

Validation of an immersed thick boundary method for simulating fluid–structure interactions of deformable membranes

Julien Sigüenza, Simon Mendez, Dominique Ambard, Frédéric Dubois, Franck Jourdan, Rémy Mozul, Franck Nicoud

▶ **To cite this version:**

Julien Sigüenza, Simon Mendez, Dominique Ambard, Frédéric Dubois, Franck Jourdan, et al.. Validation of an immersed thick boundary method for simulating fluid–structure interactions of deformable membranes. Journal of Computational Physics, 2016, 322, pp.723-746. 10.1016/j.jcp.2016.06.041 . hal-01348721

HAL Id: hal-01348721
https://hal.science/hal-01348721

Submitted on 25 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Validation of an immersed thick boundary method for simulating fluid-structure interactions of deformable membranes

J. Sigüenza[a,*], S. Mendez[a], D. Ambard[b], F. Dubois[b], F. Jourdan[b], R. Mozul[b], F. Nicoud[a]

[a] *Institut Montpelliérain Alexander Grothendieck (IMAG), UMR CNRS 5149, University of Montpellier*
[b] *Laboratoire de Mécanique et Génie Civil (LMGC), UMR CNRS 5508, University of Montpellier*

Abstract

This paper constitutes an extension of the work of Mendez, Gibaud & Nicoud: An unstructured solver for simulations of deformable particles in flows at arbitrary Reynolds numbers, Journal of Computational Physics, 256(1): 465-483 (2014), for three-dimensional simulations of deformable membranes under flow. An immersed thick boundary method is used, combining the immersed boundary method with a three-dimensional modeling of the structural part. The immersed boundary method is adapted to unstructured grids for the fluid resolution, using the reproducing kernel particle method. An unstructured finite-volume flow solver for the incompressible Navier-Stokes equations, is coupled with a finite-element solver for the structure. The validation process relying on a number of test cases proves the efficiency of the method, and its robustness is illustrated when computing the dynamics of a tri-leaflet aortic valve. The proposed immersed thick boundary method is able to tackle applications involving both thin and thick membranes/closed and open membranes, in significantly high Reynolds number flows and highly complex geometries.

Keywords: Fluid-structure interaction, Immersed boundary method, Thick membranes, Finite-element method, Unstructured fluid solver

*Corresponding author
Email address: julien.siguenza@umontpellier.fr (J. Sigüenza)

1. Introduction

Solving the fluid-structure interaction (FSI) problem involved when a membrane is deformed by a flow is a scientific challenge which has been tackled for several decades, due to its wide range of applications. When dealing with numerical simulation of flow-induced deformation of membranes, the state of the art is extremely varied. Different communities work on the topic, focusing on various applications. A large part of these applications are considering deformable particles such as capsules, vesicles or cells. All these systems are constituted by a liquid droplet enclosed by a very thin structure (its thickness is much smaller than the size of the object). This structure can be a polymer structure for capsules, a phospholipid bilayer for vesicles, or a more complex biological membrane in the case of red blood cells [1]. Due to their small size, computations of flows of these deformable particles are often based on boundary integral methods (BIM) [2]. This method can be used for low Reynolds number flows, when the flow is well described by the Stokes equations. The BIM is a very popular technique to compute flows of capsules [3, 4, 5, 6, 7, 8, 9], vesicles [10, 11, 12, 13, 14] and red blood cells [15, 16, 17], because of its precision and its relatively moderate computational cost (only the membranes and boundaries need to be discretized). When the flow is governed by the Navier-Stokes equations, methods with the fluid grid following the deformation of the interface, based on the Arbitrary Lagrangian-Eulerian (ALE) formalism are developed [18, 19, 20]. However, they are not the most popular, as they involve frequent remeshing. Authors generally prefer one-fluid formalisms, where the fluid equations are solved everywhere, over a fixed Eulerian grid. The membrane location is computed by advecting either a function as the level-set function or a second Lagrangian grid following the membrane displacement. One can cite the advected-field approach [21, 22], level-set methods [23, 24, 25, 26] and immersed boundary or immersed interface methods [27, 28, 29, 30, 31, 32, 33, 34].

The immersed boundary method (IBM) was originally developed by Peskin

et al. [35] and has since been extensively studied and applied to a wide variety of FSI problems. In a previous work, Mendez *et al.* [36] used the IBM to study fluid-structure interaction of deformable particles in flows at arbitrary Reynolds numbers, in the context of complex geometries often encountred when dealing with medical artificial devices. Since this work was only considering the 2D resolution, the main purpose of the present paper is to extend it to 3D. The IBM being originally developed to deal with zero volume structures, a membrane-like structure with an infinitely thin thickness can be considered, neglecting the bending stiffness of the membrane. Although suitable when modeling very thin capsules under flow [37, 38], this approach reaches its limitations when considering membranes having a significant bending rigidity. To capture bending effects, an additional model based on the Helfrich energy [39] can be introduced (also used by Mendez *et al.* [36]), and has been notably used to model flowing capsules and red blood cells [40, 41]. Another approach is the one introduced by Le and Tan [32], where the IBM is combined with a thin-shell model to simulate the deformation of liquid capsules under flow [32, 42, 33].

In the present paper, another approach to simulate flowing deformable membranes in the context of the IBM is proposed, the immersed thick boundary method (ITBM). This approach is inspired from the extended immersed boundary method (EIBM), introduced by Wang and Liu [29], and later adapted to the immersed finite element method (IFEM) [43, 44]. Instead of the volumeless immersed boundary, a submerged solid which occupies a finite volume within the fluid domain is considered. This approach constitutes an alternative to the one consisting in combining the IBM with a thin shell model. Indeed, rather than representing implicitly the thickness of the membrane using a thin shell model (as Dupont *et al.* [45]), the thickness is represented in an explicit manner, modeling the membrane as a 3D continuum using the classical finite-element method. A full description of the ITBM is done in section 2. The use of a classical finite-element framework in the IBM is not generally employed for thin membranes. As a consequence, the present paper presents an extensive validation work in section 3, in order to thoroughly determine the real limitations

of such an approach. The case of very thin membranes will be treated with particular attention. Note that the present approach is not limited to closed membranes. Both closed membranes and open membranes can be simulated using the ITBM, and an application to the flow through a tri-leaflet aortic valve is presented in section 4.

2. The immersed thick boundary method

In the IBM framework, two independent meshes are considered to discretize the solid domain Ω_s and the fluid domain Ω_f. The solid is discretized by a moving Lagrangian mesh, and the fluid is discretized by a fixed Eulerian mesh, which can be either structured or unstructured (Fig. 1). The different steps of the IBM are the following, as introduced by Peskin [27]:

(1) Knowing the displacement $\overrightarrow{U_m}$ of each solid node, the mechanical force $\overrightarrow{F_m}$ resulting from the membrane deformation is calculated.

(2) The mechanical force $\overrightarrow{F_m}$ is regularized on the fluid mesh, giving the volumetric force $\overrightarrow{f_j}$ on each fluid node.

(3) The Navier-Stokes equations (forced by the regularized mechanical forces) are solved on the fluid mesh, yielding the velocity of the fluid $\overrightarrow{v_j}$ on each fluid node.

(4) The velocity of the membrane $\overrightarrow{V_m}$ on each solid node is interpolated from the $\overrightarrow{v_j}$ field, enabling to deduce the new position $\overrightarrow{X_m}$ from the position at the previous timestep $\overrightarrow{X_m}^{previous}$, such as $\overrightarrow{X_m} = \overrightarrow{X_m}^{previous} + \Delta t \overrightarrow{V_m}$. The displacement is then updated $\overrightarrow{U_m} = \overrightarrow{X_m} - \overrightarrow{X_m^0}$, where $\overrightarrow{X_m^0}$ stands for the initial stress-free position, also referred to as the reference position.

In the present study, step (1) is performed by the LMGC90 solid mechanics solver [46], while steps (2)-(4) are performed by the YALES2BIO numerical tool [36, 47, 48], based on a massively parallel unstructured finite volume flow solver for the incompressible Navier-Stokes equations [49]. Note however that steps (2) and (4) could be handled by a dedicated coupling program, in the case

where modifications in the fluid solver cannot be easily performed. YALES2BIO being an in-house solver, this option was not considered further.

The main distinction between the IBM and the ITBM is that instead of having a cluster of solid nodes spread over a surface, the cluster defines a volume mesh (see Fig. 1). The different steps of the ITBM are detailed below.

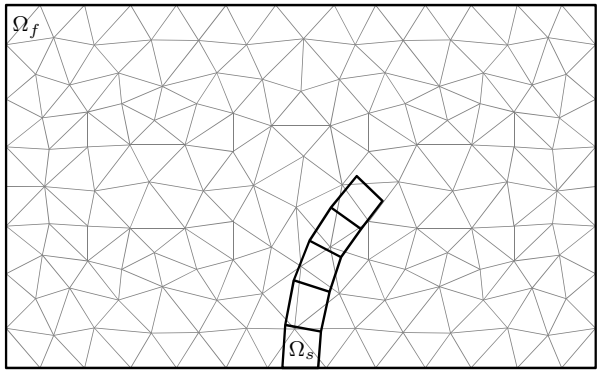


Figure 1: Schematic representation of the discretized problem in 2D.

2.1. Computation of the mechanical force

When considering immersed volumetric objects, as this is the case with the EIBM [29], the actual structural force to regularize is commonly composed of the internal mechanical force $\overrightarrow{F_m}$ resulting from the static deformation of the structure, and the inertial force $M\overrightarrow{\ddot{U}_m}$ resulting from the dynamics of the structure, the mass matrix M being written as:

$$M = (\rho_s - \rho)\, V, \tag{1}$$

where ρ_s and ρ stand for the solid and fluid densities, respectively, and V the volume matrix. When regularizing the structural force, the inertial force should thus be neglected as long as the fluid-to-solid density ratio is close to unity ($\rho_s \simeq \rho$). This assumption is even more true when the solid phase is a very thin membrane ($V \simeq 0$).

The mechanical force $\overrightarrow{F_m}$ is calculated over the three-dimensional mesh representing the solid domain, thanks to the classical finite-element method. Let

σ be the cauchy stress tensor, and ϵ the Eulerian-Almansi strain tensor. Using the virtual works principle, it is possible to identify:

$$\overrightarrow{F_m}.\overrightarrow{U_m} = \int_{\Omega_s} \sigma : \epsilon \, d\Omega. \tag{2}$$

By using the transformation between the current and the reference configuration of the domain Ω_s^0, the total Lagrangian formulation of the previous equation gives:

$$\overrightarrow{F_m}.\overrightarrow{U_m} = \int_{\Omega_s^0} \mathbb{S} : \mathbb{E} \, d\Omega^0, \tag{3}$$

where $\mathbb{S}$ is the 2nd Piola-Kirchhoff stress tensor, and $\mathbb{E}$ is the Green-Lagrangian strain tensor, which reads:

$$\mathbb{E} = \frac{1}{2} \left(\mathbb{C} - \mathbb{I} \right), \tag{4}$$

with $\mathbb{C} = \mathbb{F}^T \mathbb{F}$ the Right Cauchy-Green deformation tensor, and $\mathbb{F} = \mathbb{I} + \nabla \overrightarrow{U_m}$ the deformation gradient tensor. The right-hand side of Eq. (3) can be rewritten as the scalar product of the equivalent vector forms of $\mathbb{S}$ and $\mathbb{E}$:

$$\overrightarrow{F_m}.\overrightarrow{U_m} = \int_{\Omega_s^0} \overrightarrow{S} . \overrightarrow{E} \, d\Omega^0. \tag{5}$$

Introducing the gradient matrix $\mathcal{B}$ which contains the spatial derivatives of the shape functions, Eq. (5) becomes:

$$\overrightarrow{F_m}.\overrightarrow{U_m} = \int_{\Omega_s^0} \overrightarrow{S} . \, \mathcal{B} \, \overrightarrow{U_m} \, d\Omega^0. \tag{6}$$

Eliminating $\overrightarrow{U_m}$ on both sides and discretizing the domain as a sum of elements $\Omega_s^0 = \cup \, \Omega_{el}^0$, one finally obtains:

$$\overrightarrow{F_m} = \sum_{\Omega_{el}^0} \left[\int_{\Omega_{el}^0} \overrightarrow{S} . \, \mathcal{B} \, d\Omega^0 \right]. \tag{7}$$

In the present paper, both prismatic linear elements (see Fig. 2, left) and hexaedral quadratic elements (see Fig. 2, right) are tested. In the case of linear elements, the linear shape function is defined between two elements nodes. And in the case of quadratic elements, an interpolation node between two element nodes is needed to define the quadratic shape function. The effective resolution

H of the elements is introduced, as being the biggest distance between two nodes (between two element nodes in the case of linear elements, and between one element node and one interpolation node in the case of quadratic elements). Note that in the case of linear elements, the effective resolution H is equal to the element resolution, and to half of the element resolution in the case of quadratic elements. If the membrane thickness e is smaller than the element resolution (as this is the case in Fig. 2), the membrane is meshed only with one element in the thickness. But if the element resolution is smaller than the membrane thicknes e, the number of elements in the thickness is greater than one.

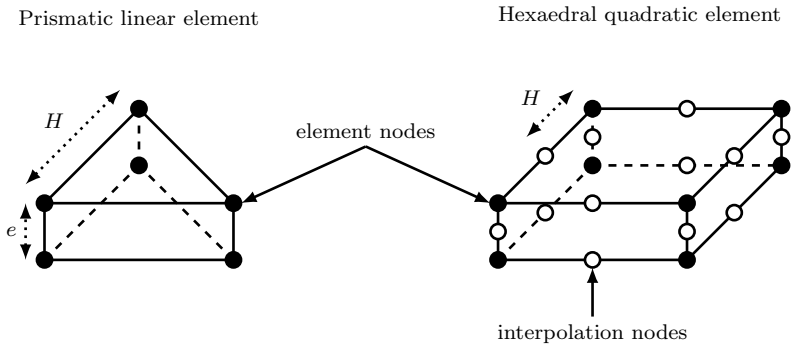


Figure 2: **Left:** Prismatic linear element where linear shape functions are defined between two element nodes. **Right:** Hexaedral quadratic element. Interpolation nodes are needed to define the quadratic shape functions.

Constitutive laws

The LMGC90 computational code features a wide library of constitutive laws. Only a few of them are used in the present paper:

- The compressible Saint-Venant Kirchoff law

$$\mathbb{S} = \frac{2G\nu}{1-2\nu}\,\mathrm{Tr}\,(\mathbb{E})\,\mathbb{I} + 2G\mathbb{E}, \tag{8}$$

where G and ν are the shear modulus and the Poisson coefficient, respectively.

- The compressible Neo-Hookean law, as introduced by Simo and Pister [50]

$$\begin{cases} \mathbb{S} = 2\dfrac{\partial W}{\partial \mathbb{C}} \\ \\ W = \dfrac{G}{2}\left(I_1 - 3\right) - G\ln J + \dfrac{G\nu}{1-2\nu}\left(\ln J\right)^2 \end{cases} \tag{9}$$

where W is the strain energy density function. $J = \det \mathbb{F} = \lambda_1\lambda_2\lambda_3$ is the Jacobian of the transformation, expressed as the product of the principal stretches λ_i and I_1 is the first invariant of the Right Cauchy-Green deformation tensor $\mathbb{C}$, which reads:

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2. \tag{10}$$

- The quasi-incompressible Yeoh law

$$\begin{cases} \mathbb{S} = 2\dfrac{\partial W}{\partial \mathbb{C}} \\ \\ W = C_1\left(\bar{I}_1 - 3\right) + C_2\left(\bar{I}_1 - 3\right)^2 + C_3\left(\bar{I}_1 - 3\right)^3 + \dfrac{G\left(1+\nu\right)}{3\left(1-2\nu\right)}\left(\ln J\right)^2 \end{cases} \tag{11}$$

where C_i are material constants and $\bar{I}_1$ is the first invariant of the isochoric Right Cauchy-Green deformation tensor $\bar{\mathbb{C}}$, such as $\bar{I}_1 = J^{-2/3}I_1$. The quantity $2C_1$ can be interpreted to be the shear modulus G, and when C_2 and C_3 are chosen to be $C_2 = C_3 = 0$, this leads to the quasi-incompressible Neo-Hookean law.

2.2. Forces regularization

The force per unit volume applied on the fluid by the membrane is computed from the mechanical forces on the solid nodes, through the process of force regularization:

$$\overrightarrow{f_j} = \sum_{m=1}^{M}\left[\overrightarrow{F_m} \times w\left(\frac{||\overrightarrow{x_j} - \overrightarrow{X_m}||}{h}\right)\right], \tag{12}$$

where w is a discrete Dirac function allowing to regularize the mechanical force at the neighboring fluid nodes from the mechanical force at the solid node location (Fig. 3).

When using a regular Cartesian mesh of constant mesh size h, the discrete Dirac function w can be easily defined as the product of one-dimensional delta

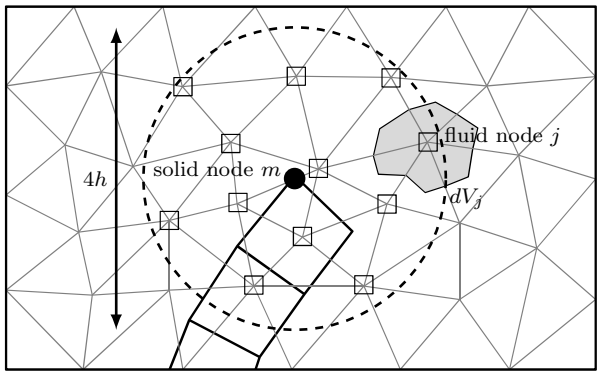


Figure 3: Schematic representation of the procedure to compute the window function w.

functions:

$$w\left(\frac{||\overrightarrow{x_j}-\overrightarrow{X_m}||}{h}\right)=D\left(\frac{x_j-X_m}{h}\right)D\left(\frac{y_j-Y_m}{h}\right)D\left(\frac{z_j-Z_m}{h}\right). \tag{13}$$

The cosine representation is often used, as introduced by Peskin [27]:

$$D(r)=\begin{cases}\dfrac{1}{4h}\left[1+\cos\left(\dfrac{\pi r}{2}\right)\right] & \text{if } |r|<2\\[2ex] 0 & \text{if } |r|\geq 2\end{cases} \tag{14}$$

When using unstructured meshes, the Cartesian version of w cannot be used. Adaptation of the immersed boundary formalism to unstructured meshes relies on the Reproducing Kernel Particle Method [51, 52, 36]. The intensity and the point of application of the force to regularize have to be ensured by the chosen discrete Dirac function w. To this respect, it proves useful to introduce the moments of the window function w at the location of the solid node m:

$$\begin{aligned}m_{a,b,c}(\overrightarrow{X_m})=\sum_{j=1}^{J}\Bigg[&\left(\frac{x_j-X_m}{h}\right)^a\left(\frac{y_j-Y_m}{h}\right)^b\left(\frac{z_j-Z_m}{h}\right)^c\\ &\times w\left(\frac{||\overrightarrow{x_j}-\overrightarrow{X_m}||}{h}\right)dV_j\Bigg].\end{aligned} \tag{15}$$

For a unit point force applied at coordinates $\overrightarrow{X_m}$, moments calculated at $\overrightarrow{X_m}$ are known: the first moment $m_{0,0,0}$ is 1, and the following ones are all 0.

When using unstructured meshes, the following isotropic extension of Eqs. (13)-(14) is naturally introduced:

$$w\left(\frac{||\overrightarrow{x_j}-\overrightarrow{X_m}||}{h}\right)=w(r)=\begin{cases}\dfrac{1}{2^{dim-1}}\dfrac{1}{4h}\left[1+\cos\left(\dfrac{\pi r}{2}\right)\right] & \text{if } |r|<2\\ \\ 0 & \text{if } |r|\geq 2\end{cases} \tag{16}$$

Since the resulting regularized force $\overrightarrow{f_j}$ does not meet the moment condition ($m_{a,b,c}=0$ except $m_{0,0,0}=1$), a modified weight function for regularization is introduced:

$$\begin{aligned}\overline{w}\left(\frac{||\overrightarrow{x_j}-\overrightarrow{X_m}||}{h}\right)&=w\left(\frac{||\overrightarrow{x_j}-\overrightarrow{X_m}||}{h}\right)\times\left[\beta_0+\beta_1\frac{x_j-X_m}{h}\right.\\ &\left.+\beta_2\frac{y_j-Y_m}{h}+\beta_3\frac{z_j-Z_m}{h}\right],\end{aligned} \tag{17}$$

where β_k are the coefficients of the polynomial correction of the original window function. The moments of the modified window function are then given by:

$$\begin{aligned}\overline{m_{a,b,c}}(\overrightarrow{X_m})=\sum_{j=1}^{J}\left[\left(\frac{x_j-X_m}{h}\right)^a\left(\frac{y_j-Y_m}{h}\right)^b\left(\frac{z_j-Z_m}{h}\right)^c\right.\\ \left.\times\overline{w}\left(\frac{||\overrightarrow{x_j}-\overrightarrow{X_m}||}{h}\right)dV_j\right].\end{aligned} \tag{18}$$

By plugging the definition of $\overline{w}$ into Eq. (18), one easily obtains that:

$$\begin{pmatrix}\overline{m_{0,0,0}}\\ \overline{m_{1,0,0}}\\ \overline{m_{0,1,0}}\\ \overline{m_{0,0,1}}\end{pmatrix}=\begin{pmatrix}m_{0,0,0} & m_{1,0,0} & m_{0,1,0} & m_{0,0,1}\\ m_{1,0,0} & m_{2,0,0} & m_{1,1,0} & m_{1,0,1}\\ m_{0,1,0} & m_{1,1,0} & m_{0,2,0} & m_{0,1,1}\\ m_{0,0,1} & m_{1,0,1} & m_{0,1,1} & m_{0,0,2}\end{pmatrix}\begin{pmatrix}\beta_0\\ \beta_1\\ \beta_2\\ \beta_3\end{pmatrix}=\mathcal{M}\begin{pmatrix}\beta_0\\ \beta_1\\ \beta_2\\ \beta_3\end{pmatrix}, \tag{19}$$

where $\mathcal{M}$ contains the moments of the isotropic window function w (Eq. (16)). The first moments of the modified window function can then be imposed to

their expected values $(1,0,0,0)$ by calculating β_k such as:

$$\begin{pmatrix}\beta_0\\ \beta_1\\ \beta_2\\ \beta_3\end{pmatrix} = \mathcal{M}^{-1}\begin{pmatrix}\overline{m_{0,0,0}}\\ \overline{m_{1,0,0}}\\ \overline{m_{0,1,0}}\\ \overline{m_{0,0,1}}\end{pmatrix} = \mathcal{M}^{-1}\begin{pmatrix}1\\ 0\\ 0\\ 0\end{pmatrix}. \tag{20}$$

In this case, the modified window function ensures that the moment of order 0 is equal to 1, and the moments of order 1 are equal to 0. To also impose the moments of order 2 to 0, the original window function has to be corrected with a quadratic correction leading to the calculation of 10 β_k coefficients [52]:

$$\begin{aligned}\overline{w}\left(\frac{||\overrightarrow{x_j}-\overrightarrow{X_m}||}{h}\right) = w\left(\frac{||\overrightarrow{x_j}-\overrightarrow{X_m}||}{h}\right)\times\Bigg[&\beta_0+\beta_1\frac{x_j-X_m}{h}\\ &+\beta_2\frac{y_j-Y_m}{h}+\beta_3\frac{z_j-Z_m}{h}+\beta_4\left(\frac{x_j-X_m}{h}\right)\left(\frac{y_j-Y_m}{h}\right)\\ &+\beta_5\left(\frac{y_j-Y_m}{h}\right)\left(\frac{z_j-Z_m}{h}\right)+\beta_6\left(\frac{z_j-Z_m}{h}\right)\left(\frac{x_j-X_m}{h}\right)\\ &+\beta_7\left(\frac{x_j-X_m}{h}\right)^2+\beta_8\left(\frac{y_j-Y_m}{h}\right)^2+\beta_9\left(\frac{z_j-Z_m}{h}\right)^2\Bigg],\end{aligned} \tag{21}$$

leading to:

$$\begin{pmatrix}\overline{m_{0,0,0}}\\ \overline{m_{1,0,0}}\\ \overline{m_{0,1,0}}\\ \overline{m_{0,0,1}}\\ \overline{m_{1,1,0}}\\ \overline{m_{0,1,1}}\\ \overline{m_{1,0,1}}\\ \overline{m_{2,0,0}}\\ \overline{m_{0,2,0}}\\ \overline{m_{0,0,2}}\end{pmatrix} = \begin{pmatrix}m_{0,0,0} & m_{1,0,0} & \cdots & m_{0,0,2}\\ m_{1,0,0} & m_{2,0,0} & \cdots & m_{1,0,2}\\ \vdots & \vdots & \ddots & \vdots\\ m_{0,0,2} & m_{1,0,2} & \cdots & m_{0,0,4}\end{pmatrix}\begin{pmatrix}\beta_0\\ \beta_1\\ \beta_2\\ \beta_3\\ \beta_4\\ \beta_5\\ \beta_6\\ \beta_7\\ \beta_8\\ \beta_9\end{pmatrix} = \mathcal{M}\begin{pmatrix}\beta_0\\ \beta_1\\ \beta_2\\ \beta_3\\ \beta_4\\ \beta_5\\ \beta_6\\ \beta_7\\ \beta_8\\ \beta_9\end{pmatrix}. \tag{22}$$

The coefficients β_k are thus given by:

$$\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \\ \beta_9 \end{pmatrix} = \mathcal{M}^{-1} \begin{pmatrix} \overline{m_{0,0,0}} \\ \overline{m_{1,0,0}} \\ \overline{m_{0,1,0}} \\ \overline{m_{0,0,1}} \\ \overline{m_{1,1,0}} \\ \overline{m_{0,1,1}} \\ \overline{m_{1,0,1}} \\ \overline{m_{2,0,0}} \\ \overline{m_{0,2,0}} \\ \overline{m_{0,0,2}} \end{pmatrix} = \mathcal{M}^{-1} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{23}$$

At the end, computing the regularized force as:

$$\overrightarrow{f_j} = \sum_{m=1}^{M} \left[\overrightarrow{F_m} \times \overline{w} \left(\frac{||\overrightarrow{x_j} - \overrightarrow{X_m}||}{h} \right) \right], \tag{24}$$

allows a proper representation of the membrane force location, and moments up to second order. Although possible in principle, the proper representation of higher order moments is not considered in this study.

2.3. Navier-Stokes equations resolution

A massively parallel unstructured finite-volume flow solver is used to solve the forced Navier-Stokes equations over the Eulerian mesh using a projection method [53]. The momentum conservation equations reads:

$$\rho \left(\frac{\partial \overrightarrow{v_j}}{\partial t} + \overrightarrow{v_j}.\nabla \overrightarrow{v_j} \right) = -\nabla p_j + \nabla. \left[\mu \left(\nabla \overrightarrow{v_j} + (\nabla \overrightarrow{v_j})^T \right) \right] + \overrightarrow{f_j}, \tag{25}$$

where $\overrightarrow{v_j}$ and p_j are the velocity vector and pressure on fluid node j, ρ the density and μ the dynamic viscosity. For an incompressible flow, the mass conservation constraint becomes:

$$\nabla.\overrightarrow{v_j} = 0. \tag{26}$$

The fluid velocity is first advanced using a 4th-order central scheme in space and a 4th-order Runge-Kutta scheme in time. A divergence-free velocity field is

obtained at the end of the time-step by solving a Poisson equation for pressure and correcting the predicted velocity. A Deflated Preconditioned Conjugate Gradient algorithm is used to solve this Poisson equation. More details about the employed numerical methods can be found in [49, 54].

2.4. Membrane convection

Adherence of the fluid over the membrane makes the fluid velocity continuous at the membrane location and equal to the membrane velocity. Thus, the velocity $\overrightarrow{V_m}$ of the solid node m is calculated by interpolating the fluid velocity at the solid nodes location from the fluid velocity at the neighboring fluid nodes:

$$\overrightarrow{V_m} = \sum_{j=1}^{J} \left[\overrightarrow{v_j} \times \overline{w} \left(\frac{||\overrightarrow{x_j} - \overrightarrow{X_m}||}{h} \right) dV_j \right]. \tag{27}$$

This interpolation process again involves the use of the discrete Dirac function $\overline{w}$, which is defined in the same manner as in the forces regularization process described in section 2.2.

2.5. Discretization

When using immersed boundary methods, authors generally recommend the use of similar solid and fluid mesh discretizations. Numerical experiences showed that the present method performes well when the ratio h/H between the fluid mesh resolution and the solid effective resolution is such that $0.5 < h/H < 2$. An initial ratio $h/H \approx 1$ is typically suited to start with.

2.6. Volume correction

The original immersed boundary method tends to suffer from a small leakage problem. Indeed, whatever the choice made for the window function, interpolation does not conserve the divergence-free character of the carrying fluid flow [27]. This issue is sometimes pointed out as a major drawback of the IBM [28].

When dealing with closed membranes, this leakage problem results in a non-conservation of the volume enclosed by the flexible membrane. In this case, the volume can be corrected by using a Lagrange Multiplier method to calculate

the smallest correction of the solid nodes coordinates to conserve the particle volume. This procedure is detailed in [36] in the case of 2D particles, and generalization for 3D particles is provided in appendix. Such a correction is however not possible when dealing with open membranes. In the present ITBM, this procedure is performed by constructing a triangulation of the middle surface of the membrane, which however limits its use to linear prismatic elements, but will be later adapted for quadratic hexaedral elements.

3. Verification and validation

This section is dedicated to the verification and the validation of the present method. Table 1 summarizes the selection of the test cases considered in this respect.

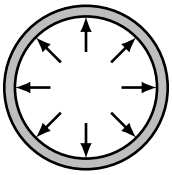
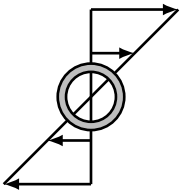
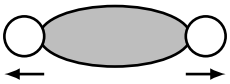
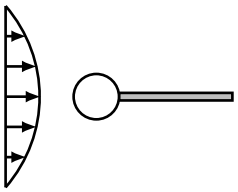
Test case (section)	**Diagram**	**Reference data**
Non-linear bending of an elastic plate (section 3.1)		Numerical results from Sze *et al.* (2004) [55]
Inflation of a spherical capsule (section 3.2)		Analytical solution
Capsule in a linear shear flow (section 3.3)		Numerical results from Lac *et al.* (2004) [56]
Red blood cell stretched by optical tweezers (section 3.4)		Experimental data from Mills *et al.* (2004) [57]
Flow-induced vibration of an elastic beam behind a cylinder (section 3.5)		Numerical results from Turek and Hron (2006) [58] and Turek *et al.* (2010) [59]

Table 1: Summary of the validation test cases presented in section 3.

Test cases of sections 3.1 and 3.2 should be considered as verification test cases, whereas the other test cases are more challenging and are referred to as validation test cases.

3.1. Non-linear bending of an elastic plate

The principle of this test case is to look at the mechanical equilibrium of a plate. As shown in Fig. 4, the plate is defined by its length L, width l and thickness e. One of its extremities is fixed, while the other is loaded by applying a force F, normal to the initial position of the plate. The objective is to verify that the mechanical equilibrium obtained by the FSI computation is the same as the one from a simple structural computation, performed with the LMGC90 solid mechanics solver. Although it appears trivial, this test case is actually a good mean to verify the operation of the coupling algorithm. Indeed, FSI and structural resolutions are completely different insofar as the displacement of the structure $\overrightarrow{U_m}$ is solved during the structural computation, whereas in the FSI computation the displacement is imposed by the velocity of the fluid, the only contribution of the solid mechanics solver being to compute the mechanical force $\overrightarrow{F_m}$ resulting from this imposed displacement. In this context, the errors generated by the procedures of forces regularization (section 2.2) and membrane convection (section 2.4) could potentially impact the mechanical equilibrium of the plate.

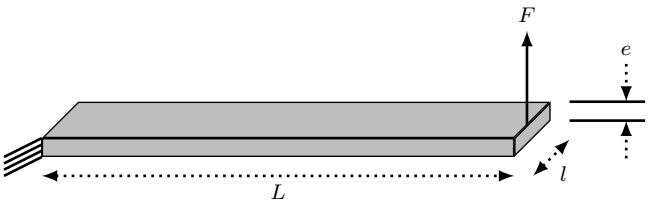


Figure 4: Geometrical parameters of the plate. The force F is applied over all the surface of the plate extremity.

3.1.1. Structural computation

The static structural problem is first solved using the LMGC90 solid mechanics solver, giving the mechanical equilibrium of the plate. The plate is modeled by a Neo-Hookean material (Eq. (9)). The parameters used for this computation are the same of those used in the benchmark test proposed by Sze *et al.* (2004) [55], and are given in Tab. 2.

F	L	l	e	E	ν
4 N	10 m	1 m	0.1 m	1.2 10^6 Pa	0

Table 2: Parameters of the benchmark test case proposed by Sze *et al.* (2004) [55].

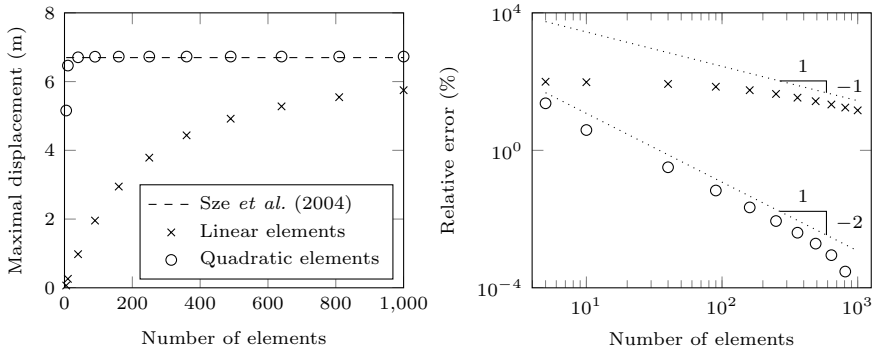


Figure 5: **Left:** Evolution of the maximal displacement as a function of the number of elements. Results obtained with both linear and quadratic hexaedral finite elements are compared with the result of Sze *et al.* (2004) [55]. **Right:** Error relative to most refined computation.

In order to study the influence of the finite element interpolation, both linear hexaedral finite elements and quadratic hexaedral finite elements are compared with the reference result of Sze *et al.* (2004) [55]. Figure 5 (left) shows the maximal displacement (in the direction of the applied force) obtained at the extremity of the plate, for different mesh resolutions. It is seen that the chosen interpolation has a strong influence on the final result. Indeed, the numerical solution converge much faster when the quadratic interpolation is chosen. When using quadratic elements, the solution reaches a constant relative difference of 0.45% with respect to the reference result. For the linear interpolation, the solution is not yet converged when reaching the maximal number of elements, and a final relative difference of 14.15% is obtained. It appears that, with linear elements, flexural deformation cannot be properly simulated without using very fine mesh resolutions. In contrast, quadratic elements offer a better accuracy even with a coarse mesh. Regarding the relative convergence error (see Fig. 5,

right) calculated from the most refined computation, it is seen that the order of convergence for both the linear and the quadratic interpolation is retrieved.

3.1.2. Fluid-structure interaction computation

The next step is to perform a FSI computation, and compare the result with the one obtained in the structural computation. A time dependent FSI problem is solved, by immersing the plate in a fluid computational domain (Fig. 6, left). The fluid box is extended from 0 to $1.15L$ in length, from -$2l$ to $2l$ in width, and from -$15e$ to $85e$ in height. The fluid mesh is an unstructured tetrahedral mesh of 23 174 elements, with a constant resolution of $h = 0.5$ m. The plate is meshed using 10 quadratic hexaedral elements (second point in Fig. 5), with an effective resolution of $H = 0.5$ m. Note that in this case, the ratio between the fluid mesh resolution and the thickness of the membrane is $h/e = 5$.

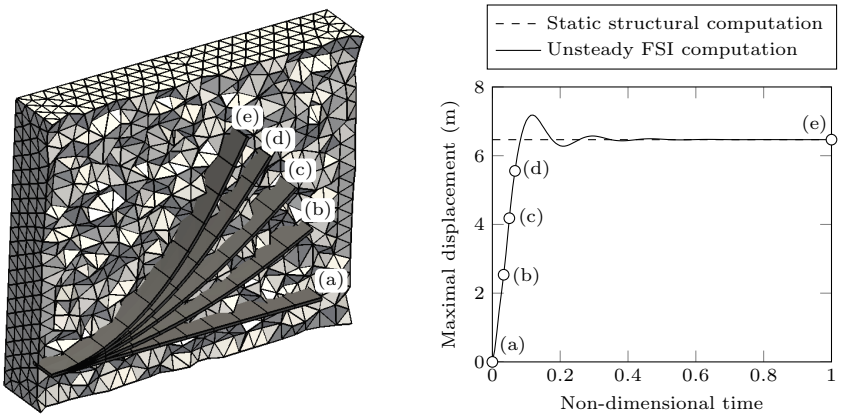


Figure 6: **Left:** Visualisation of the plate deformation within the fluid box. **Right:** Evolution of the maximal displacement of the plate extremity. The FSI computation is compared with the structural computation.

At each iteration, the external force F is added to the calculated mechanical force $\overrightarrow{F_m}$. This external force is distributed over the solid nodes located at the extremity of the plate. As the mechanical force, the applied external force is regularized on the fluid grid (see section 2.2), resulting in a motion of the fluid.

The displacement of the plate is then calculated during the step of membrane convection (see section 2.4). It is thus the motion of the fluid which enables to the plate to deform and reach the equilibrium position.

Figure 6 shows the deformation of the plate (left) and the time evolution of the maximal displacement of the plate extremity from the FSI computation (right). Applying the loading first puts the plate in motion until a steady state is finally reached. Once the fluid is at rest, the plate reaches its mechanical equilibrium. This equilibrium is compared with the equilibrium given by the structural computation. The comparison has been done for both first order and second order corrections for the computation of the window function (see section 2.2), leading to the same result: for the given rounding precision (three decimal places), the maximal displacement given by the FSI computation is strictly the same of the one from the structural computation. When increasing the ratio h/e up to 1000, FSI and structural computations still provide identical mechanical equilibriums (not shown), showing that the present method is able to simulate membranes having a thickness much smaller than the fluid mesh resolution.

3.2. Inflation of a spherical capsule

In this section, the inflation of a spherical capsule is considered. The capsule is defined by its internal and external radii, respectively R_i and R_e (Fig. 7, left). An elastic linear isotropic material is used, defined by its Young modulus E and Poisson coefficient ν. When submitted to an internal pressure P, the capsule deforms (Fig. 7, right).

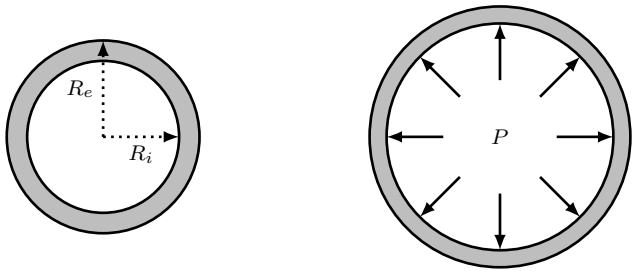


Figure 7: A spherical capsule of internal radius R_i and external radius R_e is inflated applying an internal pressure P.

Within the assumption of small perturbations, it is possible to derive an analytical solution of the radial displacement of the capsule submitted to the internal pressure P:

$$U_r\left(r\right) = \frac{{R_i}^3}{{R_e}^3 - {R_i}^3}\left[\left(1-2\nu\right)r + \left(1+\nu\right)\frac{{R_e}^3}{2r^2}\right]\frac{P}{E}, \tag{28}$$

with $R_i < r < R_e$.

3.2.1. Structural computation

As in the previous test case, the structural problem is first solved using the LMGC90 solid mechanics solver. The values of the physical parameters are given in Tab. 3, where e and R denote the thickness and mean radius of the capsule, respectively.

P	e	R	R_i	R_e	E	ν
500 Pa	0.05 m	0.5 m	$R - e/2$	$R + e/2$	1.5 10^6 Pa	0.4

Table 3: Parameters chosen for the structural computation.

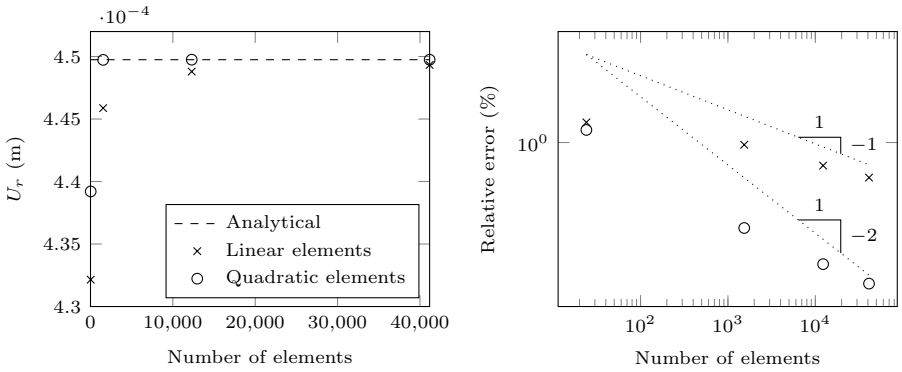


Figure 8: **Left:** Evolution of the radial displacement as a function of the number of elements. Results obtained with both linear and quadratic hexaedral finite elements are compared with the analytical solution. **Right:** Relative error for both linear and quadratic interpolations.

Here again, the influence of the finite-element interpolation is investigated. Both linear hexaedral finite elements and quadratic hexaedral finite elements are

compared with the analytical solution. Figure 8 shows the radial displacement $U_r(r)$ (for $r = R_e$) and the relative error, for different mesh resolutions. The quadratic interpolation still shows a faster convergence, with a higher order of convergence regarding the relative error. But errors obtained for both linear and quadratic elements are very small, even for the coarsest mesh. Indeed, the relative error range from 3.91% to 9.38 10^{-2}% for the linear interpolation, and from 2.34% to 7.33 10^{-5}% for the quadratic interpolation. It has been seen in section 3.1.1 that, when the mesh is not well refined, flexural deformation cannot be properly simulated using linear elements. In this case, it is demonstrated that when the deformation is purely extensional, the linear interpolation is sufficient and provides reasonable errors.

3.2.2. Fluid-structure interaction computation

An equivalent FSI problem can be solved by immersing the spherical capsule in a fluid computational domain (Fig. 9, left). The fluid box is extended from -4R to 4R in all the directions. The fluid mesh is composed of 2 101 165 tetrahedral elements, with a mesh resolution around the capsule of $h = 0.025$ m. The capsule is meshed using 1536 quadratic hexahedral elements, with an effective resolution of $H = 0.025$ m.

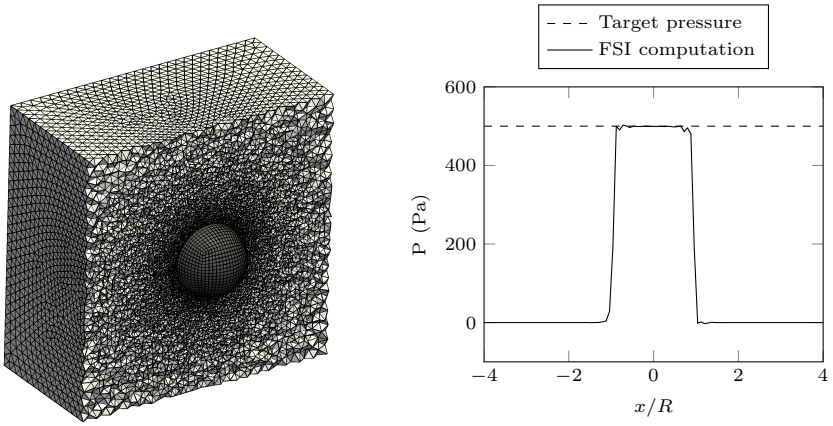


Figure 9: **Left:** Setup of the FSI computation, the capsule is immersed in a fluid box. **Right:** Pressure profile along the line $(-4R, 4R)$.

The capsule is thus surrounded by an outer fluid, and contains an inner fluid. Rather than applying a pressure on the capsule as for the structural problem, the capsule is inflated by imposing the analytical displacement on each solid node. This results in a pressurization of the inner fluid, and the inner pressure can then be compared to the target pressure applied in the structural computation ($P = 500$ Pa). Figure 9 (right) shows a pressure profile along a line which crosses the fluid computational domain. The pressure drop between the inner and the outer fluid is seen to be very close to the target pressure. Regarding the pressure at the center of the capsule, the relative error is 0.11%. The regularization process detailed in section 2.2 generates some oscillations near to the membrane location delimitating the inner and the outer fluids, but does not prevent to accuratly capture the pressure drop.

3.3. Capsule in a linear shear flow

In this test case, an initially spherical capsule of mean radius a is deposited in a linear shear flow, defined as $\vec{u} = ky\,\vec{e_x}$ (see Fig. 10, left). When deposited in the linear shear flow, the capsule begins to deform by changing orientation, and a steady deformation is finally reached (see Fig. 10, right).

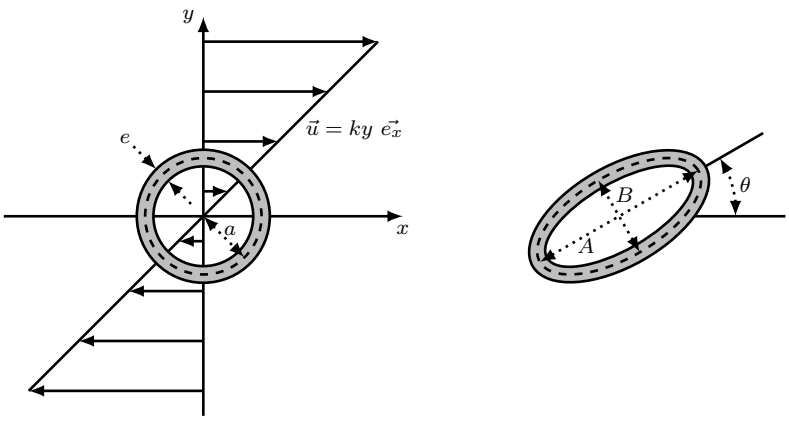


Figure 10: **Left:** The capsule is initially spherical, and deposited in a linear shear flow. **Right:** A steady deformation is reached, during which a tank-treading motion of the membrane is observed.

The orientation θ of the capsule, and the lengths A and B of the two principal axes of the ellipsoid of inertia in the shear plane can then be calculated. The deformation of the capsule in the shear plane is measured by the Taylor parameter:

$$D = \frac{A - B}{A + B}. \tag{29}$$

All the parameters relevant to the test case can be found in Tab. 4. The capsule-based Reynolds number R_e is chosen sufficiently low to satisfy the Stokes flow assumption, and the capillary number C_a which compares the viscous force to the membrane elastic force is successively set to 0.15; 0.3; 0.6. The membrane of the capsule is assumed to follow the Yeoh law (Eq. (11)), with $C_1 = G/2$ and $C_2 = C_3 = 0$. A Poisson coefficient $\nu = 0.4995$ is chosen to insure that the membrane is quasi-incompressible. The ratio a/e between the initial mean radius of the capsule and the thickness of the membrane is chosen to be significantly high. A ratio $a/e = 1000$ is used, ensuring that the membrane is very thin compared to the size of the capsule, and making the bending resistance of the membrane almost negligible. All these conditions were selected to allow a proper comparison with the numerical results reported in Lac *et al.* (2004) [56].

Capsule parameters	
Initial mean radius	a
Thickness	e
Shear modulus	G
Poisson coefficient	ν
Flow parameters	
Dynamic viscosity	μ
Density	ρ
Shear rate	k

Non-dimensional parameters	
Capillary number	$C_a = \dfrac{\mu k a}{eG}$
Reynolds number	$R_e = \dfrac{\rho k a^2}{\mu}$

Table 4: Definition of the capsule, flow, and non-dimensional parameters.

The fluid domain is extended from -16a to 16a around the spherical capsule. The fluid mesh is composed of 1 924 682 tetrahedral elements with a mesh resolution at the center of the fluid box of $h = a/10$. In order to use the volume correction procedure (see section 2.6), the capsule is meshed using 2 906 linear prismatic elements, with an effective resolution of $H = a/10$. Figure 11 shows the evolution of the capsule deformation over the computation, for $C_a = 0.6$. The field displayed on the figure corresponds to the x-coordinate of the initial (reference) position of the membrane, introduced in section 2. From (a) to (e), the capsule deforms and reach constant deformation and orientation. From (f) to (j), the well known tank-treading behaviour of the membrane turning around the inner fluid is observed.

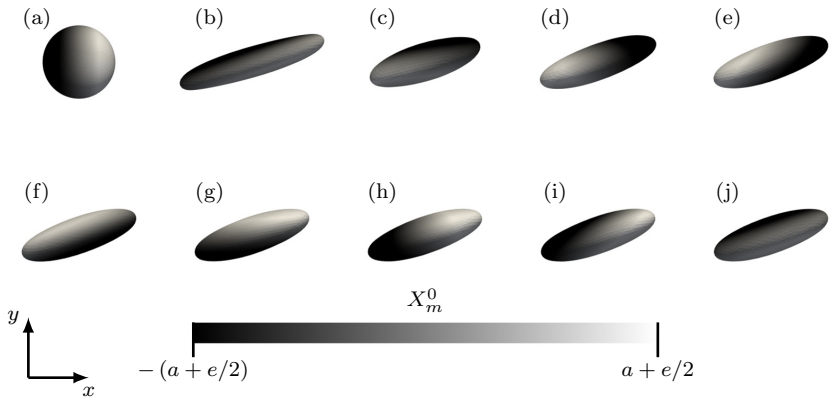


Figure 11: Visualisation of the capsule deformation over the computation. The x-coordinate of the initial position of the membrane nodes is displayed, enabling to visualize the tank-treading phenomenon.

The Taylor parameter D and the capsule inclinaison θ/π can be calculated from the steady deformation of the capsule. Figure 12 shows the evolution of both parameters as a function of the capillary number C_a. The present results are compared with the results of Lac *et al.* (2004) [56] where the membrane is considered to be infinitely thin and incompressible. A satisfactory comparison is obtained for both parameters. The relative error for the Taylor parameter D

range from 0.18% to 1.03%, and from 0.64% to 6.08% for the capsule inclinaison θ/π. The maximal relative error is obtained when $C_a = 0.6$, and still increases when increasing the capillary number (not shown). This is probably due to the incompressibility of the membrane, which is not strictly insured in the present computations since using a quasi-incompressible model. Indeed, volume variations are still allowed, and are seen to increase with the capillary number (not shown).

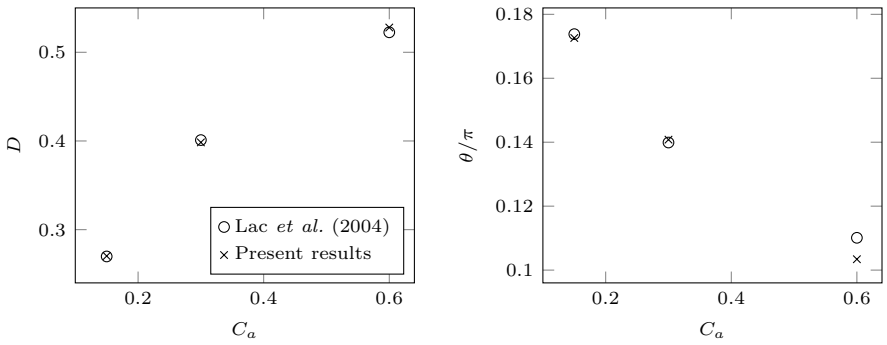


Figure 12: **Left:** Evolution of the Taylor parameter D as a function of the capillary number C_a. **Right:** Evolution of the capsule inclinaison θ/π. The present results are compared with the results of Lac *et al.* (2004) [56].

Influence of the volume correction procedure is shown in Tab. 5, considering the highest capillary number $C_a = 0.6$. A volume variation of 2.73% is observed when no volume correction is performed, whereas there is almost 0% volume variation when correcting the inner volume of the capsule. The influence of this volume correction procedure on the output parameters D and θ/π is however very small.

	Volume variation (%)	$\boldsymbol{D}$	$\boldsymbol{\theta/\pi}$
Volume correction	$\approx 10^{-4}$	0.5279	0.1034
No volume correction	2.73	0.5350	0.1019

Table 5: Influence of the volume correction on the outputs of the simulation for $C_a = 0.6$.

3.4. Red blood cell stretched by optical tweezers

The purpose is now to simulate the optical tweezers experiment, which consists in stretching a red blood cell (RBC) by using the optical trap principle [57]. An illustration of the experiment is given in Fig. 13.

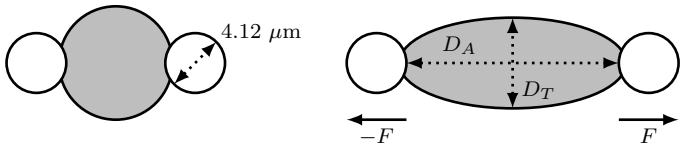


Figure 13: Illustration of the optical tweezers experiment. The effect of the beads is simulated by applying a stretching force to two opposite regions over the cell membrane, following the procedure of Mills *et al.* (2004) [57].

Two silica microbeads, each 4.12 μm in diameter, are attached to the cell at diametrically opposite points. The left bead is anchored to the surface of a glass slide while the right bead is trapped by a laser beam. The trapped bead remaining at rest, moving the slide and attached left bead stretches the cell. The experiment is simulated by applying a stretching force F to two opposite regions over the membrane in order to mimic the beads effect. For each imposed force F, the axial diameter D_A (in the direction of the stretching), and the transverse diameter D_T (orthogonal to the stretching direction) of the cell are measured (see Fig. 13, right) after the equilibrium has been reached (zero velocity field over the whole computational domain).

The average shape of a RBC has been determined by Evans [60], and is given by:

$$z = \pm 0.5 R_0 \left[1 - \frac{x^2+y^2}{R_0^2}\right] \left[A + B\frac{x^2+y^2}{R_0^2} + C\left(\frac{x^2+y^2}{R_0^2}\right)^2\right] \qquad (30)$$

where $R_0 = 3.91$ μm is the average RBC radius, $A = 0.207161$, $B = 2.002558$, and $C = -1.122762$. The thickness of the RBC membrane is known to be few nanometers [61], and is here chosen to be $e = 4.0$ nm. As in the work of Mills *et al.* (2004) [57], the complex membrane of the RBC is modeled as a three-dimensional continuum, following the Yeoh law (Eq. (11)) with $C_1 = G/2$,

$C_2 = 0$ and $C_3 = G/30$. The product between the membrane thickness e and the shear modulus G, also known as the in-plane shear modulus, is chosen to be $eG = 7.3\mu$N/m. As in the previous test case, a Poisson coefficient $\nu = 0.4995$ is chosen to insure that the membrane is quasi-incompressible. The RBC is immersed in a fluid box extended from -$3R_0$ to $3R_0$ in the direction of the stretching, from -$2R_0$ to $2R_0$ in the direction orthogonal to the stretching, and from $-R_0$ to R_0 in the height of the cell. The fluid mesh is composed of 628 660 tetrahedral elements, with a constant mesh resolution of $h = R_0/12.5$. In order to use the volume correction procedure (see section 2.6), the RBC is meshed using 3 360 linear prismatic elements, with an effective resolution of $H = R_0/12.5$. The same procedure as the one used for the test case of section 3.1 is here used. As an external force, the force F is applied on the RBC membrane. The fluid thus starts moving, and the cell deforms. After a transient phase, the mechanical forces inside the membrane and the applied external force balance, and a steady deformation is obtained. Figure 14 shows the deformation of the RBC for different values of the applied force F, which ranges from 0 to 193 pN. Each stretching force thus corresponds to one computation. Starting from the well known biconcave shape of the RBC, the cell is more and more elongated when increasing the force, and a large fold appears. Note that this kind of shape is also observed in the work of Mills *et al.* (2004) [57].

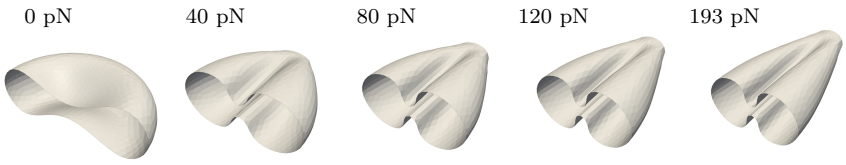


Figure 14: Visualisation of the red blood cell deformation over the entire range of stretching force. Only half of the cell is displayed.

Figure 15 shows the mechanical response of a RBC stretched by optical tweezers, as a function of the applied force. As the cell is more and more elongated when increasing the stretching force, it is seen that the axial diameter D_A increases. The elongation of the cell leads to its contraction in the orthogonal

direction, resulting in a decreasing of the transverse diameter D_T. The present computation accurately captures experimental trends over the entire range of stretching force, and provides a good prediction of the diameters.

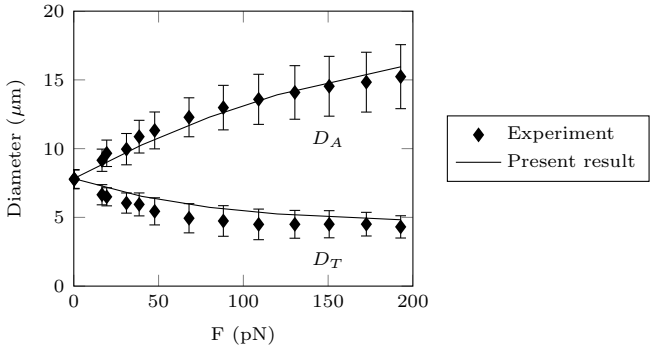


Figure 15: Axial (D_A) and transverse (D_T) diameters of the RBC stretched by optical tweezers. Comparison with the experimental data from Mills *et al.* (2004) [57].

As in section 3.3, the volume correction has only a small impact on the outcome of the simulation. For the maximum imposed force of 193 pN, there is no influence of this correction on the measured axial diameter D_A, whereas the transverse diameter D_T varies from 4.82 μm with correction (with $\approx 10^{-4}$ % volume variation) to 4.79 μm without correction (with 1.59% volume variation).

3.5. Flow-induced vibration of an elastic beam behind a cylinder

The last validation test case presented is the flow-induced vibration of an elastic beam behind a cylinder. This situation is more challenging than the former ones, since a periodic motion of the solid beam where inertia effects must be properly predicted is sought for. This test case was also selected because of the well-documented results obtained by numerous research groups which used different numerical methods for the FSI numerical resolution. These methods are summarized and briefly described in Tab. 6. For more information, the reader is referred to the paper of Turek *et al.* (2010) [59]. The configuration of the problem, as introduced by Turek and Hron (2006) [58], is detailed in Fig. 16.

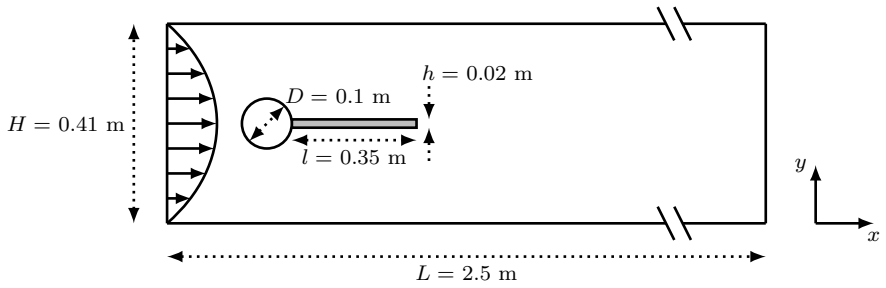


Figure 16: Details of the beam-cylinder configuration.

The cylinder center is positioned at (0.2 m, 0.2 m) (measured from the left bottom corner of the channel). The right bottom corner of the elastic beam is positioned at (0.6 m, 0.19 m). Note that the setting is non-symmetric (the centerline of the beam is at $y = 0.2$ m while the centerline of the channel is $y = 0.205$ m), preventing the extreme sensitivity of the results on the details of the flow computation. The problem is initially in 2D but is here extended to 3D by slightly extruding the 2D domain in the third direction, and imposing periodic boundary conditions on the upper and lower walls for both the fluid and the solid. A parabolic velocity profile is prescribed at the inlet of the channel:

$$v\left(0, y, z\right) = 1.5U \frac{y\left(H - y\right)}{\left(H/2\right)^2}, \tag{31}$$

where U is the bulk velocity. Two configurations were simulated, as referred to in [58, 59]: FSI1 and FSI3. The corresponding sets of parameters are gathered in Tab. 7. The elastic beam is modeled by a Saint-Venant law (Eq. (8)). The displacement of the beam extremity, initially positioned at (0.6 m, 0.2 m), is tracked during the computation for comparison with previous results. For each configuration, three different meshes were tested as summarized in Tab. 8. For each of the meshes, the fluid mesh is tetrahedric, and the mesh of the beam is made of quadratic hexaedral elements. Concerning the fluid, the mesh resolution is not constant over the whole fluid domain, the resolutions provided in Tab. 8 correspond to the mesh size around the elastic beam.

Method	Description
1 (Schäfer)	Implicit partitioned approach [62, 63] combining the finite-volume multigrid flow solver FASTEST and the finite-element structural solver FEAP, using an ALE formulation.
2a (Rannacher)	Monolithic variational formulation based on a unified Eulerian framework ("interface capturing").
2b (Rannacher)	Monolithic variational formulation based on the standard ALE approach ("interface fitting").
3 (Turek/Hron)	Fully implicit monolithic ALE-FEM approach.
4 (Breuer)	Partitioned approach combining the finite-volume scheme FASTEST-3D for the flow and the finite-element structural solver Carat, using an ALE formulation.
5 (Krafczyk/Rank)	Explicit partitioned approach which combines the Lattice-Boltzmann flow solver VirtualFluids (VF) and the structural p-FEM solver AdhoC.
6 (Wall)	A strongly coupled iterative staggered scheme [64, 65] based on an ALE formulation, Q2Q2 elements for the fluid and an EAS formulation for the structure.
7 (Bletzinger)	Partitioned approach based on a combination of three independent softwares: in-house codes CARAT++ and CoMA for structural shell analysis, coupling control and data transfer between non-matching grids, and the open source finite volume flow solver OpenFOAM.

Table 6: Summary of the different methods used in [59] to simulate the present test case.

	FSI1	**FSI3**
Flow parameters		
Density	$\rho = 1000$ kg/m^3	$\rho = 1000$ kg/m^3
Dynamic viscosity	$\mu = 1$ Pa.s	$\mu = 1$ Pa.s
Inlet velocity	$U = 0.2$ m/s	$U = 2$ m/s
Reynolds number	$\rho UD/\mu = 20$	$\rho UD/\mu = 200$
Beam parameters		
Density	$\rho_s = 1000$ kg/m^3	$\rho_s = 1000$ kg/m^3
Shear modulus	$G = 0.5$ MPa	$G = 2$ MPa
Poisson coefficient	$\nu = 0.4$	$\nu = 0.4$

Table 7: Parameters of the test case, two configurations are simulated.

	$\boldsymbol{M_1}$	$\boldsymbol{M_2}$	$\boldsymbol{M_3}$
Fluid			
Number of elements	160 707	971 069	6 582 336
Mesh resolution h	1.0 10^{-2} m	5.0 10^{-3} m	2.5 10^{-3} m
Solid			
Number of elements	72	560	4 480
Effective resolution H	1.0 10^{-2} m	5.0 10^{-3} m	2.5 10^{-3} m

Table 8: Parameters of the meshes.

The FSI1 configuration is first simulated, leading to a stationary displacement of the elastic beam, consistent with [59]. Table 9 shows the two components of the displacement of the tracked point located at the extremity of the beam. The results obtained in [59] for different numerical methods are first displayed in the table. The present results are also displayed for the different meshes which have been tested in the study. From the 3 levels of mesh M_1, M_2 and M_3 used to simulate the present test case, it is possible to calculate the apparent spatial order p of the method. Let dx_1, dx_2, dx_3 be the spatial resolutions of meshes M_1, M_2 and M_3, respectively (with $dx = h = H$ provided

	u_x $[\times 10^{-5}]$ (m)	u_y $[\times 10^{-4}]$ (m)
2a	**2.4800**	**7.7800**
2b	2.2695	8.1556
3	2.2705	8.2088
5	**2.2160**	8.2010
6	2.2680	8.2310
7	2.2640	**8.2800**
M_1	2.8547	8.9378
M_2	2.6724	7.7917
M_3	2.5841	8.2243
M_∞	2.5012	8.4866
p	1.05	1.41
e_∞ (%)	3.32	3.09

Table 9: Results for the FSI1 configuration.

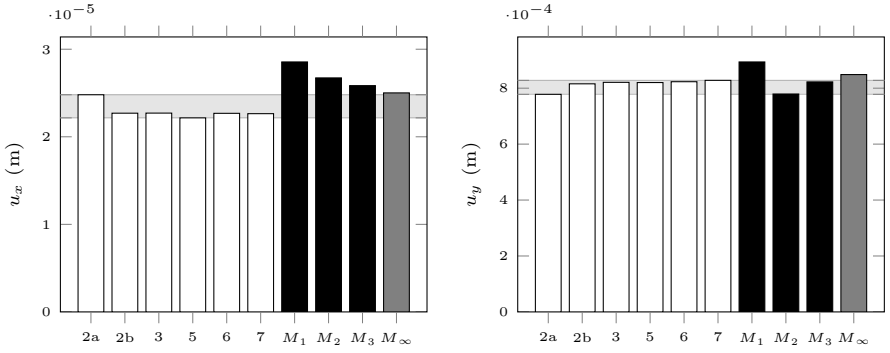


Figure 17: u_x and u_y components of the displacement of the beam extremity. The results obtained with the methods reported in [59] (displayed in white, and described in Tab. 6) can be compared with the present results, obtained for 3 different mesh resolutions (displayed in black). The extrapolated results are also provided (displayed in gray).

in Tab. 8), and $r = dx_1/dx_2 = dx_2/dx_3$ be the refinement factor, the apparent spatial order p is given by [66]:

$$p = \frac{1}{\ln(r)} \left| \ln \left| \frac{\phi_1 - \phi_2}{\phi_2 - \phi_3} \right| \right|, \tag{32}$$

where ϕ_k denotes the output variable of the simulation for the k^{th} mesh. The extrapolated value ϕ_∞ can then be calculated as follows:

$$\phi_\infty = \frac{r^p \phi_3 - \phi_2}{r^p - 1} = \frac{r^p \phi_2 - \phi_1}{r^p - 1}. \tag{33}$$

The extrapolated relative error reads:

$$e_\infty = \left| \frac{\phi_\infty - \phi_3}{\phi_\infty} \right|. \tag{34}$$

The extrapolated values of the 2 components of the displacement of the beam extremity (corresponding to the infinitely refined mesh M_∞) as well as the corresponding apparent spatial order p and extrapolated relative error are also summarized in Tab. 9. Orders obtained from the output variables are greater than 1, which is satisfactory for an immersed boundary method. One can also note that extrapolated relative errors are small.

For a better comparison, the whole results summarized in Tab. 9 are displayed in Fig. 17. A meaningful comparison between the reference results summarized in [59] and the present results is difficult to carry out for several reasons:

- In [59], the test case is 2D, and has been extended to 3D for the validation of the present method.
- Equivalent values of h and H for the simulations performed in [59] are not provided. Only the number of resolved unknowns is provided, and ranges from 11 250 (for method 2a) to 19 320 832 (for method 3), which is quite a wide range.
- It is difficult to determine if the results of the simulations performed in [59] are well converged for all the methods summarized, given the variability of the number of resolved unknowns.

At least, one can say that the present method is able to retrieve a stationary displacement of the beam extremity which is in good agreement with values reported in [59]. Indeed, regarding extrapolated results displayed in Fig. 17, it is seen that the present results converge to the range of results reported in [59].

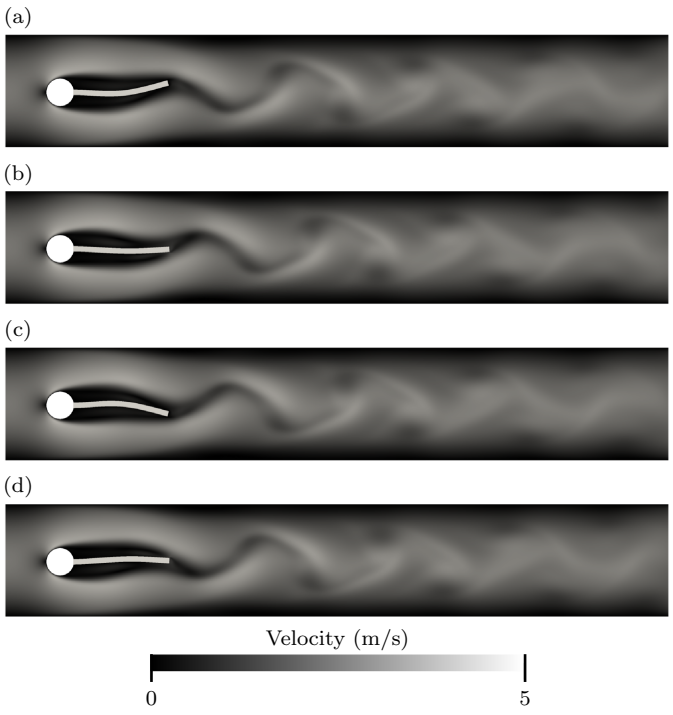


Figure 18: Visualisation of the flow-induced vibration of the elastic beam over one period of oscillation, for the FSI3 configuration. The velocity field around the beam is also displayed.

The set of parameters used for the FSI3 configuration leads to a periodic displacement of the beam extremity, as shown in Fig. 18. In this case, the mean displacement is provided in Tab. 10, with the amplitude of the oscillation. Two frequencies can also be calculated from the oscillatory signal of u_x and u_y, respectively f_x and f_y. Again, spatial orders larger than unity are obtained for all the output variables. Extrapolated relative errors show that the present results are well converged with respect to the frequencies, almost converged with respect to the u_y displacement, but hardly converged with respect to the u_x

displacement, which needs refinement to reach convergence. This may reflect the difficulty of the method to capture displacements tangent to the beam, coming from shear sollicitations.

	u_x $[\times 10^{-3}]$ (m)	u_y $[\times 10^{-3}]$ (m)	f_x (s^{-1})	f_y (s^{-1})
1	-2.91 ± 2.77	1.47 ± 35.26	**11.63**	**4.98**
2a	-2.48 ± 2.24	**1.27** ± 36.50	**10.10**	5.10
2b	-2.84 ± 2.67	1.28 ± 34.61	10.84	5.42
3	-2.88 ± 2.72	1.47 ± 34.99	10.93	5.46
4	**-4.54** ± **4.34**	1.50 ± **42.50**	10.12	5.05
5	-2.88 ± 2.71	1.48 ± 35.10	11.00	5.50
6	**-2.00** ± **1.89**	1.45 ± **29.00**	10.60	5.30
7	-3.04 ± 2.87	**1.55** ± 36.63	10.99	**5.51**
M_1	-0.55 ± 0.49	2.11 ± 13.89	10.83	5.41
M_2	-2.06 ± 1.94	1.22 ± 29.21	10.92	5.46
M_3	-2.57 ± 2.41	1.39 ± 32.75	10.95	5.47
M_∞	-2.83 ± 2.63	1.43 ± 33.81	10.97	5.49
p	1.57 1.62	2.39 2.11	1.58	1.34
e_∞ (%)	9.18 8.59	2.78 3.15	0.14	0.20

Table 10: Results for the FSI3 configuration.

The results of Tab. 10 are also displayed in Fig. 19, for a better comparison. The variability of the results reported in [59] shows the extreme sensitivity of the present configuration. This time, the number of resolved unknowns reported in [59] ranges from 11 250 (for method 2a) to 2 480 814 (for method 5). The 3 meshes provide similar frequencies of oscillation, which are in good agreement with the results reported in [59]. However, regarding mean values and amplitudes of the displacement, it is seen that only meshes M_2 and M_3 provide a satisfactory comparison with the results reported in [59]. One can note that the extrapolated results are very close to the results obtained with method 5, which

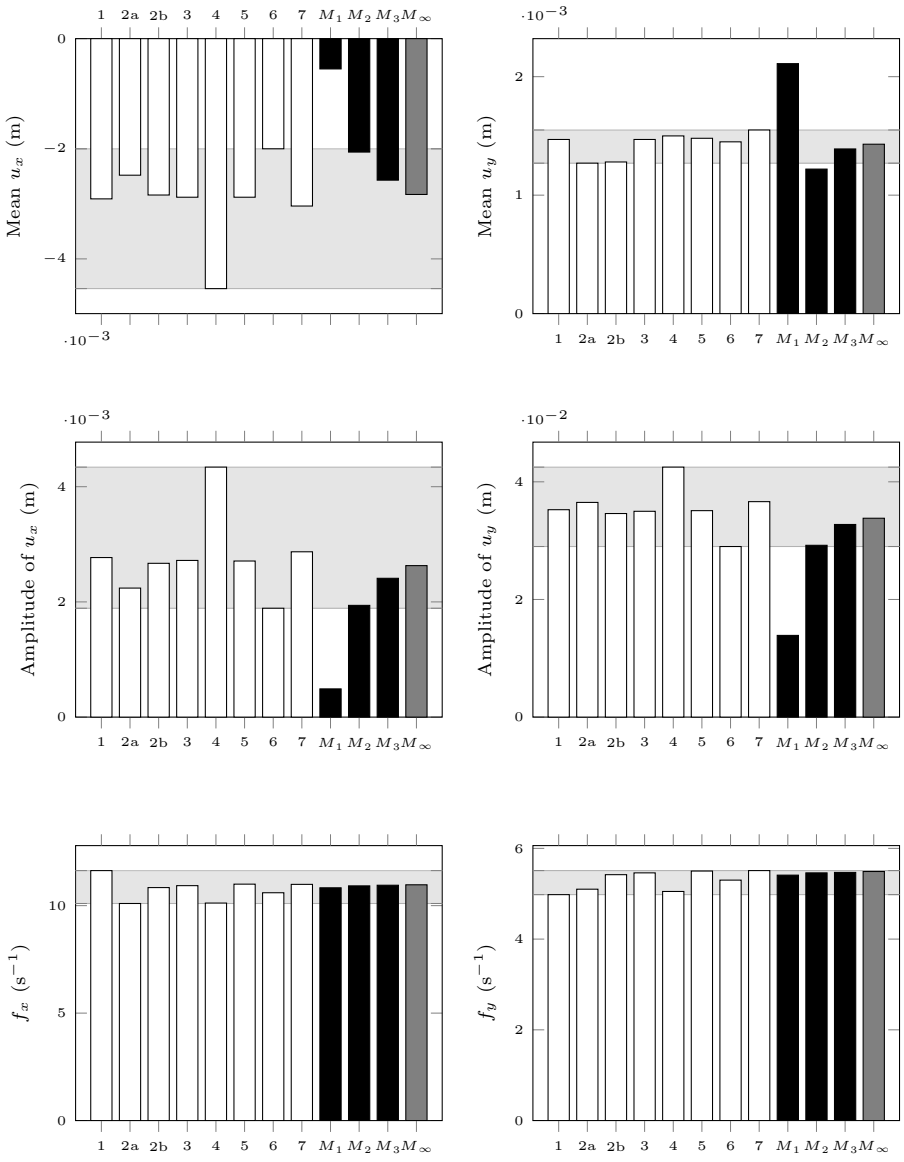


Figure 19: Results obtained for the FSI3 configuration. The first line shows the two components of the mean displacement, the second line the corresponding amplitudes, and the frequencies are displayed in the third line. The results obtained with the methods reported in [59] (displayed in white, and described in Tab. 6) can be compared with the present results, obtained for 3 different mesh resolutions (displayed in black). The extrapolated results are also provided (displayed in gray).

is the one resolving the most of unknowns.

The same procedure as the one described in Eq. (32) can be used to determine the apparent temporal order of the method, considering 3 different time steps dt_1, dt_2 and dt_3 for the simulation, and using the mesh M_2. The beam displacements obtained for these 3 time steps are summarized in Tab. 11 with the corresponding extrapolated values, apparent orders and extrapolated relative errors. The apparent temporal orders obtained are around $p = 1$, which is consistent with the explicit Euler time-integration used for the membrane convection, and extrapolated relative errors are below 1%.

	$\boldsymbol{u_x}$ $\left[\times\mathbf{10^{-3}}\right]$ **(m)**	$\boldsymbol{u_y}$ $\left[\times\mathbf{10^{-3}}\right]$ **(m)**
$dt_1 = 5.00\ 10^{-5}$ s	-2.0489 ± 1.9274	1.1925 ± 29.1467
$dt_2 = 2.50\ 10^{-5}$ s	-2.0403 ± 1.9191	1.1843 ± 29.0895
$dt_3 = 1.25\ 10^{-5}$ s	-2.0363 ± 1.9150	1.1868 ± 29.0565
dt_∞	-2.0326 ± 1.9110	1.1878 ± 29.0109
p	1.07 1.00	1.74 0.79
e_∞ (%)	0.18 0.21	0.09 0.16

Table 11: Determination of the apparent temporal order.

4. Application: flow through an aortic valve

In this section, an application of the method is presented to illustrate its potential. The chosen application is the flow throught an aortic valve, which has been widely investigated in a number of works [67, 68, 69, 70, 71, 72, 73]. Figure 20 (left) shows the meshed fluid geometry used for the computation, corresponding to an idealized geometry of the aorta. The diameter of the aorta is chosen to be $D = 2$ cm. The fluid mesh is composed of 752 460 tetrahedral elements, with a constant mesh resolution of $h = 0.5$ mm. The geometry of the aortic valve is composed of three leaflets, and has been constructed to fit the arotic root, as displayed in Fig. 20 (right). The whole aortic valve is meshed using 450 quadratic hexahedral elements, with an effective resolution of $H =$

0.5 mm.

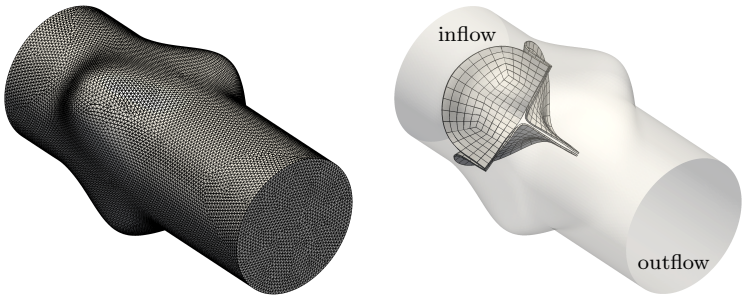


Figure 20: **Left:** Meshed geometry of the idealized aorta. **Right:** Meshed geometry of the model of aortic valve positioned inside the aorta.

The mean velocity prescribed as inflow is given, for $0 < t < T = \dfrac{60}{n_{bpm}}$:

$$U_{mean}(t) = \begin{cases} \dfrac{1}{2}\dfrac{\mu Re}{\rho D}\left[1 + \sin\left(\dfrac{2\pi}{F_I T}\left(t - \dfrac{F_I T}{4}\right)\right)\right] & \text{if } t \leq F_I T \\ \\ 0 & \text{if } t > F_I T \end{cases} \tag{35}$$

where n_{bpm} stands for the number of beats per minutes and F_I is a parameter which allows to set the duration of the injection phase.

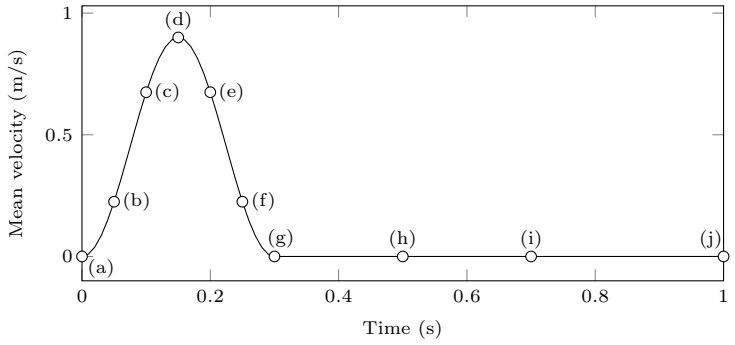


Figure 21: Mean velocity prescribed as inflow. The signal corresponds to one cycle for $F_I = 0.3$. Snapshots at instants (a) to (j) are displayed in Fig. 22-23.

Figure 21 shows the time evolution of the mean velocity $U_{mean}(t)$. The inlet velocity profile is chosen to be parabolic, even if this assumption is not suitable

given the relatively high value of the Womersley number (greater than 10). The parameters of the present study are summarized in Tab. 12, and are inspired from the work of De Hart *et al.* [68]. The leaflets of the valve are modeled by a Neo-Hookean law. The flow solver represents the turbulent flow by Large Eddy Simulation and the Sigma model [74] is used to model the effect of the scales which are too small to be properly discretized by the fluid mesh.

Flow parameters	
Density	$\rho = 1000$ kg/m^3
Dynamic viscosity	$\mu = 4.0\ 10^{-3}$ Pa.s
Reynolds number	$Re = 4500$
Number of beats per minute	$n_{bpm} = 60$
Fraction of injection	$F_I = 0.3$
Valve parameters	
Shear modulus	$G = 3.0\ 10^4$ Pa
Poisson coefficient	$\nu = 0$
Thickness of the leaflets	$e = 0.2$ mm

Table 12: Parameters of the flow and the valve leaflets.

Four cycles were simulated, and different snapshots of the valve deformation over the fourth cycle are displayed in Fig. 22. Labels from (a) to (j) correspond to the ones displayed in Fig. 21. From (a) to (d), the inlet mean velocity is increasing, resulting in the opening of the valve. Regarding snapshot (b) in Fig. 22-23, it is seen that the valve inflates while the opening deformation initiates at the middle of the valve and then spreads toward the extremities. This observation is consistent with the work of Hsu *et al.* [73], where the same opening behavior was observed. From (d) to (g), the inlet velocity is decreasing, and the valve is gradually closing with a strong flapping of the leaflets. At the end of this phase, the inlet velocity is zero, but one leaflet remains open. Then, the leaflet is gradually closing from (g) to (j), and the valve is finally closed at the end of the cycle. One can notice that for the four cycles simulated, this

phenomenon does not appear every time, or when appearing it is not always on the same leaflet (not shown). These cycle-to-cycle variations were expected, given the value of the Reynolds number.

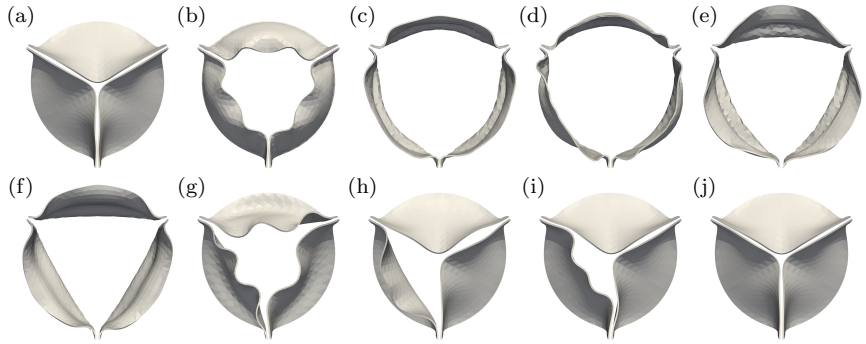


Figure 22: Snapshots of the valve deformation over the fourth cycle.

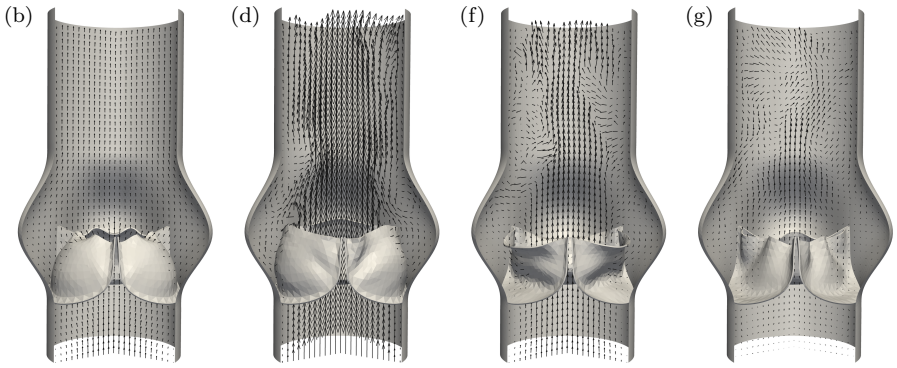


Figure 23: Snapshots of the instantaneous velocity vectors over the fourth cycle. Only four snapshots out of the ten displayed in Fig. 22 are shown.

Figure 23 shows four snapshots of the instantaneous velocity vectors over the fourth cycle. The flapping of the leaflets generates numerous of vortices downstream of the valve, reflecting the presence of flow instabilities, also observed in [68, 70, 73]. Further flow analysis could be conducted, but is out of the scope of this paper. Still, the present simulation has proven the robustness of the ITBM, which leads to a realistic description of a very complex flow at high Reynolds

number and non-trivial deformation of the membrane.

5. Conclusion

A numerical method for simulating the fluid-structure interaction problem in the case of highly deformable membranes has been presented. The approach is based on the immersed boundary method adapted to unstructured grids with the reproducing kernel particle method. The specificity of the approach is that the membrane constituting the immersed boundary is in fact a thick boundary, modeled as a 3D continuum. This enables to use a classical finite-element formulation to solve the membrane mechanics.

The computing accuracy of the method has been demonstrated by a number of test cases where the results provided by the present numerical tool were confronted to either previous numerical, analytical or experimental data. The test cases of section 3 showed that both closed and open membranes can be properly simulated with the ITBM. Simulating very thin membranes having a thickness much smaller than the fluid mesh resolution was expected to lead to problematic issues, especially regarding the procedures of regularization/convection (described in sections 2.2 and 2.4, respectively). The test cases of sections 3.3 and 3.4 however showed that this is actually well managed by the present method.

The ITBM was then used to simulate the flow through a tri-leaflet aortic valve, at a significantly high Reynolds number. The highly complex valve deformation and flow instabilities observed in this study demonstrated the strong robustness of the method, which however needs to be further validated on such a highly complex FSI problem. A combined experimental and numerical study is currently undergoing in this direction, and will be the subject of a future paper.

Acknowledgement

Vincent Moureau and Ghislain Lartigue from the CORIA lab, and the SUCCESS scientific group are acknowledge for providing the YALES2 solver

which constitutes the basis of the YALES2BIO tool. The authors acknowledge supports from ANR (FORCE project ANR-11-JS09-0011), from BPIfrance (DAT@DIAG project) and from the NUMEV Labex. Etienne Gibaud, Vladeta Zmijanovic, Desiree Pott and Simon Sonntag are thanked for helpful discussions. The European Society for Artificial Organs (ESAO) is also thanked for providing a funding. This work was performed using HPC resources from GENCI-CINES (Grants 2014-c2014037194 and 2015-c2015037194).

Appendix: Volume correction

The IBM does not intrinsically conserve the volume enclosed by the flexible membrane as the interpolation of the velocity field required before convecting the solid nodes does not conserve the divergence-free character of the carrying fluid flow. A Lagrange Multiplier method is used to find the smallest correction that should be applied to the solid nodes location in order to ensure the conservation of the particle volume. The membrane middle surface is composed of F triangular elements, and M Lagrangian markers. Each face f has three markers f_1, f_2 and f_3. The volume enclosed by the membrane middle surface can be calculated as:

$$V\left(\underline{\underline{X}}\right) = \frac{1}{18}\sum_{f=1}^{F}\Big[\overrightarrow{X_{f_1}}.\left(\overrightarrow{X_{f_2}}\times\overrightarrow{X_{f_3}}\right) + \overrightarrow{X_{f_2}}.\left(\overrightarrow{X_{f_3}}\times\overrightarrow{X_{f_1}}\right) + \overrightarrow{X_{f_3}}.\left(\overrightarrow{X_{f_1}}\times\overrightarrow{X_{f_2}}\right)\Big]. \tag{36}$$

At the beginning of the calculation, the volume of the particle V_0 is calculated. At the end of each time step, the coordinates of the Lagrangian markers $\underline{\underline{X}}$ are predicted, after time advancement of the solid nodes position. As already stated, $V\left(\underline{\underline{X}}\right) \neq V_0$ since the interpolation does not conserve the divergence-free character of the velocity field.

The aim is then to find the smallest markers displacements $\underline{\underline{\delta X}}$ in norm, so that $V\left(\underline{\underline{X}}^{corr}\right) = V_0$, where $\underline{\underline{X}}^{corr} = \underline{\underline{X}} + \underline{\underline{\delta X}}$ denotes the matrix containing the final coordinates. Introducing a Lagrange multiplier Λ, the sought correction

displacements $\underline{\underline{\delta X}}$ minimize the following cost function:

$$J_\Lambda\left(\underline{\underline{\delta X}}\right)=\sum_{m=1}^{M}\left[(\delta X_m)^2+(\delta Y_m)^2+(\delta Z_m)^2\right]+\Lambda\left[V\left(\underline{\underline{X}}+\underline{\underline{\delta X}}\right)-V_0\right]. \quad (37)$$

Zeroing the partial derivatives of J_Λ with respect to the location correction $(\delta X_m, \delta Y_m, \delta Z_m)$ of the Lagrangian marker m leads to:

$$\begin{cases} 2\delta X_m+\Lambda\dfrac{\partial V\left(\underline{\underline{X}}+\underline{\underline{\delta X}}\right)}{\partial \delta X_m}=0 \\ 2\delta Y_m+\Lambda\dfrac{\partial V\left(\underline{\underline{X}}+\underline{\underline{\delta X}}\right)}{\partial \delta Y_m}=0 \\ 2\delta Z_m+\Lambda\dfrac{\partial V\left(\underline{\underline{X}}+\underline{\underline{\delta X}}\right)}{\partial \delta Z_m}=0 \end{cases} \quad (38)$$

with:

$$\begin{aligned} \frac{\partial V\left(\underline{\underline{X}}+\underline{\underline{\delta X}}\right)}{\partial \delta X_m}=\frac{1}{6}\Bigg[&\sum_{f:\, m=f_1}\frac{\partial\left(\overrightarrow{X_{f_1}}+\overrightarrow{\delta X_{f_1}}\right)}{\partial \delta X_m}\left[\left(\overrightarrow{X_{f_2}}+\overrightarrow{\delta X_{f_2}}\right)\times\left(\overrightarrow{X_{f_3}}\right.\right.\\ &\left.\left.+\overrightarrow{\delta X_{f_3}}\right)\right]+\sum_{f:\, m=f_2}\frac{\partial\left(\overrightarrow{X_{f_2}}+\overrightarrow{\delta X_{f_2}}\right)}{\partial \delta X_m}\left[\left(\overrightarrow{X_{f_3}}+\overrightarrow{\delta X_{f_3}}\right)\times\left(\overrightarrow{X_{f_1}}+\overrightarrow{\delta X_{f_1}}\right)\right] \\ &+\sum_{f:\, m=f_3}\frac{\partial\left(\overrightarrow{X_{f_3}}+\overrightarrow{\delta X_{f_3}}\right)}{\partial \delta X_m}\left[\left(\overrightarrow{X_{f_1}}+\overrightarrow{\delta X_{f_1}}\right)\times\left(\overrightarrow{X_{f_2}}+\overrightarrow{\delta X_{f_2}}\right)\right]\Bigg]. \end{aligned} \quad (39)$$

Assuming that the correction displacements are small, so that $\overrightarrow{X_m}+\overrightarrow{\delta X_m}\approx\overrightarrow{X_m}$, one obtains:

$$\overrightarrow{\delta X_m}=\Lambda\,\overrightarrow{\alpha_m}, \quad (40)$$

with:

$$\begin{aligned} \overrightarrow{\alpha_m}=-\frac{1}{12}\Bigg[&\sum_{f:\, m=f_1}\left(\overrightarrow{X_{f_2}}\times\overrightarrow{X_{f_3}}\right)+\sum_{f:\, m=f_2}\left(\overrightarrow{X_{f_3}}\times\overrightarrow{X_{f_1}}\right) \\ &+\sum_{f:\, m=f_3}\left(\overrightarrow{X_{f_1}}\times\overrightarrow{X_{f_2}}\right)\Bigg]. \end{aligned} \quad (41)$$

When J_Λ is minimum, $\partial J_\Lambda/\partial\Lambda=0$, which means that $V\left(\underline{\underline{X}}+\underline{\underline{\delta X}}\right)=V_0$.

Equations (36) and (40) thus lead to:

$$\frac{1}{18}\sum_{f=1}^{F}\left[\left(\overrightarrow{X_{f_1}}+\Lambda\,\overrightarrow{\alpha_{f_1}}\right).\left[\left(\overrightarrow{X_{f_2}}+\Lambda\,\overrightarrow{\alpha_{f_2}}\right)\times\left(\overrightarrow{X_{f_3}}+\Lambda\,\overrightarrow{\alpha_{f_3}}\right)\right]\right.$$

$$+\left(\overrightarrow{X_{f_2}}+\Lambda\,\overrightarrow{\alpha_{f_2}}\right).\left[\left(\overrightarrow{X_{f_3}}+\Lambda\,\overrightarrow{\alpha_{f_3}}\right)\times\left(\overrightarrow{X_{f_1}}+\Lambda\,\overrightarrow{\alpha_{f_1}}\right)\right] \tag{42}$$

$$\left.+\left(\overrightarrow{X_{f_3}}+\Lambda\,\overrightarrow{\alpha_{f_3}}\right).\left[\left(\overrightarrow{X_{f_1}}+\Lambda\,\overrightarrow{\alpha_{f_1}}\right)\times\left(\overrightarrow{X_{f_2}}+\Lambda\,\overrightarrow{\alpha_{f_2}}\right)\right]\right]-V_0=0.$$

After some algebra, the following third-order polynomial equation in Λ is obtained: $A\Lambda^3+B\Lambda^2+C\Lambda+D=0$, with:

$$A=\frac{1}{18}\sum_{f=1}^{F}\left[\overrightarrow{\alpha_{f_1}}.\left(\overrightarrow{\alpha_{f_2}}\times\overrightarrow{\alpha_{f_3}}\right)+\overrightarrow{\alpha_{f_2}}.\left(\overrightarrow{\alpha_{f_3}}\times\overrightarrow{\alpha_{f_1}}\right)+\overrightarrow{\alpha_{f_3}}.\left(\overrightarrow{\alpha_{f_1}}\times\overrightarrow{\alpha_{f_2}}\right)\right], \tag{43}$$

$$B=\frac{1}{6}\sum_{f=1}^{F}\left[\overrightarrow{X_{f_1}}.\left(\overrightarrow{\alpha_{f_2}}\times\overrightarrow{\alpha_{f_3}}\right)+\overrightarrow{X_{f_2}}.\left(\overrightarrow{\alpha_{f_3}}\times\overrightarrow{\alpha_{f_1}}\right)+\overrightarrow{X_{f_3}}.\left(\overrightarrow{\alpha_{f_1}}\times\overrightarrow{\alpha_{f_2}}\right)\right], \tag{44}$$

$$C=\frac{1}{6}\sum_{f=1}^{F}\left[\overrightarrow{\alpha_{f_1}}.\left(\overrightarrow{X_{f_2}}\times\overrightarrow{X_{f_3}}\right)+\overrightarrow{\alpha_{f_2}}.\left(\overrightarrow{X_{f_3}}\times\overrightarrow{X_{f_1}}\right)+\overrightarrow{\alpha_{f_3}}.\left(\overrightarrow{X_{f_1}}\times\overrightarrow{X_{f_2}}\right)\right], \tag{45}$$

$$D=V\left(\underline{\underline{X}}\right)-V_0. \tag{46}$$

This third-order polynomial equation is then solved numerically and Λ is computed as the real valued root (there is always one at least) of smallest amplitude. Once Λ is found, the Lagrangian markers positions are updated to ensure volume conservation, as follows:

$$\overrightarrow{X_m}\rightarrow\overrightarrow{X_m}+\Lambda\,\overrightarrow{\alpha_m}. \tag{47}$$

References

[1] C. Misbah, Vesicles, capsules and red blood cells under flow, J. Phys.: Conf. Series392 (2012) 012005.

[2] C. Pozrikidis, Boundary Integral and Singularity Methods for Linearized Viscous Flow, Cambridge University Press., 1992.

[3] D. Barthès-Biesel, Capsule motion in flow: Deformation and membrane buckling, Comp. Rend. Phys.10 (8) (2009) 764–774.

[4] D. Barthès-Biesel, Modeling the motion of capsules in flow, Curr. Op. Coll. & Interf. Sc.16 (2011) 3–12.

[5] S. Ramanujan, C. Pozrikidis, Deformation of liquid capsules enclosed by elastic membranes in simple shear flow: large deformations and the effect of fluid viscosities, J. Fluid Mech.361 (1998) 117–143.

[6] E. Lac, A. Morel, D. Barthès-Biesel, Hydrodynamic interaction between two identical capsules in simple shear flow, J. Fluid Mech.573 (2007) 149–169.

[7] W. R. Dodson III, P. Dimitrakopoulos, Dynamics of strain-hardening and strain-softening capsules in strong planar extensional flows via an interfacial spectral boundary element algorithm for elastic membranes, J. Fluid Mech.641 (2009) 263–296.

[8] J. Walter, A.-V. Salsac, D. Barthès-Biesel, P. Le Tallec, Coupling of finite element and boundary integral methods for a capsule in a Stokes flow, Int. J. Numer. Meth. Eng.83 (2010) 829–850.

[9] H. C. Woolfenden, M. G. Blyth, Motion of a two-dimensional elastic capsule in a branching channel flow, J. Fluid Mech.669 (2011) 3–31.

[10] G. Ghigliotti, T. Biben, C. Misbah, Rheology of a dilute two-dimensional suspension of vesicles, J. Fluid Mech.653 (2010) 489–518.

[11] T. Biben, A. Farutin, C. Misbah, Three-dimensional vesicles under shear flow: Numerical study of dynamics and phase diagram, Phys. Rev. E83 (031921).

[12] S. K. Veerapaneni, D. Gueyffier, D. Zorin, G. Biros, A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D, J. Comput. Phys.228 (2009) 2334–2353.

[13] G. Boedec, M. Leonetti, M. Jaeger, 3D vesicle dynamics simulations with a linearly triangulated surface, J. Comput. Phys.230 (2011) 1020–1034.

[14] H. Zhao, E. S. G. Shaqfeh, The dynamics of a vesicle in simple shear flow, J. Fluid Mech.674 (2011) 578–604.

[15] C. Pozrikidis, Numerical simulation of the flow-induced deformation of red blood cells, Ann. of Biomed. Eng.31 (2003) 1194–1205.

[16] H. Zhao, A. H. G. Isfahani, L. N. Olson, J. B. Freund, A spectral boundary integral method for flowing blood cells, J. Comput. Phys.229 (2010) 3726–3744.

[17] Z. Peng, R. J. Asaro, Q. Zhu, Multiscale modelling of erythrocytes in Stokes flow, J. Fluid Mech.686 (2011) 299–337.

[18] T. Gao, H. H. Hu, Deformation of elastic particles in viscous shear flow, J. Comput. Phys.228 (2009) 2132–2151.

[19] C. Bui, V. Lleras, O. Pantz, Dynamics of red blood cells in 2D, ESAIM: Proc.28 (2009) 182–194.

[20] T. Klöppel, W. A. Wall, A novel two-layer, coupled finite element approach for modeling the nonlinear elastic and viscoelastic behavior of human erythrocytes, Biomech. Model. Mechanobiol.10 (2011) 445–459.

[21] T. Biben, C. Misbah, Tumbling of vesicles under shear flow within an advected-field approach, Phys. Rev. E67 (031908).

[22] J. Beaucourt, F. Rioual, T. Séon, T. Biben, C. Misbah, Steady to unsteady dynamics of a vesicle in a flow, Phys. Rev. E69 (011906).

[23] G.-H. Cottet, E. Maitre, A level set method for fluid-structure interactions with immersed surfaces, Math. Mod. Meth. App. Sc.16 (3) (2006) 415–438.

[24] E. Maitre, T. Milcent, G.-H. Cottet, A. Raoult, Y. Usson, Applications of level set methods in computational biophysics, Math. Comput. Mod.49 (11–12) (2009) 2161–2169.

[25] D. Salac, M. Miksis, A level set projection model of lipid vesicles in general flows, J. Comput. Phys.230 (2011) 8192–8215.

[26] A. Laadhari, P. Saramito, C. Misbah, Vesicle tumbling inhibited by inertia, Phys. Fluids24 (2012) 031901.

[27] C. S. Peskin, The immersed boundary method, Acta Num.11 (2002) 479–517.

[28] L. Lee, R. J. Leveque, An immersed interface method for incompressible Navier–Stokes equations, SIAM J. Sci. Comput.25 (3) (2003) 832–856.

[29] X. Wang, W. K. Liu, Extended immersed boundary method using FEM and RKPM, Comput. Meth. Appl. Mech. Eng.193 (2004) 1305–1321.

[30] Y. Kim, M.-C. Lai, Simulating the dynamics of inextensible vesicles by the penalty immersed boundary method, J. Comput. Phys.229 (2010) 4840–4853.

[31] D.-V. Le, B. C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, J. Comput. Phys.220 (2006) 109–138.

[32] D.-V. Le, Large deformation of liquid capsules enclosed by thin shells immersed in the fluid, J. Comput. Phys.229 (2010) 4097–4116.

[33] D.-V. Le, K.-H. Chiam, Hydrodynamic interaction between two nonspherical capsules in shear flow, Phys. Rev. E84 (056322).

[34] P. Bagchi, R. M. Kalluri, Dynamic rheology of a dilute suspension of elastic capsules: effect of capsule tank-treading, swinging and tumbling, J. Fluid Mech.669 (498–526).

[35] C. S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys.25 (1977) 220–252.

[36] S. Mendez, E. Gibaud, F. Nicoud, An unstructured solver for simulations of deformable particles in flows at arbitrary Reynolds numbers, J. Comput. Phys.256 (1) (2014) 465–483.

[37] S. K. Doddi, P. Bagchi, Three-dimensional computational modeling of multiple deformable cells flowing in microvessels, Phys. Rev. E79 (046318).

[38] P. Bagchi, R. M. Kalluri, Rheology of a dilute suspension of liquid-filled elastic capsules, Phys. Rev. E81 (056320).

[39] W. Helfrich, Elastic properties of lipid bilayers: Theory and possible experiments, Z. Naturforsch 28 c (1973) 693–703.

[40] P. Bagchi, A. Z. K. Yazdani, Analysis of membrane tank-tread of nonspherical capsules and red blood cells, Eur. Phys. J. E35 (12103).

[41] A. Z. K. Yazdani, P. Bagchi, Influence of membrane viscosity on capsule dynamics in shear flow, J. Fluid Mech.718 (2013) 569–595.

[42] D.-V. Le, Subdivision elements for large deformation of liquid capsules enclosed by thin shells, Comput. Meth. Appl. Mech. Eng.199 (2010) 2622–2632.

[43] L. Zhang, A. Gerstenberger, X. Wang, W. K. Liu, Immersed finite element method, Comput. Meth. Appl. Mech. Eng.193 (2004) 2051–2067.

[44] W. K. Liu, Y. Liu, D. Farrell, L. Zhang, X. S. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, J. Lee, J. Hong, X. Chen, H. Hsu, Immersed finite element method and its applications to biological systems, Comput. Meth. Appl. Mech. Eng.195 (2006) 1722–1749.

[45] C. Dupont, A.-V. Salsac, B. Barthès-Biesel, M. Vidrascu, P. Le Tallec, Influence of bending resistance on the dynamics of a spherical capsule in shear flow, Phys. Fluids27 (051902).

[46] F. Radjai, F. Dubois (Eds.), Discrete Numerical Modeling of Granular Materials, Wiley-ISTE, 2011.

[47] C. Chnafa, S. Mendez, F. Nicoud, Image-based large-eddy simulation in a realistic left heart, Comput. Fluids94 (2014) 173–187.

[48] V. Zmijanovic, S. Mendez, V. Moureau, F. Nicoud, About the numerical robustness of biomedical benchmark cases: Interlaboratory FDA's idealized medical device, accepted for publication in Int. J. Numer. Meth. Biomed. Eng..

[49] V. Moureau, P. Domingo, L. Vervisch, Design of a massively parallel CFD code for complex geometries, Comp. Rend. Méc.339 (2-3) (2011) 141–148.

[50] J. C. Simo, K. S. Pister, Remarks on rate constitutive equations for finite deformation problems: computational implications, Comput. Meth. Appl. Mech. Eng.46 (1984) 201–215.

[51] W. K. Liu, S. Jun, Y. F. Zhang, Reproducing kernel particle methods, Int. J. Numer. Meth. Fluids20 (1995) 1081–1106.

[52] A. Pinelli, I. Z. Naqavi, U. Piomelli, J. Favier, Immersed-boundary methods for general finite-difference and finite-volume Navier–Stokes solvers, J. Comput. Phys.229 (2010) 9073–9091.

[53] A. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comp.22 (1968) 745–762.

[54] M. Malandin, N. Maheu, V. Moureau, Optimization of the deflated conjugate gradient algorithm for the solving of elliptic equations on massively parallel machines, J. Comput. Phys.238 (2013) 32–47.

[55] K. Y. Sze, X. H. Liu, S. H. Lo, Popular benchmark problems for geometric nonlinear analysis of shells, Finite Elem. Anal. Des.40 (2004) 1551–1569.

[56] E. Lac, D. Barthès-Biesel, N. A. Pelekasis, J. Tsamopoulos, Spherical capsules in three-dimensional unbounded Stokes flows: effect of the membrane constitutive law and onset of buckling, J. Fluid Mech.516 (2004) 303–334.

[57] J. P. Mills, L. Qie, M. Dao, C. T. Lim, S. Suresh, Nonlinear elastic and viscoelastic deformation of the human red blood cell with optical tweezers, Mech. Chem. Biosys.1 (3) (2004) 169–180.

[58] S. Turek, J. Hron, Fluid-Structure Interaction: Modelling, Simulation, Optimisation, Vol. 53, Springer, 2006, Ch. Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow, pp. 371–385.

[59] S. Turek, J. Hron, M. Razzaq, H. Wobker, M. Schäfer, Fluid-Structure Interaction II - Modelling, Simulation, Optimization, Vol. 73, Springer, 2010, Ch. Numerical Benchmarking of Fluid-Structure Interaction: A Comparison of Different Discretization and Solution Approaches, pp. 413–424.

[60] E. A. Evans, Y. C. Fung, Improved measurements of the erythrocyte geometry, Microv. Res.4 (1972) 335–347.

[61] G. Lenormand, S. Hénon, A. Richet, J. Siméon, F. Gallet, Direct measurement of the area expansion and shear moduli of the human red blood cell membrane skeleton, Biophys. J.81 (2001) 43–56.

[62] M. Schäfer, M. Heck, S. Yigit, Fluid-Structure Interaction: Modelling, Simulation, Optimisation, Vol. 53, Springer, 2006, Ch. An Implicit Partitioned Method for the Numerical Simulation of Fluid-Structure Interaction, pp. 171–194.

[63] D. C. Sternel, M. Schäfer, M. Heck, S. Yigit, Efficiency and accuracy of fluid-structure interaction simulations using an implicit partitioned approach, Comput. Mech.43 (2008) 103–113.

[64] U. Küttler, W. A. Wall, Fixed-point fluid-structure interaction solvers with dynamic relaxation, Comput. Mech.43 (2008) 61–72.

[65] W. A. Wall, D. P. Mok, E. Ramm, Partitioned analysis approach of the transient coupled response of viscous fluids and flexible structures, in:

W. Wunderlich (Ed.), Solids, Structures and Coupled Problems in Engineering, Proc. ECCM '99. Munich, August/September 1999, 1999.

[66] I. B. Celik, U. Ghia, P. J. Roache, C. J. Freitas, H. Coleman, P. E. Raad, Procedure for estimation and reporting of uncertainty due to discretization in cfd applications, J. Fluids Eng.130, article id 078001.

[67] J. De Hart, G. W. M. Peters, P. J. G. Schreurs, F. P. T. Baaijens, A two-dimensional fluid-structure interaction model of the aortic value, J. Biomech.33 (2000) 1079–1088.

[68] J. De Hart, G. W. M. Peters, P. J. G. Schreurs, F. P. T. Baaijens, A three-dimensional computational analysis of fluid-structure interaction in the aortic valve, J. Biomech.36 (2003) 103–112.

[69] J. De Hart, G. W. M. Peters, P. J. G. Schreurs, F. P. T. Baaijens, Collagen fibers reduce stresses and stabilize motion of aortic valve leaflets during systole, J. Biomech.37 (2004) 303–311.

[70] I. Borazjani, Fluid-structure interaction, immersed boundary-finite element method simulations of bio-prosthetic heart valves, Comput. Meth. Appl. Mech. Eng.257 (2013) 103–116.

[71] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, T. J. R. Hughes, An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves, Comput. Meth. Appl. Mech. Eng.284 (2015) 1005–1053.

[72] M.-C. Hsu, D. Kamensky, Y. Bazilevs, M. S. Sacks, T. J. R. Hughes, Fluid-structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation, Comput. Mech.54 (2014) 1055–1071.

[73] M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M. C. H. Wu, J. Mineroff, A. Reali, Y. Bazilevs, M. S. Sacks, Dynamic and fluid-structure

interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models, Comput. Mech.55 (2015) 1211–1225.

[74] F. Nicoud, H. Baya Toda, O. Cabrit, S. Bose, J. Lee, Using singular values to build a subgrid-scale model for large eddy simulations, Phys. Fluids23 (085106).