set of collision forces: F, and set of collision torques: T

for pair in collision map **do**
  if pair.$j = -1$ **then**
    $F_{i\mathcal{F}_t} \leftarrow \frac{1}{\epsilon_{\mathcal{F}}}(\text{massCenters}_{\text{pair}.i} - \text{pair.ImagCoord})(\rho - \text{pair}.d_i)^2$
    $F_{\text{pair}.i}$.add($F_{i\mathcal{F}_t}$)
  else
    $F_{ij} \leftarrow \frac{1}{\epsilon}(\text{massCenters}_{\text{pair}.i} - \text{massCenters}_{\text{pair}.j})(\rho - \text{pair}.d_{ij})^2$
    $F_{\text{pair}.i}$.add(F_{ij}), $F_{\text{pair}.j}$.add($-F_{ij}$)
  end if
end for

longer corresponds to that of the vector connecting the contact points X_i and X_j. Thus, the equations are slightly modified:

$$\vec{F_{ij}} = \frac{1}{\epsilon}\overrightarrow{X_jX_i}(\rho - d_{ij})^2\mathbb{1}_{d_{ij}\leq\rho}, \qquad \vec{F_{i\mathcal{F}_t}} = \frac{1}{\epsilon_{\mathcal{F}}}\overrightarrow{X_{\mathcal{F}_t}X_i}(\rho - d_i)^2\mathbb{1}_{d_i\leq\rho}.$$

The values of the parameters ϵ and $\epsilon_{\mathcal{F}}$ have to be chosen smaller than those of the previous system. These modified expressions imply that the collision forces will lead to body rotations. The torque used to model these rotations is added to the external torques T_e of the Euler equation in (2), and is defined by:

$$\overrightarrow{T_i} = -\overrightarrow{x_i^{CM}X_i} \times \overrightarrow{F_i}.$$

This torque is computed for each body in the collision map. The same approach is used for articulated bodies. The forces and torques are added for each element of the collision map. The algorithms given by Algorithm 4 and 5 illustrate the implementation of the lubrication model for spherical and complex-shaped bodies.

3. Numerical experiments

3.1. Performance of the narrow-band approach

The following test illustrates the performance of the narrow-band approach of the fast marching method, both in two-dimensional and three-dimensional setups. The geometry consists in two spheres of same center but different radii. The radius of the inner and outer sphere is respectively set to $r_i = 0.1$

Algorithm 5 Collision force algorithm for complex-shaped bodies

Input: Output of the pre-processing phase and collision map
Output: Set of collision forces F, and set of collision torques T

for pair in collision map **do**
  if pair.$j = -1$ **then**
    $F_{i\mathcal{F}_t} \leftarrow \frac{1}{\epsilon_{\mathcal{F}}}(\text{pair}.X_i - \text{pair}.X_{\mathcal{F}_t})(\rho - \text{pair}.d_i)^2$
    $F_{\text{pair}.i}$.add$(F_{i\mathcal{F}_t})$
    $T_{\text{pair}.i}$.add$(-(\text{pair}.X_i - \text{massCenters}_{\text{pair}.i}).cross(F_{i\mathcal{F}_t}))$
  else
    $F_{ij} \leftarrow \frac{1}{\epsilon}(\text{pair}.X_i - \text{pair}.X_j)(\rho - \text{pair}.d_{ij})^2$
    $F_{\text{pair}.i}$.add(F_{ij}), $F_{\text{pair}.j}$.add$(-F_{ij})$
    $T_{\text{pair}.i}$.add$(-(\text{pair}.X_i - \text{massCenters}_{\text{pair}.i}).cross(F_{ij}))$
    $T_{\text{pair}.j}$.add$(-(\text{pair}.X_j - \text{massCenters}_{\text{pair}.j}).cross(-F_{ij}))$
  end if
end for

and $r_o = 2.0$. First, we use the fast marching method to determine the distance field from the inner sphere boundary to the outer sphere boundary. This is equivalent to applying the narrow-band approach to the inner sphere by setting the threshold distance d_{max} to $d_{max} = r_o - r_i = 1.9$. Then, we continue to apply the narrow-band approach to the inner sphere but considering smaller thresholds. For the two-dimensional test the thresholds are set to $d_{max} = \{0.0625, 0.125, 0.25, 0.5, 1.0\}$, and we fix the mesh size to $h = 0.01$. The mesh contains $E_{2D} = 294918$ elements. In three dimensions, the mesh has $E_{3D} = 1385385$ elements, using a size equal to $h = 0.05$. The set of thresholds is $d_{max} = \{0.125, 0.25, 0.5, 1.0\}$.

The table 1 shows the execution time of the narrow-band approach for the different thresholds as well as the speed-up in two and three dimensions. We add the number of band elements E_b, i.e., the number of mesh elements in the zone where the distance field is actually computed, and the element ratio $\frac{E_{nD}}{E_b}$, where n corresponds to the dimension. It can be observed that the execution time decreases significantly when considering smaller thresholds. This test underscores the efficacy and significance of the narrow-band approach in our collision model.

3.2. Performance of the parallel collision algorithm

The aim of this second numerical experiment is to observe the performance of the parallel implementation of the collision algorithm using the narrow-band fast marching method as well as to analyze the proportion of the execution

Table 1: Results of the performance test for the narrow-band approach in two and three dimensions. The table shows the execution time and speed-up for different threshold values d_{max}

Threshold d_{max}	Band elements E_b	Element ratio $\frac{E_{2D}}{E_b}$	Execution time in 2D	Speed-up
1.9	294918	1.00	1.959 s	/
1.0	89898	3.28	0.722 s	2.71
0.5	27082	10.89	0.411 s	4.76
0.25	9430	31.27	0.316 s	6.20
0.125	4104	71.86	0.276 s	7.09
0.0625	2262	130.38	0.261 s	7.50
Threshold d_{max}	Band elements E_b	Element ratio $\frac{E_{3D}}{E_b}$	Execution time in 3D	Speed-up
1.9	1385385	1.00	22.851 s	/
1.0	254365	5.45	5.869 s	3.89
0.5	43784	31.64	2.368 s	9.65
0.25	10238	135.32	1.815 s	12.59
0.125	4537	305.35	1.759 s	12.99

time that is attributed to the collision model during the resolution of the fluid-solid interaction. The choice of the preconditioners, a direct LU solver in two dimensions and a block preconditioner in 3D, are detailed in [3]. We consider different numbers of spherical bodies distributed in a rectangular domain filled with a steady fluid. No gravity is applied and the magnitude of the collision forces is chosen close to zero, so that the bodies do not move during the simulation. Each domain configuration is simulated for ten time iterations in sequential and in parallel, using different numbers of processors. For each case, various details are presented: the number of bodies, the number of mesh nodes, the number of body-body and body-wall interactions during each iteration as well as the speed-up of the collision algorithm (preprocessing phase, collision detection algorithm, and computation of collision force) after the ten iterations. This speed-up $\frac{T_{np1}}{T_{npN}}$ is given by the ratio between the sequential execution time and the parallel execution time, when running the simulation on N processors. The results are given in Table 2 and Table 3.

One can observe that the parallel execution reduces the execution times, and that the speed-up increases with the number of processors. For multiple interactions, the speed-up remains almost constant independently of the number of bodies. The execution time spent in collision detection represents the majority of the total execution time of the algorithm. For the simulations conducted in this section, the execution time of the detection phase repre-

Table 2: Performance of the parallel collision algorithm in two dimensions. The table shows the speed-up of the execution time when running the simulations in parallel

Bodies	Mesh nodes	Body-body	Body-wall	$\frac{T_{np1}}{T_{np4}}$	$\frac{T_{np1}}{T_{np16}}$	$\frac{T_{np1}}{T_{np24}}$
1	16462	0	1	3.14	8.22	11.04
25	17780	0	16	1.88	2.06	2.12
49	18284	28	24	1.90	2.08	2.15
81	18813	72	32	1.92	2.09	2.13
100	19139	100	36	1.81	2.01	2.10

Table 3: Performance of the parallel algorithm in three dimensions. The sequential and parallel execution times are compared for different numbers of processors

Bodies	Mesh nodes	Body-body	Body-wall	$\frac{T_{np1}}{T_{np4}}$	$\frac{T_{np1}}{T_{np16}}$	$\frac{T_{np1}}{T_{np24}}$
1	17073	0	1	2.23	6.49	7.99
27	18213	0	26	1.94	3.99	5.03
64	18789	0	56	1.98	3.98	5.22
125	20301	70	98	1.99	3.59	5.13

sents 99% of the total collision time. The difference in speed-ups between the two and three-dimensional case could be explained by the larger execution time taken by the distance computation, since the number of node-node connections in 3D is, on average, larger than in 2D. Furthermore, the execution time of the two-dimensional collision model represents up to 20% of the total time for the fluid-solid interaction resolution, when considering multiple interactions and using a direct solver. In three dimensions, using a block preconditioner, the proportion associated to the collision algorithm does not exceed 5%.

4. Validation

To validate our model we consider the motion of a circular solid in an incompressible Newtonian fluid in two and three dimensions.

For the validation in 2D, we consider a rigid body of radius $r = 0.125$cm and density $\rho_{\mathcal{S}} = 1.25\frac{\text{g}}{\text{cm}^3}$ initially located at $(1\text{cm}, 4\text{cm})$ in a channel of width 2cm and height 6cm. Under the effect of gravity $g = 981\frac{\text{cm}}{\text{s}^2}$ the particle falls towards the bottom of a channel filled with a fluid of density $\rho_f = 1\frac{\text{g}}{\text{cm}^3}$ and viscosity $\mu = 0.1\frac{\text{g}}{\text{cm s}}$. Both fluid and particle are at rest at time $t = 0.0$s. The mesh size and time step used in this simulation are respectively fixed at $h = 0.01$cm and $\Delta t = 0.001$s. The width of the collision

zone is set to $\rho = 0.015$cm. One can use a stiffness parameter for particle-fluid domain interaction $\epsilon_{\mathcal{F}} < \frac{h^2}{2}$, and for the results shown in the figures we used $\epsilon_{\mathcal{F}} = 5 * 10^{-6}$. In Figure 1, we plot the time evolution of four quantities and compare them to literature results from Wan and Turek [27], Wang, Guo and Mi [28]. These quantities are the vertical coordinate of the center of mass y^{cm}, the vertical translational velocity v_y, the Reynolds number Re and the translational kinetic energy E_t defined by:

$$Re = \frac{2r\rho_{\mathcal{S}}\sqrt{v_x^2 + v_y^2}}{\mu} \quad \text{and} \quad E_t = 0.5\pi r^2 \rho_{\mathcal{S}}(v_x^2 + v_y^2),$$

where v_x is the horizontal translational velocity. One can observe that all results are in good agreement. The small differences before and after collision can be explained by different definitions of collision parameters.

We perform the same simulation for the 3D validation. Initially, a sphere of radius $r = 0.75$cm and density $\rho_s = 1.12\frac{\text{g}}{\text{cm}^3}$ is located at $(0\text{cm}, 12.75\text{cm}, 0\text{cm})$ in a computational box of dimensions $[-4\text{cm}, 4\text{cm}] \times [0\text{cm}, 15\text{cm}] \times [-4\text{cm}, 4\text{cm}]$. The box is filled with a fluid of density $\rho_f = 0.962\frac{\text{g}}{\text{cm}^3}$ and viscosity $\mu = 1.13\frac{\text{g}}{\text{cm s}}$. The mesh size and the time step are set to $h = 0.1$cm and $\Delta t = 0.01$s. Regarding the lubrication force parameters, we fixed the width of the collision zone to $\rho = 0.2625$cm and the stiffness parameter to $\epsilon_{\mathcal{F}} = 5 * 10^{-6}$. The comparison of the present results to literature results [30] is given in Figure 2. We obtain the same results for the vertical coordinate of the center of mass and the vertical translational velocity.

5. Applications

5.1. Isolated object

In this subsection we simulate the interaction between a complex-shaped body falling under the effect of gravity and the boundary of the computational domain filled with an incompressible Newtonian fluid in two and three dimensions.

For the first case, we consider a two-dimensional ellipse located in a channel of width $L = \frac{16}{130}$cm and infinite length. The channel contains a fluid of density $\rho_f = 1.195\frac{\text{g}}{\text{cm}^3}$ and viscosity $\mu = 0.305\frac{\text{g}}{\text{cm s}}$. The ellipse's long and short axes are respectively set to $a = 0.1$cm and $b = 0.05$cm. Its density is equal to $\rho_{\mathcal{S}} = 1.35\frac{\text{g}}{\text{cm}^3}$. Due to its initial orientation, set to $\theta = \frac{\pi}{3}$ with respect to the horizontal axis, the ellipse will collide with the right and left walls. To model these collisions the length of the lubrication zone is fixed at $\rho = 0.05$cm and a stiffness parameter $\epsilon_{\mathcal{F}} = 10^{-5}$ is used. The simulation

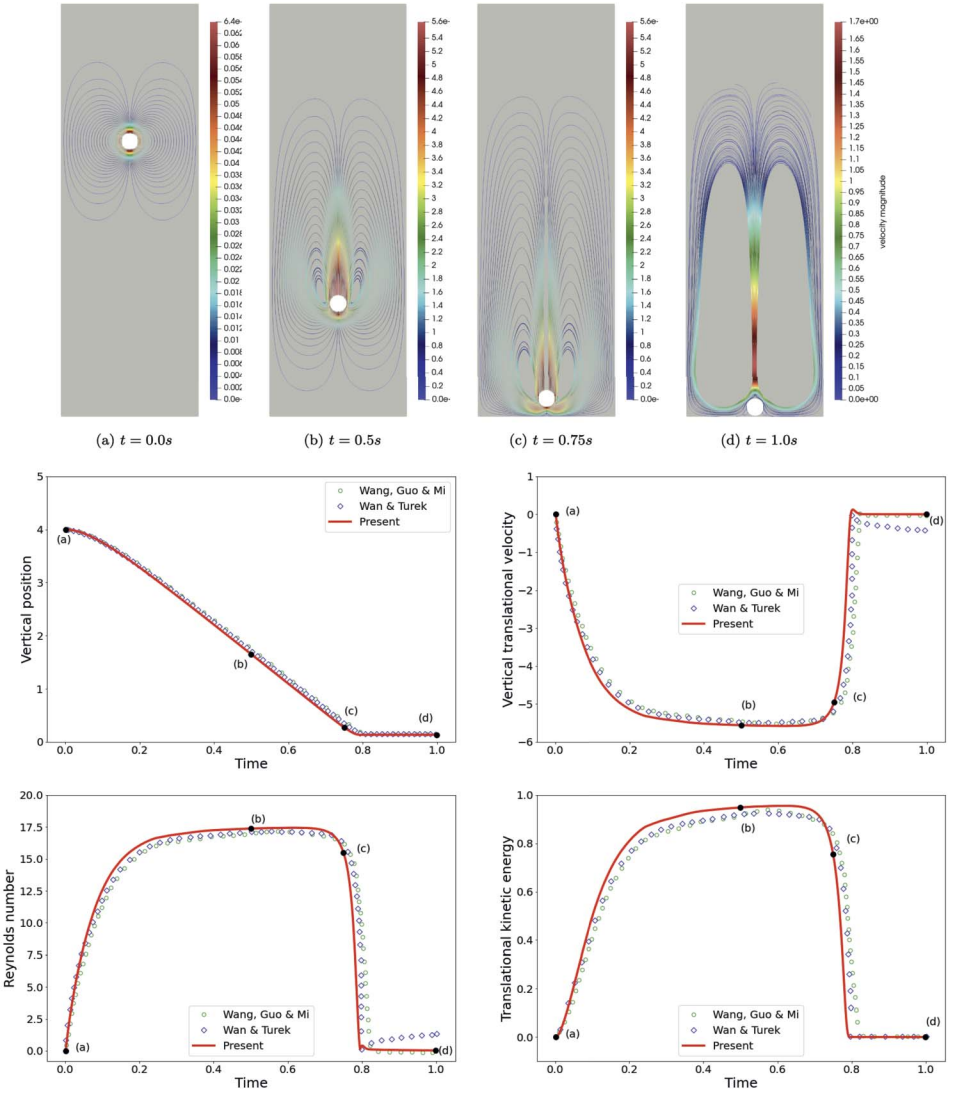


Figure 1: Evolution of the vertical position of the center of mass, vertical translational velocity, Reynolds number and translational kinetic energy. The red line are the present results, the green dotted and blue dotted lines correspond to results respectively taken from [28] and [27]. The streamlines and body position at four different time instants are represented.

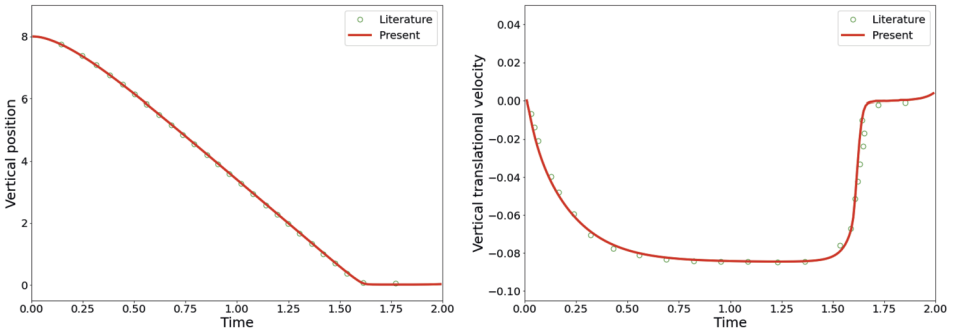
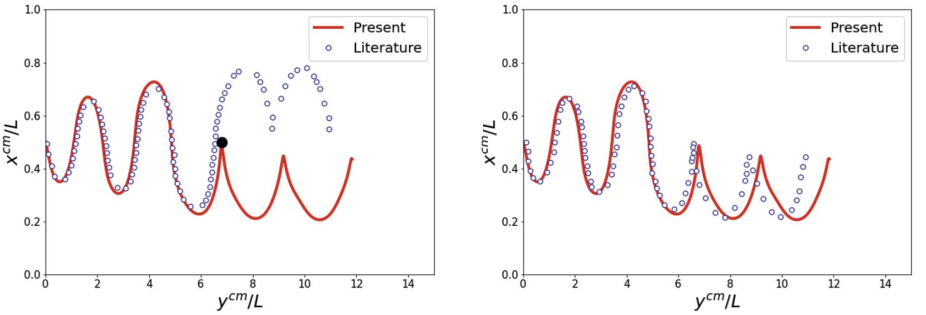


Figure 2: Comparison of the vertical position of the center of mass and vertical translational velocity observed in present results, red line, to literature results [30], dotted line.

is performed using a mesh size $h = 0.001$cm and a time step $\Delta t = 0.001$s. Figure 3a shows the trajectory of the ellipse compared to literature [29]. This trajectory shows that the ellipse first performs oscillations between the right and left wall before reaching a horizontal position, at the instant given by the black point. Then, it performs rotations close to a single wall. In the present results, these rotations are taking place near the left wall, but in the reference article, they happen near the right wall. Given that the ellipse is in a horizontal position at the beginning of these rotations, any small difference in the collision model or the numerical resolution techniques can explain this difference. To show that the present results remain close to those in the literature, we consider in Figure 3b the symmetric trajectory for the rotations of [29]. This test case allows us to validate our implementation for the case of arbitrarily shaped bodies.

The second case represents the simulation of a three-dimensional ellipsoid of density $\rho_{\mathcal{S}} = 1.25\frac{\text{g}}{\text{cm}^3}$ and axis of length $a = 0.4$cm and $b = c = 0.2$cm falling in a rectangular computational domain filled with a fluid of density $\rho_f = 1\frac{\text{g}}{\text{cm}^3}$ and viscosity $\mu = 0.01\frac{\text{g}}{\text{cm s}}$. The dimensions of the domain are set to $[0\text{cm}, 1\text{cm}] \times [0\text{cm}, 8\text{cm}] \times [0\text{cm}, 0.4\text{cm}]$ and at $t = 0$s the ellipsoid is located at $(0.5\text{cm}, 6\text{cm}, 0.2\text{cm})$ with its long axis oriented in vertical direction. The mesh size and time step of this simulation are $h = 0.0125$cm and $\Delta t = 0.001$s. The interactions between the ellipsoid and the boundary of the computational domain are modelled by defining a collision zone of width $\rho = 0.04$cm and a stiffness parameter $\epsilon_{\mathcal{F}} = 3 * 10^{-6}$. The evolution of the horizontal position of the ellipsoid is shown in Figure 4, and is compared to literature results [17]. The trajectories before and after the first interac-



(a) Trajectory (b) Trajectory with symmetric results

Figure 3: The left graph shows the trajectory of the center of mass of the ellipse. The red line corresponds to the present results and the blue dotted line represents the results in the literature [29]. The right graph shows the same trajectory but compares it to rotations happening on the other wall.

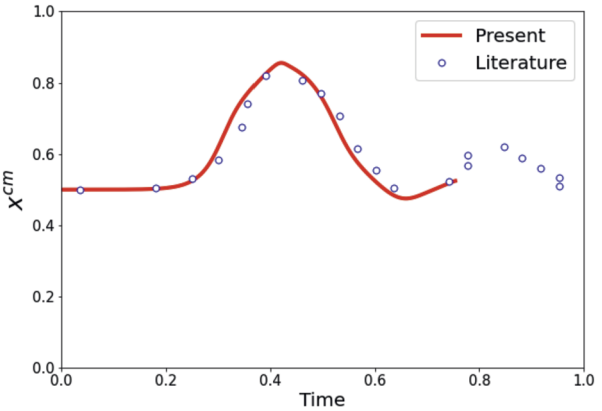


Figure 4: Evolution of the horizontal position of the center of the ellipsoid. The dotted line shows the literature results [17] and the red line represents the present results.

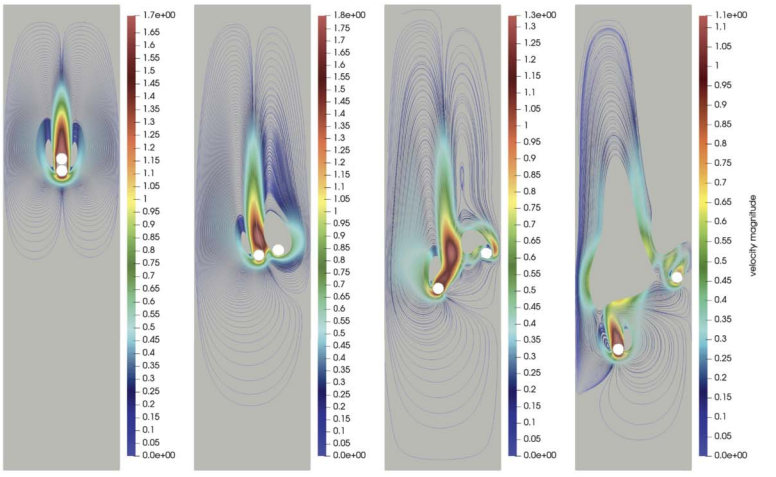
tion between the ellipsoid and the boundary of the computational domain, at time $t \approx 0.45$s, are in good agreement.

5.2. Multiple objects

Two disks falling in an incompressible fluid. Collisions between two spherical bodies placed on a same vertical line and falling under the effect of

gravity, in a channel filled with incompressible Newtonian fluid, are known as the *drafting, kissing and tumbling* phenomenon. We consider two disks of radius $r = 0.1$cm and density $\rho_{\mathcal{S}} = 1.01 \frac{\text{g}}{\text{cm}^3}$ placed in a fluid domain of dimensions $[0\text{cm}, 2\text{cm}] \times [0\text{cm}, 8\text{cm}]$. To have a slight asymmetry, necessary to speed up the insurgence of the phenomenon, the coordinates of the upper disk $\mathcal{S}_1$ center are fixed at $(0.999\text{cm}, 7.2\text{cm})$ and those of the lower disk $\mathcal{S}_2$ center at $(1.0\text{cm}, 6.8\text{cm})$. The fluid density and viscosity are respectively given by $\rho_f = 1.0 \frac{\text{g}}{\text{cm}^3}$ and $\mu = 0.01 \frac{\text{g}}{\text{cm s}}$. The simulation is performed on the time interval $[0\text{s}, 5\text{s}]$ with a time step equal to $\Delta t = 0.001$s and a mesh size set to $h = 0.01$. To model the *drafting, kissing and tumbling* phenomenon as in [6], one fixes the width of the collision zone at $\rho = 0.0225$cm and uses stiffness parameters for the disk-disk and disk-fluid domain collisions given by $\epsilon = 7 * 10^{-5}$ and $\epsilon_{\mathcal{F}} = 5 * 10^{-5}$. Figure 5 shows the results of the simulation. When both disks are next to each other, the motion of the lower disk reduces the fluid pressure behind it and therefore the resistance of the fluid for the upper disk. As a result, the upper body falls faster until it collides with the lower one. The collision forces cause the separation of the disks which then move in opposite directions. Figure 5 also compares the vertical and horizontal position of the two disks over time to literature [6]. It can be observed that the results of the collision phase have a similar behavior. In the present simulation, the bodies fall faster to the bottom, therefore the separation phase takes place sooner than in literature results. This explains the differences on the graphs.

100 *disks falling in an incompressible fluid.* We simulate the interactions between 100 circular particles in two dimensions. The particles of radius $r = 0.03125$cm and density $\rho_{\mathcal{S}} = 1.1 \frac{\text{g}}{\text{cm}^3}$ are placed in a vertical channel of height 2cm and width 1cm which is filled with a fluid of density $\rho_f = 1.0 \frac{\text{g}}{\text{cm}^3}$ and viscosity $\mu = 0.01 \frac{\text{g}}{\text{cm s}}$. At the beginning of the simulation, the fluid and the particles are at rest. Then the particles move towards the bottom of the domain under the effect of gravity. The initial configuration is depicted in the first image of Figure 6. We run the simulation on the time interval $[0\text{s}, 5.5\text{s}]$ and use a time step of $\Delta t = 0.002$s. The mesh size is given by $h = 0.01$cm. The collision force is defined on range $\rho = 0.025$cm. According to simulations of multiple objects in the literature [18], the intensity of the force must be important to prevent the objects from overlapping. For this reason, we use stiffness parameters fixed at $\epsilon = 5 * 10^{-6}$ and $\epsilon_{\mathcal{F}} = 10^{-7}$. Figure 6 shows the position of the 100 particles at four time instants. Due to the collision forces, the particles near the domain walls fall slower towards the bottom than the central particles. During the simulation the particles



(a) $t = 1.5s$ (b) $t = 2.5s$ (c) $t = 3.0s$ (d) $t = 4.0s$

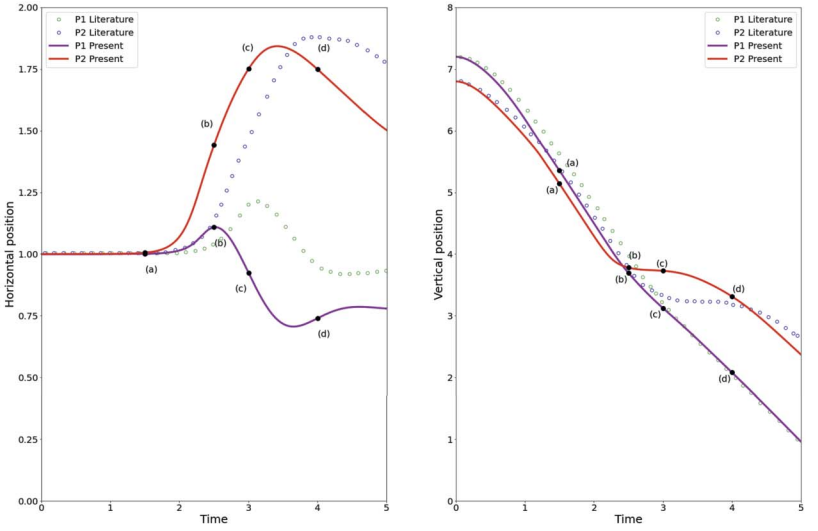


Figure 5: The upper figures show the position of the two disks as well as the streamlines at four different time instants. The lower graphs compare the evolution of the disks horizontal and vertical position to literature [6]. The solid lines correspond to the present results and the dotted ones to literature results.

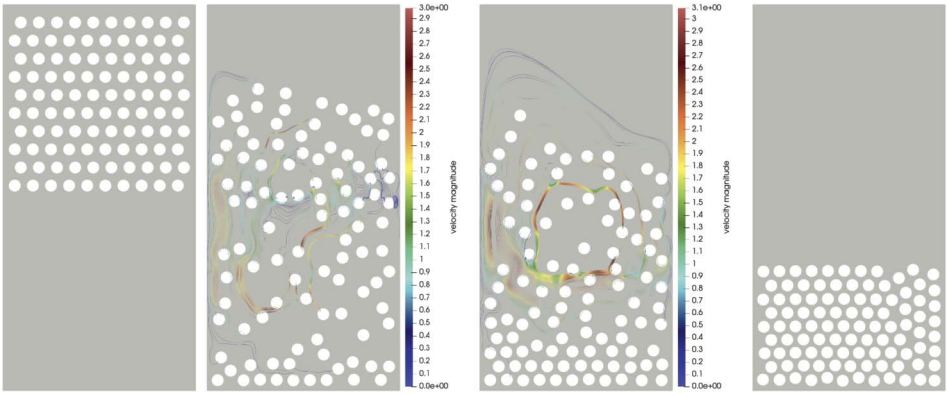


Figure 6: Streamlines and position at four different time instants of 100 disks falling under the effect of gravity in an incompressible Newtonian fluid.

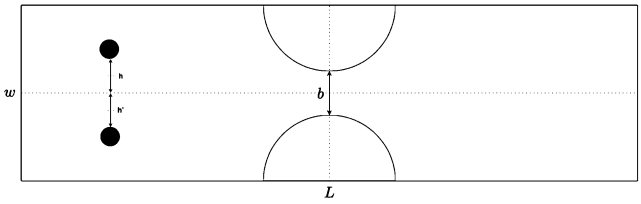


Figure 7: Geometry of symmetric stenotic artery.

settle one onto the other on the bottom of the channel. At final time, almost all particles are in a stationary position and packed in a hexagonal lattice. Article [10] conducts a similar simulation but utilizes different parameter values.

Two objects in a Stenotic Artery Flow. For this application we consider the motion of two circular particles in a symmetric stenotic artery in two dimensions. In contrast with the previous test cases, the particles do not fall under the effect of gravity, but the motion of the particles is due to a pressure difference between the inlet and the outlet. This simulation is presented in [22] and [13].

The geometry, given in Figure 7, is a channel of length $L = 32d$ and width $w = 8d$. The centerlines are represented by dotted lines. The diameter d of the particles is set to $d = 8.5 * 10^{-4}$cm. To create the stenosis, the authors of [22] and [13] add two semicircular symmetrical protuberances. The radius

of these protuberances depends on the width of the stenosis throat, which is fixed at $b = 1.75d$. The fluid viscosity is set to $\mu = 0.01\frac{\text{g}}{\text{cm s}}$ and both fluid and particle densities are equal to $\rho_f = \rho_{\mathcal{S}} = 1\frac{\text{g}}{\text{cm}^3}$. At time $t = 0$, the fluid and the particles are at rest. Then the particles move towards the protuberances due to the pressure difference $\Delta p = 541\text{Pa}$ between the inlet and outlet of the channel. The two particles are initially located at $8d$ to the left of the vertical centerline and asymmetrically with respect to the horizontal centerline: $h = 2d + \frac{d}{4000}$ and $h' = 2d$. This slight asymmetry allows them to cross the stenosis throat. The mesh size and time step of the simulation are respectively set to $h = 6.07 * 10^{-5}\text{cm}$ and $\Delta t = 10^{-5}\text{s}$. The time interval is $[0.0\text{s}, 0.01\text{s}]$. Collision forces are applied when the particles are close to the protuberances. The width of the collision zone is fixed at $\rho = 0.0003\text{cm}$ and the stiffness parameters of the particle-particle and particle-domain interactions are given by $\epsilon = 10^{-11}$ and $\epsilon_{\mathcal{F}} = 7.5 * 10^{-12}$. Figure 8 shows the trajectory of the two particles: they have a similar motion (snapshots 1-9) until they get close to the stenosis throat (snapshot 10). Since the width of the stenosis throat is too small to allow both particles to pass simultaneously, the upper particle stops and changes direction due to collision forces. Snapshot 11 shows that it moves back to allow the lower particle to pass the throat. Finally, the upper particle follows the lower particle through the artery (snapshots 11-17). The same particle behavior is present in the cited articles. The particle trajectory after passing the stenosis throat depends on the initial configuration and the definition of the collision parameters.

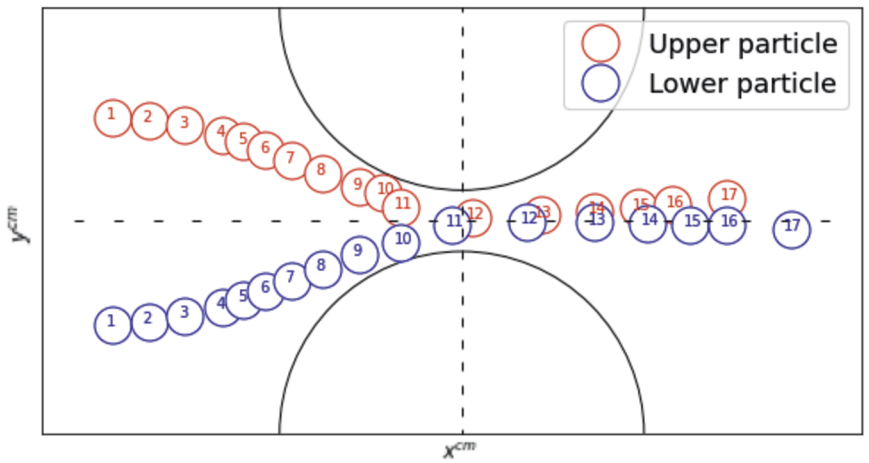


Figure 8: Position of two particles at different time instants.

Three-sphere swimmer in a Stokes flow. For the last test case, we consider a tilted three-sphere swimmer placed close to the boundary of a horizontal channel in Stokes regime. The radius of the spheres is set to $r = 1$cm and the length of the rods to $l = 10$cm. The height and width of the computational domain are fixed at $h = 150$cm and $w = 40$cm. The center of mass of the central sphere is initially located at $(15\text{cm}, 10\text{cm})$, and the density of the swimmer is given by $\rho_{\mathcal{S}} = 0.1\frac{\text{g}}{\text{cm}^3}$. The fluid has a density equal to $\rho_f = 1\frac{\text{g}}{\text{cm}^3}$ and a viscosity of $\mu = 1\frac{\text{g}}{\text{cm s}}$. The mesh size is set to $h = 0.3$cm and the time step to $\Delta t = 0.1$s. Due to the orientation of the swimmer at $t = 0.0$s and due to its swimming strategy, leading to a straight motion, the swimmer gets closer to the boundary of the channel. Once its right sphere arrives in the collision zone of width $\rho = 0.03$cm, lubrication forces, defined by a stiffness parameter set to $\epsilon_{\mathcal{F}} = 5 * 10^{-8}$, are applied to the swimmer, who starts to change direction. The swimmer continues rotating until its left sphere reaches the collision zone. Here, the repulsive force acting on this sphere propels the swimmer upwards, distancing it from the boundary. Figure 9 captures the behavior of the three-sphere swimmer as it approaches and interacts with the boundary.

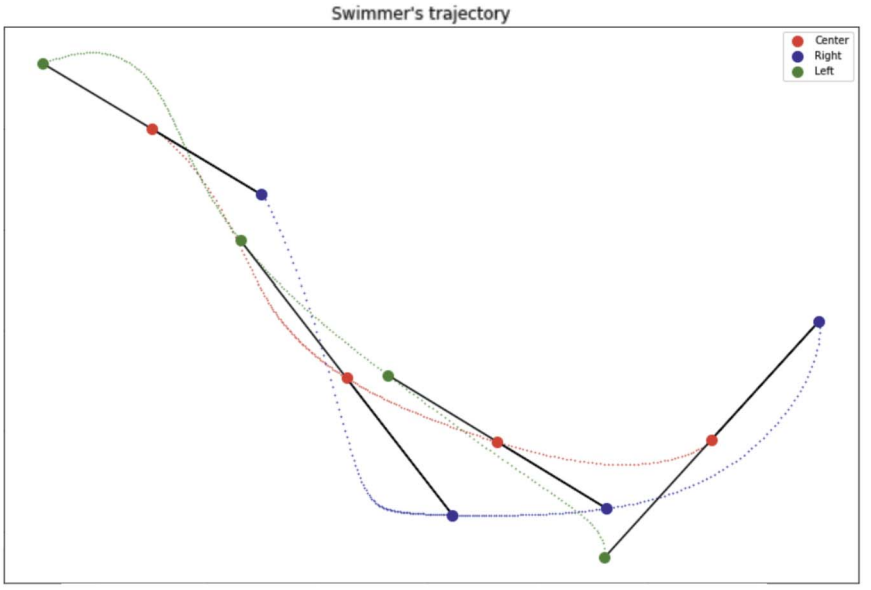


Figure 9: Behavior of the three-sphere swimmer close to the boundary of the computational domain.

Reproducibility

The validation benchmarks and all applications illustrated in this paper are available in a public GitHub repository [20]. All the results can be reproduced.

6. Conclusions

In this paper, we present a repulsive collision-avoidance model for arbitrarily shaped bodies, where the force direction and magnitude are based on the computation of the distance function using the narrow-band fast marching method. We have validated the model against established literature results and observed a strong agreement. Various applications to two and three-dimensional cases are illustrated. Within the appendix, we introduce a tool for the insertion of solid bodies into meshed domains, which allows simulations where the conforming treatment of the fluid-solid interface is required. Determining the optimal values for the collision parameters is still an open problem which could be tackled by anticipating the moments of collision and controlling the displacement of the bodies. We aim to extend our work to encompass elastic bodies and collisions involving multiple contact points in a forthcoming publication.

Appendix A. Insertion of arbitrary solids into discretized fluid domains

In certain biological processes, like particle transport within blood vessels, solid bodies navigate through geometrically intricate domains. For numerical simulations, reconstructing these environments usually leans more towards image-based methods rather than traditional CAD descriptions, which complicates the mesh-fitted definition of solid bodies inside these environments. Consequently, we developed a tool allowing the incorporation of arbitrary bodies into meshed domains, identifying both their geometric and material properties. The tool relies on the MMG library [2] that allows remeshing from a level-set function: given the initial triangulated environment and a level-set description of the solid body, it outputs a new triangulation that includes the solid, and where the interface between the two is conforming. Furthermore, our tool has the capability to simultaneously introduce multiple objects with predetermined positions and orientations, ensuring that physical markers associated with the fluid and the solids are correctly transferred onto the new mesh.

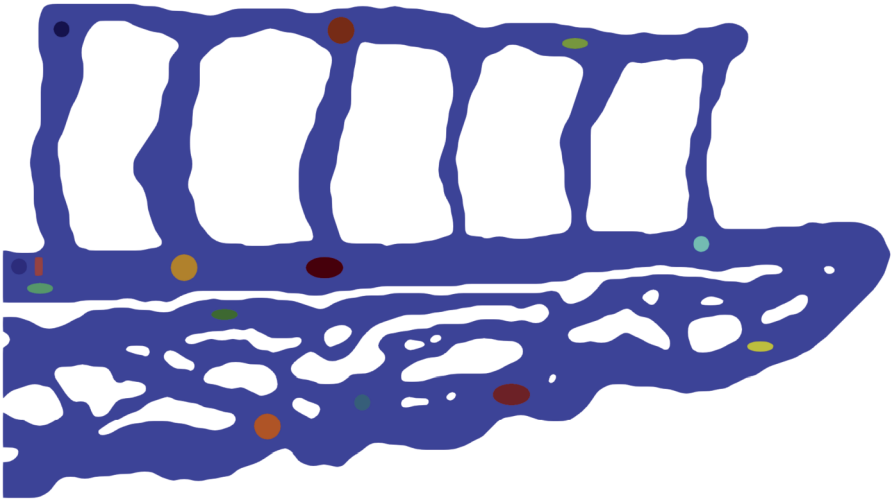
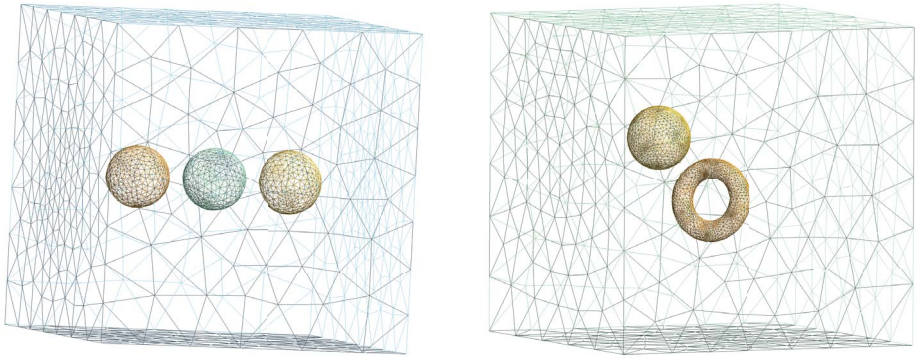


Figure 10: Insertion of two-dimensional bodies inside a complex discretized environment.



(a) Three-sphere swimmer

(b) Complex-shaped bodies

Figure 11: Two cases showing the insertion of bodies in three dimensions.

The capabilities of the tool are illustrated in Figures 10 and 11. Figure 10 shows a two-dimensional environment of complex geometry where a collection of objects have been successfully inserted without changing their shapes. In particular, all corners of the rectangular body have been recovered. Figure 11 illustrates a simpler three-dimensional domain where multiple bodies have been again successfully inserted. As detailed in the first part of the pa-

per, all three generated meshes are apt for simulating fluid-solid interactions with collision treatments.

Acknowledgements

The authors acknowledge the financial support of the French Agence Nationale de la Recherche (grant ANR-21-CE45-0013 project NEMO), ITI Irmia++, and Cemosis. In addition, the authors thank A. Froehly and L. Cirrottola for their helpful comments and discussions regarding the MMG library.

References

[1] Ardekani, M. N., Costa, P., Breugem, W.-P. and Brandt, L. (2016). Numerical Study of the Sedimentation of Spheroidal Particles. *International Journal of Multiphase Flow* **87** 16–34. MR3553759

[2] Balarac, G., Basile, F., Bénard, P., Bordeu, F., Chapelier, J. B., Cirrottola, L., Caumon, G., Dapogny, C., Frey, P., Froehly, A., Ghigliotti, G., Laraufie, R., Lartigue, G., Legentil, C., Mercier, R., Moureau, V., Nardoni, C., Pertant, S. and Zakari, M. (2022). Tetrahedral remeshing in the context of large-scale numerical simulation and high performance computing. *MathematicS In Action* **11** 129–164.

[3] Berti, L., Chabannes, V., Giraldi, L. and Prud'homme, C. Fluid-rigid body interaction using the finite element method and ALE formulation: framework, implementation and benchmarking. preprint.

[4] Broms, A. and Tornberg, A.-K. (2023). A Barrier Method for Contact Avoiding Particles in Stokes Flow. preprint.

[5] Ericson, C. (2004). *Real-time collision detection*. Crc Press.

[6] Feng, Z.-G. and Michaelides, E. E. (2004). The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. *Journal of Computational Physics* **195** 602–628.

[7] Formaggia, L. and Quarteroni, A., eds. (2009). *Cardiovascular mathematics: modeling and simulation of the circulatory system. MS & A: modeling, simulation & applications* **1**. Springer, Milano. MR2522375

[8] Glowinski, R. (2003). *Finite element methods for incompressible viscous flow. Handbook of Numerical Analysis* **9**. Elsevier. MR2009826

[9] Jain, R., Tschisgale, S. and Fröhlich, J. (2019). A collision model for DNS with ellipsoidal particles in viscous fluid. *International Journal of Multiphase Flow* **120** 103087. MR3993133

[10] Juárez, L. H., Glowinski, R. and Pan, T. (2004). Numerical simulation of fluid flow with moving and free boundaries. *SeMA Journal: Boletín de la Sociedad Española de Matemática Aplicada* **30** 49–102.

[11] Kanchi, H. and Masud, A. (2007). A 3D adaptive mesh moving scheme. *International Journal for Numerical Methods in Fluids* **54** 923–944. MR2333015

[12] Lefebvre, A. (2009). Numerical simulation of gluey particles. *ESAIM: Mathematical Modelling and Numerical Analysis* **43** 53–80. MR2494794

[13] Li, H., Fang, H., Lin, Z., Xu, S. and Chen, S. (2004). Lattice Boltzmann simulation on particle suspensions in a two-dimensional symmetric stenotic artery. *Physical Review E* **69** 031919.

[14] Maury, B. (1999). Direct Simulations of 2D Fluid-Particle Flows in Biperiodic Domains. *Journal of Computational Physics* **156** 325–351. MR1727335

[15] Muth, B., Müller, M.-K., Müller, M., Eberhard, P. and Luding, S. (2007). Collision Detection and Administration Methods for Many Particles with Different Sizes. In *4th International Conference on Discrete Element Methods, DEM 4* 1–18. Minerals Engineering Int.

[16] Najafi, A. and Golestanian, R. (2004). A Simplest Swimmer at Low Reynolds Number: Three Linked Spheres. *Physical Review E* **69** 062901.

[17] Pan, T.-W., Glowinski, R. and Galdi, G. P. (2002). Direct simulation of the motion of a settling ellipsoid in Newtonian fluid. *Journal of Computational and Applied Mathematics* **149** 71–82. MR1952967

[18] Pan, T. W., Joseph, D. D., Bai, R., Glowinski, R. and Sarin, V. (2002). Fluidization of 1204 spheres: simulation and experiment. *Journal of Fluid Mechanics* **451** 169–191. MR1886011

[19] Pouchol, M., Ahmad, A., Crespin, B. and Terraz, O. (2009). A hierarchical hashing scheme for nearest neighbor search and broad-phase collision detection. *Journal of Graphics, GPU, and Game Tools* **14** 45–59.

[20] Prud'homme, C., Chabannes, V., Metivet, T., Hild, R., Trophime, Abdoulaye, S., Saigre, T., Ricka, P., Berti, L. and Van Landeghem, C. feelpp/feelpp.

[21] Sethian, J. A. (1996). A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences* **93** 1591–1595. MR1374010

[22] Shu, J. W. a. C. (2010). Particulate Flow Simulation Via a Boundary Condition-enforced Immersed Boundary-lattice Boltzmann Scheme. *Communications in Computational Physics* **7** 793–812.

[23] Singh, P., Hesla, T. I. and Joseph, D. D. (2003). Distributed Lagrange multiplier method for particulate flows with collisions. *International Journal of Multiphase Flow* **29** 495–509.

[24] Usman, K., Walayat, K., Mahmood, R. and Kousar, N. (2018). Analysis of solid particles falling down and interacting in a channel with sedimentation using fictitious boundary method. *AIP Advances* **8** 065201.

[25] Vemuri, B. C., Chen, L., Vu-Quoc, L., Zhang, X. and Walton, O. (1998). Efficient and Accurate Collision Detection for Granular Flow Simulation. *Graphical Models and Image Processing* **60** 403–422.

[26] Wachs, A., Uhlmann, M., Derksen, J. and Huet, D. P. (2023). Modeling of short-range interactions between both spherical and non-spherical rigid particles. In *Modeling Approaches and Computational Methods for Particle-Laden Turbulent Flows* 217–264. Elsevier. MR3951718

[27] Wan, D. and Turek, S. (2006). Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method. *International Journal for Numerical Methods in Fluids* **51** 531–566. MR2227587

[28] Wang, L., Guo, Z. L. and Mi, J. C. (2014). Drafting, kissing and tumbling process of two particles with different sizes. *Computers & Fluids* **96** 20–34. MR3198060

[29] Xia, Z., Connington, K. W., Rapaka, S., Yue, P., Feng, J. J. and Chen, S. (2009). Flow patterns in the sedimentation of an elliptical particle. *Journal of Fluid Mechanics* **625** 249–272.

[30] Zamora, E., Battaglia, L., Storti, M., Cruchaga, M. and Ortega, R. (2019). Numerical and experimental study of the motion of a sphere in a communicating vessel system subject to sloshing. *Physics of Fluids* **31** 087106.

Céline Van Landeghem
Cemosis
IRMA UMR 7501
CNRS
Université de Strasbourg
France
E-mail address: celine.vanlandeghem@math.unistra.fr

Luca Berti
Cemosis
IRMA UMR 7501
CNRS
Université de Strasbourg
France
E-mail address: lberti@unistra.fr

Vincent Chabannes
Cemosis
IRMA UMR 7501
CNRS
Université de Strasbourg
France
E-mail address: chabannes@unistra.fr

Agathe Chouippe
ICube Laboratory
UMR 7357
Université de Strasbourg
France
E-mail address: chouippe@unistra.fr

Laetitia Giraldi
CALISTO team
INRIA
Université Côte d'Azur
France
E-mail address: laetitia.giraldi@inria.fr

Yannick Hoarau
ICube Laboratory
UMR 7357
Université de Strasbourg
France
E-mail address: hoarau@unistra.fr

Christophe Prud'homme
Cemosis
IRMA UMR 7501
CNRS
Université de Strasbourg
France
E-mail address: prudhomme@unistra.fr

Received May 16, 2023