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, France
b Laboratoire de Mécanique et Génie Civil (LMGC), UMR CNRS 5508, University of Montpellier, France

A R T I C L E   I N F O

Article history:
Received 17 November 2015
Received in revised form 27 May 2016
Accepted 23 June 2016
Available online 5 July 2016

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

A B S T R A C T

This paper constitutes an extension of the work of Mendez et al. (2014) [36], 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.

© 2016 Elsevier Inc. All rights reserved.

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 [3–9], vesicles [10–14] and red blood cells [15–17], 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 [18–20]. 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 [23–26] and immersed boundary or immersed interface methods [27–34].

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 interactions of deformable particles in flows at arbitrary Reynolds numbers, in the context of complex geometries often encountered 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 $\Omega_s$ and the fluid domain $\Omega_f$. 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 $\vec{U}_m$ of each solid node $m$, the mechanical force $\vec{F}_m$ resulting from the membrane deformation is calculated,
2. The mechanical force $\vec{F}_m$ is regularized on the fluid mesh, resulting in a field of volumetric force $\vec{f}_j$, calculated on each fluid node $j$, which represents the forces exerted by the membrane on the fluid,
3. The Navier–Stokes equations (forced by the regularized mechanical forces) are solved on the fluid mesh, yielding the velocity of the fluid $\vec{V}_j$ on each fluid node $j$.
4. The velocity of the membrane $\vec{V}_m$ on each solid node $m$ is interpolated from the $\vec{V}_j$ field, enabling to deduce the new position $\vec{X}_m$ from the position at the previous timestep $\vec{X}_m^{\text{previous}}$, such as $\vec{X}_m = \vec{X}_m^{\text{previous}} + \Delta t \vec{V}_m$. The displacement is then updated $\vec{U}_m = \vec{X}_m - \vec{X}_m^0$, where $\vec{X}_m^0$ 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] (http://www.math.univ-montp2.fr/~yales2bio/), 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.

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 $\vec{F}_m$ resulting from the static deformation of the structure, and the inertial force $M \vec{U}_m$ resulting from the dynamics of the structure, the mass matrix $M$ being written as:
where \( \rho_s \) and \( \rho \) 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 \((V \simeq 0)\).

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

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 \( \sigma \) be the Cauchy stress tensor, and \( \epsilon \) the Eulerian–Almansi strain tensor. Using the virtual work principle, it is possible to identify:

\[
\overrightarrow{F}_m \cdot \overrightarrow{U}_m = \int_{\Omega} \sigma : \epsilon \, d\Omega. \tag{2}
\]

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

\[
\overrightarrow{F}_m \cdot \overrightarrow{U}_m = \int_{\Omega^0} \overrightarrow{S} : \overrightarrow{E} \, d\Omega^0, \tag{3}
\]

where \( \overrightarrow{S} \) is the 2nd Piola–Kirchhoff stress tensor, and \( \overrightarrow{E} \) is the Green–Lagrangian strain tensor, which reads:

\[
\overrightarrow{E} = \frac{1}{2} (\overrightarrow{C} - I), \tag{4}
\]

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

\[
\overrightarrow{F}_m \cdot \overrightarrow{U}_m = \int_{\Omega^0} \overrightarrow{S} \cdot \overrightarrow{E} \, d\Omega^0. \tag{5}
\]

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

\[
\overrightarrow{F}_m \cdot \overrightarrow{U}_m = \int_{\Omega^0} \overrightarrow{S} \cdot \overrightarrow{B} \overrightarrow{U}_m \, d\Omega^0. \tag{6}
\]

Eliminating \( \overrightarrow{U}_m \) on both sides and discretizing the domain as a sum of elements \( \Omega^0 = \cup \Omega^0_{el} \), one finally obtains:

\[
\overrightarrow{F}_m = \sum_{\Omega^0_{el}} \left[ \int_{\Omega^0_{el}} \overrightarrow{S} \cdot \overrightarrow{B} \, d\Omega^0 \right]. \tag{7}
\]

In the present paper, both prismatic linear elements (see Fig. 2, left) and hexahedral 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 thickness $e$, the number of elements in the thickness is greater than one.

![Prismatic linear element and Hexahedral quadratic element](image)

**Fig. 2.** Left: Prismatic linear element. Linear shape functions are defined between two element nodes. Right: Hexahedral quadratic element. Interpolation nodes are needed to define the quadratic shape functions.

### 2.1.1. 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
  \[
  S = \frac{2Gv}{1-2v} \text{Tr}(\mathbf{E}) \mathbb{I} + 2G\mathbf{E},
  \]
  where $G$ and $v$ are the shear modulus and the Poisson coefficient, respectively.

- The compressible Neo-Hookean law, as introduced by Simo and Pister [50]
  \[
  \begin{aligned}
  S &= 2 \frac{\partial W}{\partial \mathbf{C}} \\
  W &= \frac{G}{2} (I_1 - 3) - G \ln J + \frac{Gv}{1-2v} (\ln J)^2
  \end{aligned}
  \]  
  where $W$ is the strain energy density function. $J = \det F = \lambda_1\lambda_2\lambda_3$ is the Jacobian of the transformation, expressed as the product of the principal stretches $\lambda_i$ and $I_1$ is the first invariant of the Right Cauchy-Green deformation tensor $\mathbf{C}$, which reads:
  \[
  I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2.
  \]

- The quasi-incompressible Yeoh law
  \[
  \begin{aligned}
  S &= 2 \frac{\partial W}{\partial \mathbf{C}} \\
  W &= C_1 (\tilde{I}_1 - 3) + C_2 (\tilde{I}_1 - 3)^2 + C_3 (\tilde{I}_1 - 3)^3 + \frac{G(1+v)}{3(1-2v)} (\ln J)^2
  \end{aligned}
  \]
  where $C_i$ are material constants and $\tilde{I}_1$ is the first invariant of the isochoric Right Cauchy-Green deformation tensor $\mathbf{\tilde{C}}$, such as $\tilde{I}_1 = J^{-2/3} I_1$. The quantity $2C_1$ can be interpreted to be the shear modulus $G$. 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:
\[
\vec{f}_j = \sum_{m=1}^{M} \left( \vec{F}_m \times w \left( \frac{||\vec{x}_j - \vec{x}_m||}{h} \right) \right),
\]

(12)

where \( M \) is the total number of solid nodes, and \( w \) is a discrete Dirac function allowing to regularize the mechanical force on 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 functions:

\[
w \left( \frac{||\vec{x}_j - \vec{x}_m||}{h} \right) = D \left( \frac{x_j - x_m}{h} \right) D \left( \frac{y_j - y_m}{h} \right) D \left( \frac{z_j - z_m}{h} \right).
\]

(13)

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

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

(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 (RKPM) [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 \( m \):

\[
m_{a,b,c}(\vec{x}_m) = \sum_{j=1}^{J} \left[ \left( \frac{x_j - x_m}{h} \right)^a \left( \frac{y_j - y_m}{h} \right)^b \left( \frac{z_j - z_m}{h} \right)^c \times w \left( \frac{||\vec{x}_j - \vec{x}_m||}{h} \right) \right] dV_j,
\]

(15)

where \( J \) is the total number of fluid nodes. For a unit point force applied at coordinates \( \vec{x}_m \), moments calculated at \( \vec{x}_m \) 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:

\[
w \left( \frac{||\vec{x}_j - \vec{x}_m||}{h} \right) = w(r) = \begin{cases} 
\frac{1}{2^{d-1}} \frac{1}{4h} \left[ 1 + \cos\left(\frac{\pi r}{2}\right) \right] & \text{if } |r| < 2 \\
0 & \text{if } |r| \geq 2
\end{cases}
\]

(16)

Since the resulting regularized force \( \vec{f}_j \) 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:

\[
\overline{w} \left( \frac{||\vec{x}_j - \vec{x}_m||}{h} \right) = w \left( \frac{||\vec{x}_j - \vec{x}_m||}{h} \right) \times \left[ \beta_0 + \beta_1 \frac{x_j - x_m}{h} + \beta_2 \frac{y_j - y_m}{h} + \beta_3 \frac{z_j - z_m}{h} \right]
\]

(17)

where \( \beta_k \) are the coefficients of the polynomial correction of the original window function. The moments of the modified window function are then given by:

\[
\overline{m}_{a,b,c}(\vec{x}_m) = \sum_{j=1}^{J} \left[ \left( \frac{x_j - x_m}{h} \right)^a \left( \frac{y_j - y_m}{h} \right)^b \left( \frac{z_j - z_m}{h} \right)^c \times \overline{w} \left( \frac{||\vec{x}_j - \vec{x}_m||}{h} \right) \right] dV_j.
\]

(18)
By plugging the definition of \( \mathcal{W} \) into Eq. (18), one easily obtains:

\[
\begin{pmatrix}
\beta_0 \\
\beta_1 \\
\beta_2 \\
\beta_3
\end{pmatrix} = \mathcal{M}^{-1} \begin{pmatrix}
m_{0,0} & m_{1,0} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} \\
m_{1,0} & m_{2,0} & m_{1,1} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} \\
m_{0,1} & m_{0,2} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} \\
m_{0,0} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1}
\end{pmatrix} \begin{pmatrix}
\beta_0 \\
\beta_1 \\
\beta_2 \\
\beta_3
\end{pmatrix} = \mathcal{M} \begin{pmatrix}
\beta_0 \\
\beta_1 \\
\beta_2 \\
\beta_3
\end{pmatrix},
\]

(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 \( \beta_k \) such as:

\[
\begin{pmatrix}
\beta_0 \\
\beta_1 \\
\beta_2 \\
\beta_3
\end{pmatrix} = \mathcal{M}^{-1} \begin{pmatrix}
m_{0,0} & m_{1,0} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} \\
m_{1,0} & m_{2,0} & m_{1,1} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} \\
m_{0,1} & m_{0,2} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} \\
m_{0,0} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1}
\end{pmatrix} = \mathcal{M}^{-1} \begin{pmatrix}
1 \\
0 \\
0 \\
0
\end{pmatrix}.
\]

(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 \( \beta_k \) coefficients [52]:

\[
\mathcal{W} \left( \frac{||x_j^m - x_m^m||}{h} \right) = w \left( \frac{||x_j^m - x_m^m||}{h} \right) \times \left[ \beta_0 + \beta_1 \frac{x_j - x_m}{h} + \beta_2 \frac{y_j - y_m}{h} + \beta_3 \frac{z_j - z_m}{h} \right.
\]

\[
\left. + \beta_4 \left( \frac{x_j - x_m}{h} \right)^2 + \beta_5 \left( \frac{y_j - y_m}{h} \right)^2 + \beta_6 \left( \frac{z_j - z_m}{h} \right)^2 \right].
\]

leading to:

\[
\begin{pmatrix}
m_{0,0} & m_{1,0} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} \\
m_{1,0} & m_{2,0} & m_{1,1} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} \\
m_{0,1} & m_{0,2} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} \\
m_{0,0} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1}
\end{pmatrix} \begin{pmatrix}
\beta_0 \\
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4 \\
\beta_5 \\
\beta_6 \\
\beta_7 \\
\beta_8 \\
\beta_9
\end{pmatrix} = \mathcal{M}^{-1} \begin{pmatrix}
m_{0,0} & m_{1,0} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} \\
m_{1,0} & m_{2,0} & m_{1,1} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} \\
m_{0,1} & m_{0,2} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} \\
m_{0,0} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1}
\end{pmatrix} \begin{pmatrix}
1 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{pmatrix}.
\]

(22)

The coefficients \( \beta_k \) are thus given by:

\[
\begin{pmatrix}
\beta_0 \\
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4 \\
\beta_5 \\
\beta_6 \\
\beta_7 \\
\beta_8 \\
\beta_9
\end{pmatrix} = \mathcal{M}^{-1} \begin{pmatrix}
m_{0,0} & m_{1,0} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} \\
m_{1,0} & m_{2,0} & m_{1,1} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} & m_{1,0} \\
m_{0,1} & m_{0,2} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} \\
m_{0,0} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1} & m_{0,1}
\end{pmatrix} \begin{pmatrix}
\beta_0 \\
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4 \\
\beta_5 \\
\beta_6 \\
\beta_7 \\
\beta_8 \\
\beta_9
\end{pmatrix} = \mathcal{M}^{-1} \begin{pmatrix}
1 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{pmatrix}.
\]

(23)

At the end, computing the regularized force as:

\[
\vec{f}_j = \sum_{m=1}^{M} \mathcal{F}_m \times \mathcal{W} \left( \frac{||x_j^m - x_m^m||}{h} \right).
\]

(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:

$$\rho \left( \frac{\partial \vec{v}_j}{\partial t} + \vec{v}_j \cdot \nabla \vec{v}_j \right) = -\nabla p_j + \nabla \left[ \mu \left( \nabla \vec{v}_j + (\nabla \vec{v}_j)^T \right) \right] + \vec{f}_j,$$

where $\vec{v}_j$ and $p_j$ are the velocity vector and pressure on fluid node $j$, $\rho$ the density and $\mu$ the dynamic viscosity. For an incompressible flow, the mass conservation equation becomes:

$$\nabla \cdot \vec{v}_j = 0.$$  

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 (DPCG) 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 $\vec{v}_m$ of the solid node $m$ is calculated by interpolating the fluid velocity on the solid nodes location from the fluid velocity at the neighboring fluid nodes:

$$\vec{v}_m = \sum_{j=1}^{j} \vec{v}_j \times \bar{w} \left( \frac{||\vec{x}_j - \vec{x}_m||}{h} \right) d\vec{v}_j.$$  

This interpolation process again involves the use of the discrete Dirac function $\bar{w}$, 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 performs 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 for the membrane [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 volume of the particle. 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 hexahedral 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 test cases considered in this respect. 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 $U_m$ 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 $F_m$ 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.

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

<table>
<thead>
<tr>
<th>Test case (section)</th>
<th>Diagram</th>
<th>Reference data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-linear bending of an elastic plate (section 3.1)</td>
<td><img src="image1.png" alt="Diagram" /></td>
<td>Numerical results from Sze et al. (2004) [55]</td>
</tr>
<tr>
<td>Inflation of a spherical capsule (section 3.2)</td>
<td><img src="image2.png" alt="Diagram" /></td>
<td>Analytical solution</td>
</tr>
<tr>
<td>Capsule in a linear shear flow (section 3.3)</td>
<td><img src="image3.png" alt="Diagram" /></td>
<td>Numerical results from Lac et al. (2004) [56]</td>
</tr>
<tr>
<td>Red blood cell stretched by optical tweezers (section 3.4)</td>
<td><img src="image4.png" alt="Diagram" /></td>
<td>Experimental data from Mills et al. (2004) [57]</td>
</tr>
<tr>
<td>Flow-induced vibration of an elastic beam behind a cylinder (section 3.5)</td>
<td><img src="image5.png" alt="Diagram" /></td>
<td>Numerical results from Turek and Hron (2006) [58] and Turek et al. (2010) [59]</td>
</tr>
</tbody>
</table>

#### 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 case proposed by Sze et al. (2004) [55], and are given in Table 2.

![Diagram](image6.png)

**Fig. 4.** Geometrical parameters of the plate. The force $F$ is applied over all the surface of the plate extremity.
In order to study the influence of the finite element interpolation, both linear hexahedral finite elements and quadratic hexahedral finite elements are compared with the reference result of Sze et al. (2004) [55]. Fig. 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 maximum 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.

### Table 2

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

<table>
<thead>
<tr>
<th>$F$</th>
<th>$L$</th>
<th>$l$</th>
<th>$e$</th>
<th>$E$</th>
<th>$\nu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 N</td>
<td>10 m</td>
<td>1 m</td>
<td>0.1 m</td>
<td>$1.2 \times 10^6$ Pa</td>
<td>0</td>
</tr>
</tbody>
</table>

**Fig. 5.** Left: Evolution of the maximal displacement as a function of the number of elements. Results obtained with both linear and quadratic hexahedral finite elements are compared with the result of Sze et al. (2004) [55]. Right: Relative error calculated from the most refined computation, for both linear and quadratic interpolations.

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

### 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 hexahedral 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 $h/e = 5$.

At each iteration, the external force $F$ is added to the calculated mechanical force $F_m$. 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 the plate to deform and reach the equilibrium position.

Fig. 6 shows the deformation of the plate (left) and the time evolution of the 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 $\nu$. When submitted to an internal pressure $P$, the capsule deforms (Fig. 7, right).

![Fig. 7. A spherical capsule of internal radius $R_i$ and external radius $R_e$ is inflated applying an internal pressure $P$.](image)

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 $P$:

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

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 Table 3, where $e$ and $R$ denote the thickness and mean radius of the capsule, respectively.

<table>
<thead>
<tr>
<th>$P$</th>
<th>$e$</th>
<th>$R$</th>
<th>$R_i$</th>
<th>$R_e$</th>
<th>$E$</th>
<th>$\nu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 Pa</td>
<td>0.05 m</td>
<td>0.5 m</td>
<td>$R - e/2$</td>
<td>$R + e/2$</td>
<td>$1.5 \times 10^6$ Pa</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Here again, the influence of the finite-element interpolation is investigated. Both linear hexahedral finite elements and quadratic hexahedral finite elements are compared with the analytical solution. Fig. 8 shows the radial displacement $U_r (r)$ (for $r = R_e$) (left) and the relative error (right), 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 $9.38 \times 10^{-2}\%$ for the linear interpolation, and from $2.34\%$ to $7.33 \times 10^{-3}\%$ 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.

![Graph](image)

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

### 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.

![Diagram](image)

**Fig. 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 (Eq. (28)) 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). Fig. 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 fluid, but does not prevent to accurately 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 $\vec{u} = ky \hat{e}_x$ (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). The orientation $\theta$ 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:
\[ D = \frac{A - B}{A + B}. \] (29)

All the parameters relevant to the test case can be found in Table 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 ensure that the membrane is quasi-incompressible.

Fig. 10. Left: An initially spherical capsule is deposited in a linear shear flow. Right: The capsule reaches a steady deformation and orientation, with a tank-treading motion of the membrane.

Table 4

<table>
<thead>
<tr>
<th>Capsule parameters</th>
<th>Flow parameters</th>
<th>Non-dimensional parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial mean radius</td>
<td>Dynamic viscosity</td>
<td>Capillary number ( C_a = \frac{\mu k a}{\varepsilon G} )</td>
</tr>
<tr>
<td>Thickness</td>
<td>Density</td>
<td>Reynolds number ( R_e = \frac{\rho k a^2}{\mu} )</td>
</tr>
<tr>
<td>Shear modulus</td>
<td>Shear rate</td>
<td></td>
</tr>
<tr>
<td>Poisson coefficient</td>
<td>( \nu )</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 11. Visualization 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 fluid domain is extended from \(-16a\) to \(16a\) around the spherical capsule. The fluid mesh is composed of 1924682 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 2906 linear prismatic elements, with an effective resolution of \( H = a/10 \). Fig. 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 behavior of the membrane turning around the inner fluid is observed.

The Taylor parameter $D$ and the capsule orientation $\theta/\pi$ can be calculated from the steady deformation of the capsule. Fig. 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 $D$ ranges from 0.18% to 1.03%, and from 0.64% to 6.08% for the capsule orientation $\theta/\pi$. The maximum 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 ensured in the present computations since using a quasi-incompressible model. Indeed, volume variations of the membrane are still allowed, and are seen to increase with the capillary number (not shown).

![Fig. 12. Left: Evolution of the Taylor parameter $D$ as a function of the capillary number $C_a$. Right: Evolution of the capsule orientation $\theta/\pi$. The present results are compared with the results of Lac et al. (2004) [56].](image)

Influence of the volume correction procedure (see section 2.6) is shown in Table 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 $\theta/\pi$ is however very small.

<table>
<thead>
<tr>
<th>Volume variation (%)</th>
<th>$D$</th>
<th>$\theta/\pi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume correction</td>
<td>$\approx 10^{-4}$</td>
<td>0.5279</td>
</tr>
<tr>
<td>No volume correction</td>
<td>2.73</td>
<td>0.5350</td>
</tr>
</tbody>
</table>

### 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.

![Fig. 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].](image)

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:
\[ z = \pm 0.5R_0 \left[ 1 - \frac{x^2 + y^2}{R_0^2} \right] \left[ A_1 + A_2 \frac{x^2 + y^2}{R_0^2} + A_3 \left( \frac{x^2 + y^2}{R_0^2} \right)^2 \right] \]  

where \( R_0 = 3.91 \mu m \) is the average RBC radius, \( A_1 = 0.207161 \), \( A_2 = 2.002558 \), and \( A_3 = -1.122762 \). 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 \( eG = 7.3 \) μN/m. As in the previous test case, a Poisson coefficient \( \nu = 0.4995 \) is chosen to ensure 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. Fig. 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) \([57]\).

![Fig. 14. Visualization of the red blood cell deformation over the entire range of stretching force. Only half of the cell is displayed.](image)

Fig. 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.

![Fig. 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) \([57]\).](image)

As in section 3.3, the volume correction (see section 2.6) 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 Table 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.

![Fig. 16. Details of the beam-cylinder configuration.](image)

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2a (Rannacher)</td>
<td>Monolithic variational formulation based on a unified Eulerian framework (“interface capturing”).</td>
</tr>
<tr>
<td>2b (Rannacher)</td>
<td>Monolithic variational formulation based on the standard ALE approach (“interface fitting”).</td>
</tr>
<tr>
<td>3 (Turek/Hron)</td>
<td>Fully implicit monolithic ALE-FEM approach.</td>
</tr>
<tr>
<td>4 (Breuer)</td>
<td>Partitioned approach combining the finite-volume scheme FASTEST-3D for the flow and the finite-element structural solver Carat, using an ALE formulation.</td>
</tr>
<tr>
<td>5 (Krafczyk/Rank)</td>
<td>Explicit partitioned approach which combines the Lattice–Boltzmann flow solver VirtualFluids (VF) and the structural p-FEM solver AdhoC.</td>
</tr>
<tr>
<td>6 (Wall)</td>
<td>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.</td>
</tr>
<tr>
<td>7 (Bletzinger)</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 at 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:

\[
v (0, y, z) = 1.5 U \frac{y (H - y)}{(H/2)^{2}},
\]

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 Table 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 Table 8. For each of the meshes, the fluid mesh is tetrahedral, and the mesh of the beam is made of quadratic hexahedral elements. Concerning the fluid, the mesh resolution is not constant over the whole fluid domain, the resolutions provided in Table 8 correspond to the mesh size around the elastic beam.

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

\[
p = \frac{1}{\ln(r)} \ln \frac{\phi_1 - \phi_2}{\phi_2 - \phi_3}.
\]
where \( \phi_k \) denotes the output variable of the simulation for the \( k \)th mesh. The extrapolated value \( \phi_\infty \) can then be calculated as follows:

\[
\phi_\infty = \frac{r^p \phi_3 - \phi_2}{r^p - 1} = \frac{r^p \phi_2 - \phi_1}{r^p - 1}.
\]

(33)

The extrapolated relative error reads:

\[
e_\infty = \left| \frac{\phi_\infty - \phi_3}{\phi_\infty} \right|.
\]

(34)

Table 7
Parameters of the test case, two configurations are simulated: FSI1 and FSI3

<table>
<thead>
<tr>
<th>Flow parameters</th>
<th>FSI1</th>
<th>FSI3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>( \rho = 1000 \text{ kg/m}^3 )</td>
<td>( \rho = 1000 \text{ kg/m}^3 )</td>
</tr>
<tr>
<td>Dynamic viscosity</td>
<td>( \mu = 1 \text{ Pa s} )</td>
<td>( \mu = 1 \text{ Pa s} )</td>
</tr>
<tr>
<td>Inlet velocity</td>
<td>( U = 0.2 \text{ m/s} )</td>
<td>( U = 2 \text{ m/s} )</td>
</tr>
<tr>
<td>Reynolds number</td>
<td>( \rho U D / \mu = 20 )</td>
<td>( \rho U D / \mu = 200 )</td>
</tr>
</tbody>
</table>

Table 8
Characteristics of the fluid and solid meshes, three meshes are considered: \( M_1 \), \( M_2 \) and \( M_3 \)

<table>
<thead>
<tr>
<th>Fluid</th>
<th>( M_1 )</th>
<th>( M_2 )</th>
<th>( M_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of elements</td>
<td>160707</td>
<td>971069</td>
<td>6582336</td>
</tr>
<tr>
<td>Mesh resolution ( h )</td>
<td>( 1.0 \times 10^{-2} \text{ m} )</td>
<td>( 5.0 \times 10^{-3} \text{ m} )</td>
<td>( 2.5 \times 10^{-3} \text{ m} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solid</th>
<th>( M_1 )</th>
<th>( M_2 )</th>
<th>( M_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of elements</td>
<td>72</td>
<td>560</td>
<td>4480</td>
</tr>
<tr>
<td>Effective resolution ( H )</td>
<td>( 1.0 \times 10^{-2} \text{ m} )</td>
<td>( 5.0 \times 10^{-3} \text{ m} )</td>
<td>( 2.5 \times 10^{-3} \text{ m} )</td>
</tr>
</tbody>
</table>

Table 9
Results for the FSI1 configuration.

<table>
<thead>
<tr>
<th>( u_x \times 10^{-5} ) (m)</th>
<th>( u_y \times 10^{-4} ) (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2a</td>
<td>2.4800</td>
</tr>
<tr>
<td>2b</td>
<td>2.2695</td>
</tr>
<tr>
<td>3</td>
<td>2.2705</td>
</tr>
<tr>
<td>5</td>
<td>2.2160</td>
</tr>
<tr>
<td>6</td>
<td>2.2680</td>
</tr>
<tr>
<td>7</td>
<td>2.2640</td>
</tr>
<tr>
<td>( M_1 )</td>
<td>2.8547</td>
</tr>
<tr>
<td>( M_2 )</td>
<td>2.6724</td>
</tr>
<tr>
<td>( M_3 )</td>
<td>2.5841</td>
</tr>
<tr>
<td>( M_\infty )</td>
<td>2.5012</td>
</tr>
<tr>
<td>( p )</td>
<td>1.05</td>
</tr>
<tr>
<td>( e_\infty ) (%)</td>
<td>3.32</td>
</tr>
</tbody>
</table>

The extrapolated values of the 2 components of the displacement of the beam extremity (corresponding to the infinitely refined mesh \( M_\infty \)) as well as the corresponding apparent spatial order \( p \) and extrapolated relative error are also summarized in Table 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 Table 9 are displayed in Fig. 17. A meaningful comparison between the results reported 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 11250 (for method 2a) to 19320832 (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 (in gray), it is seen that the present results converge to the range of results reported in [59].

![Fig. 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 Table 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).](image1)

![Fig. 18. Visualization 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.](image2)

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 Table 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 solicitations.

Fig. 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 Table 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).

The results of Table 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 11250 (for method 2a) to 2480814 (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 (in gray) are very close to the results obtained with method 5, which is the one resolving the most of unknowns.

Table 10
Results for the FSI3 configuration.

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

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 Table 11 with the corresponding extrapolated values, apparent orders and extrapolated relative errors. The apparent temporal orders obtained are around $p = 1$, and extrapolated relative errors are below 1%.

Table 11
Determination of the apparent temporal order.

<table>
<thead>
<tr>
<th>$dt$ (s)</th>
<th>$u_x \times 10^{-3}$ (m)</th>
<th>$u_y \times 10^{-3}$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$dt_1 = 5.00 \times 10^{-5}$ s</td>
<td>$-2.0489 \pm 1.9274$</td>
<td>$1.1925 \pm 29.1467$</td>
</tr>
<tr>
<td>$dt_2 = 2.50 \times 10^{-5}$ s</td>
<td>$-2.0403 \pm 1.9191$</td>
<td>$1.1843 \pm 29.0895$</td>
</tr>
<tr>
<td>$dt_3 = 1.25 \times 10^{-5}$ s</td>
<td>$-2.0363 \pm 1.9150$</td>
<td>$1.1868 \pm 29.0565$</td>
</tr>
<tr>
<td>$dt_{\infty}$</td>
<td>$-2.0326 \pm 1.9110$</td>
<td>$1.1878 \pm 29.0109$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>$1.07 \pm 1.00$</td>
<td>$1.74 \pm 0.79$</td>
</tr>
<tr>
<td>$\varepsilon_{\infty}$ (%)</td>
<td>$0.18 \pm 0.21$</td>
<td>$0.09 \pm 0.16$</td>
</tr>
</tbody>
</table>

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 through an aortic valve, which has been widely investigated in a number of works [67–73]. Fig. 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 aortic 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.

Fig. 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 $0 < t < T = \frac{60}{n_{bpm}}$:

$$U_{\text{mean}}(t) = \begin{cases} 
\frac{1}{2} \frac{\mu \text{Re}}{\rho D} \left[ 1 + \sin \left( \frac{2\pi}{F_I T} \left( t - \frac{F_I T}{4} \right) \right) \right] & \text{if } t \leq F_I T \\
0 & \text{if } t > F_I T 
\end{cases}$$  \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. Fig. 21 shows the time evolution of the mean velocity $U_{\text{mean}}(t)$. 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 Table 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 (Eq. (9)). 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.

![Fig. 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 Figs. 22–23.](image)

**Table 12**

Parameters of the flow and the valve.

<table>
<thead>
<tr>
<th>Flow parameters</th>
<th>Valve parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>$\rho_s = 1000 \text{ kg/m}^3$</td>
</tr>
<tr>
<td>Dynamic viscosity</td>
<td>$\mu = 4.0 \times 10^{-3} \text{ Pa.s}$</td>
</tr>
<tr>
<td>Reynolds number</td>
<td>$\text{Re} = 4500$</td>
</tr>
<tr>
<td>Number of beats per minute</td>
<td>$n_{bpm} = 60$</td>
</tr>
<tr>
<td>Fraction of injection</td>
<td>$F_I = 0.3$</td>
</tr>
</tbody>
</table>

![Fig. 22. Snapshots of the valve deformation over the fourth cycle.](image)

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 Figs. 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 (the commissures).
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 (see snapshot (h) in Fig. 22). 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.

Fig. 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.

![Fig. 23. Snapshots of the instantaneous velocity vectors over the fourth cycle. Only four snapshots out of the ten displayed in Fig. 22 are shown.](image)

5. Conclusion

A numerical method for simulating fluid–structure interaction problems 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 this 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 3.1, 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 complex FSI problem. A combined experimental and numerical study is currently undergoing in this direction, and will be the subject of a future paper.

Acknowledgements

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 N°11112018W) and from the NUMEV Labex (ANR-10-LABX-20). 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 A. 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:

$$V \left( \vec{X} \right) = \frac{1}{18} \sum_{f=1}^{F} \left[ \vec{X}_{f_1} \cdot \left( \vec{X}_{f_2} \times \vec{X}_{f_3} \right) + \vec{X}_{f_2} \cdot \left( \vec{X}_{f_3} \times \vec{X}_{f_1} \right) + \vec{X}_{f_3} \cdot \left( \vec{X}_{f_1} \times \vec{X}_{f_2} \right) \right].$$  \hspace{1cm} (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 $\vec{X}$ are predicted, after time advancement of the solid nodes position. As already stated, $V \left( \vec{X} \right) \neq V_0$ since the interpolation does not conserve the divergence-free character of the velocity field.

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

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

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

$$\begin{aligned}
2\delta X_m + \Lambda \frac{\partial V \left( \vec{X} + \delta \vec{X} \right)}{\partial \delta X_m} &= 0 \\
2\delta Y_m + \Lambda \frac{\partial V \left( \vec{X} + \delta \vec{X} \right)}{\partial \delta Y_m} &= 0 \\
2\delta Z_m + \Lambda \frac{\partial V \left( \vec{X} + \delta \vec{X} \right)}{\partial \delta Z_m} &= 0
\end{aligned} \hspace{1cm} (38)

with:

$$\frac{\partial V \left( \vec{X} + \delta \vec{X} \right)}{\partial \delta X_m} = \frac{1}{6} \left[ \sum_{f: m = f_1} \frac{\partial \left( \vec{X}_{f_1} + \delta \vec{X}_{f_1} \right)}{\partial \delta X_m} \left( \vec{X}_{f_2} + \delta \vec{X}_{f_2} \right) \times \left( \vec{X}_{f_3} + \delta \vec{X}_{f_3} \right) \right]$$

$$+ \sum_{f: m = f_2} \frac{\partial \left( \vec{X}_{f_2} + \delta \vec{X}_{f_2} \right)}{\partial \delta X_m} \left( \vec{X}_{f_3} + \delta \vec{X}_{f_3} \right) \times \left( \vec{X}_{f_1} + \delta \vec{X}_{f_1} \right)$$

$$+ \sum_{f: m = f_3} \frac{\partial \left( \vec{X}_{f_3} + \delta \vec{X}_{f_3} \right)}{\partial \delta X_m} \left( \vec{X}_{f_1} + \delta \vec{X}_{f_1} \right) \times \left( \vec{X}_{f_2} + \delta \vec{X}_{f_2} \right).$$  \hspace{1cm} (39)

Assuming that the correction displacements are small, so that $\vec{X}_m + \delta \vec{X}_m \approx \vec{X}_m$, one obtains:

$$\delta \vec{X}_m = \Lambda \vec{\alpha}_m,$$  \hspace{1cm} (40)

with:

$$\vec{\alpha}_m = -\frac{1}{12} \left[ \sum_{f: m = f_1} \left( \vec{X}_{f_2} \times \vec{X}_{f_3} \right) + \sum_{f: m = f_2} \left( \vec{X}_{f_3} \times \vec{X}_{f_1} \right) + \sum_{f: m = f_3} \left( \vec{X}_{f_1} \times \vec{X}_{f_2} \right) \right].$$  \hspace{1cm} (41)

When $J_{\Lambda}$ is minimum, $\partial J_{\Lambda} / \partial \Lambda = 0$, which means that $V \left( \vec{X} + \delta \vec{X} \right) = V_0$. Equations (36) and (40) thus lead to:
\[
\begin{align*}
\frac{1}{18} & \sum_{f=1}^{F} \left[ (\mathbf{x}_{f1} + \Lambda \mathbf{\alpha}_{f1}) \cdot (\mathbf{x}_{f2} + \Lambda \mathbf{\alpha}_{f2}) \times (\mathbf{x}_{f3} + \Lambda \mathbf{\alpha}_{f3}) \right] \\
& \quad + (\mathbf{x}_{f1} + \Lambda \mathbf{\alpha}_{f1}) \cdot (\mathbf{x}_{f2} + \Lambda \mathbf{\alpha}_{f2}) \times (\mathbf{x}_{f3} + \Lambda \mathbf{\alpha}_{f3}) \\
& \quad + (\mathbf{x}_{f1} + \Lambda \mathbf{\alpha}_{f1}) \cdot (\mathbf{x}_{f2} + \Lambda \mathbf{\alpha}_{f2}) \times (\mathbf{x}_{f3} + \Lambda \mathbf{\alpha}_{f3}) \right] - V_0 = 0.
\end{align*}
\]

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

\[
\begin{align*}
A &= \frac{1}{18} \sum_{f=1}^{F} \left[ (\mathbf{\alpha}_{f1} \cdot (\mathbf{x}_{f2} \times \mathbf{x}_{f3}) + \mathbf{\alpha}_{f2} \cdot (\mathbf{x}_{f3} \times \mathbf{x}_{f1}) + \mathbf{\alpha}_{f3} \cdot (\mathbf{x}_{f1} \times \mathbf{x}_{f2})) \right], \\
B &= \frac{1}{6} \sum_{f=1}^{F} \left[ \mathbf{x}_{f1} \cdot (\mathbf{x}_{f2} \times \mathbf{x}_{f3}) + \mathbf{x}_{f2} \cdot (\mathbf{x}_{f3} \times \mathbf{x}_{f1}) + \mathbf{x}_{f3} \cdot (\mathbf{x}_{f1} \times \mathbf{x}_{f2}) \right], \\
C &= \frac{1}{6} \sum_{f=1}^{F} \left[ \mathbf{x}_{f1} \cdot (\mathbf{\alpha}_{f2} \times \mathbf{x}_{f3}) + \mathbf{x}_{f2} \cdot (\mathbf{\alpha}_{f3} \times \mathbf{x}_{f1}) + \mathbf{x}_{f3} \cdot (\mathbf{\alpha}_{f1} \times \mathbf{x}_{f2}) \right], \\
D &= V \left( \frac{\Lambda}{\Lambda} \right) - V_0.
\end{align*}
\]

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

\[
\mathbf{x}_{m} \rightarrow \mathbf{x}_{m} + \Lambda \mathbf{\alpha}_{m}.
\]

References
