POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Fictitious domain methods for simulating fluid-structure interaction of cardiac valves

Laurea Magistrale in Mathematical Engineering - Ingegneria Matematica

Author: Federica Botta

Advisor: Prof. Luca Dede'

Co-advisors: Pasquale Africa, Michele Bucelli, Ivan Fumagalli

Academic year: 2021-2022

1. Introduction

The cardiac valves are one of the heart's most important components. In recent years, the *Immersed Boundary Method* [5] has become widespread in modeling the interaction between cardiac valves and blood since it is particularly suitable for simulating scenarios where the deformations of the solid body are large. The most recent formulation, proposed by Boffi, Cavallini, and Gastaldi in [2], is called the *Fictitious Domain* approach. This last approach has not been applied to the simulation of cardiac valves in blood flow yet. Therefore, this thesis aims to fill the gap in the literature by reproducing these realistic simulations employing the Fictitious Domain approach. In order to do so, we implement this method in `life`^{x} [1], a high-performance library for solving multi-physics and multi-scale problems. We initially test the implemented solver on simple physical scenarios. Then, we proceed to simulate more complex scenarios, particularly the simulation of a cardiac valve in a blood vessel under open valve condition. The numerical method allows us to accurately reproduce the physiological processes of the blood vessel's opening and closing due to the cardiac valves' oscillation.

2. Mathematical model

Let $\Omega \subset \mathbb{R}^d$, with $d = 2, 3$, be a region containing both a fluid and a structure. The *Navier-Stokes equations* describe the motion of a viscous, incompressible, homogeneous Newtonian flow. The fluid velocity is denoted by $\mathbf{u} : \Omega \times (0, T] \to \mathbb{R}^d$ and the fluid pressure by $p : \Omega \times (0, T] \to \mathbb{R}$. The physical constants of the fluid include the density ρ_{f}, the dynamic viscosity μ_{f} and the *Cauchy stress tensor* $\boldsymbol{\sigma}_{\mathrm{f}}$. For the solid mechanics, an incompressible visco-hyperelastic material that fills at time $t \in (0, T]$ a region $\mathcal{B}_t \subset \Omega$. The solid density, dynamic viscosity and Cauchy stress tensor are denoted by ρ_{s}, μ_{s} and $\boldsymbol{\sigma}_{\mathrm{s}}$, respectively. The latter can be decomposed into the sum of two contributions: viscous and pure elastic parts. The key feature of this method is considering the structure overlapping with the fluid. Therefore, the resulting physical constants consist of a combination of the fluid and the solid ones. It is typically assumed in the immersed methods literature that fluid and solid materials have the same viscosity $\mu = \mu_{\mathrm{f}} = \mu_{\mathrm{s}}$. For this reason, the overall Cauchy stress tensor is:

$$\boldsymbol{\sigma} = \begin{cases} \boldsymbol{\sigma}^{\mathrm{v}} & \text{in } \Omega \setminus \mathcal{B}_t, \\ \boldsymbol{\sigma}^{\mathrm{v}} + \boldsymbol{\sigma}^{\mathrm{e}} & \text{in } \mathcal{B}_t. \end{cases}$$

Moreover, the density of the system is the following:

$$\rho = \begin{cases} \rho_{\mathrm{f}} & \text{in } \Omega \setminus \mathcal{B}_t, \\ \rho_{\mathrm{s}} & \text{in } \mathcal{B}_t \end{cases}$$

and their difference is $\delta_\rho = \rho_{\mathrm{s}} - \rho_{\mathrm{f}}$. In this manner, fluid and solid can be treated in their natural formulation: *Eulerian* for the fluid and *Lagrangian* for the solid. To consider the equation of motion of the solid in the Lagrangian description, a reference domain $\mathcal{B} \subset \mathbb{R}^d$ is selected. Then, $\mathcal{B}_t$ is represented as the image of a position $\mathbf{X}$ using $\mathbf{s} \in \mathcal{B}$, such that:

$$\mathbf{X}(\cdot, t) : \mathcal{B} \to \mathcal{B}_t$$
$$\mathbf{x} \in \mathcal{B}_t \iff \mathbf{x} = \mathbf{X}(\mathbf{s}, t) \text{ for some } \mathbf{s} \in \mathcal{B}.$$

We assume that the solid is moving at the same velocity as the fluid, such that:

$$\mathbf{u}(\mathbf{x}, t) = \frac{\partial \mathbf{X}}{\partial t}(\mathbf{s}, t) \qquad \text{for } \mathbf{x} = \mathbf{X}(\mathbf{s}, t).$$

This condition can be interpreted as a constraint for the problem. Therefore, it can be enforced in a variational form by introducing the *Distributed Lagrange multipliers*. Let us set $\mathbf{\Lambda} = \left[H^1(\mathcal{B})\right]^d$ and the bilinear forms $\mathbf{c}_1, \mathbf{c}_2$ as the scalar product in $\left[H^1(\mathcal{B})\right]^d$ space. Hence, the kinematic constraint is rewritten as:

$$\mathbf{c}_1(\boldsymbol{\nu}, \mathbf{u}(\mathbf{X}(\cdot, t), t)) - \mathbf{c}_2\left(\boldsymbol{\nu}, \frac{\partial \mathbf{X}}{\partial t}(t)\right) = 0 \quad \forall \boldsymbol{\nu} \in \mathbf{\Lambda}$$

Finally, assuming that the external volume forces are zero, the following saddle point problem is obtained:

Problem 1. Given $\mathbf{u}_0 \in \left[H^1_0(\Omega)\right]^d$ and $\mathbf{X}_0 \in \left[W^{1,\infty}(\mathcal{B})\right]^d$. Find, for almost every $t \in (0, T]$, $(\mathbf{u}(t), p(t)) \in \left[H^1_0(\Omega)\right]^d \times L^2_0(\Omega)$, $\mathbf{X}(t) \in \left[H^1(\mathcal{B})\right]^d$ and $\boldsymbol{\lambda}(t) \in \mathbf{\Lambda}$ such that there holds:

$$\begin{cases} \rho_{\mathrm{f}}\left(\dfrac{\mathrm{d}}{\mathrm{d}t}(\mathbf{u}(t), \mathbf{v}) + ((\mathbf{u}(t) \cdot \boldsymbol{\nabla})\, \mathbf{u}(t), \mathbf{v})\right) \\ \quad +2\mu(\epsilon(\mathbf{u}(t)), \epsilon(\mathbf{v})) - (\boldsymbol{\nabla} \cdot \mathbf{v}, p(t)) \\ \quad +\mathbf{c}_1(\boldsymbol{\lambda}(t), \mathbf{v}(\mathbf{X}(t))) = 0 & \forall \mathbf{v} \in \left[H^1_0(\Omega)\right]^d, \\ (\boldsymbol{\nabla} \cdot \mathbf{u}(t), q) = 0 & \forall q \in L^2_0(\Omega), \\ \delta_\rho\left(\dfrac{\partial^2 \mathbf{X}}{\partial t^2}(t), \mathbf{Y}\right)_{\mathcal{B}} + (\mathbb{P}^{\mathrm{e}}(\mathbb{F}(t)), \boldsymbol{\nabla}_{\mathbf{s}} \mathbf{Y})_{\mathcal{B}} \\ \quad -\mathbf{c}_2(\boldsymbol{\lambda}(t), \mathbf{Y}) = 0 & \forall \mathbf{Y} \in \left[H^1(\mathcal{B})\right]^d, \\ \mathbf{c}_1(\boldsymbol{\nu}, \mathbf{u}(\mathbf{X}(\cdot, t), t)) - \mathbf{c}_2\left(\boldsymbol{\nu}, \dfrac{\partial \mathbf{u}}{\partial t}(t)\right) = 0 & \forall \boldsymbol{\nu} \in \mathbf{\Lambda}, \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) & \text{in } \Omega, \\ \mathbf{X}(\mathbf{s}, 0) = \mathbf{X}_0(\mathbf{s}) & \text{in } \mathcal{B}, \\ \dfrac{\partial \mathbf{X}}{\partial t}(\mathbf{s}, 0) = \mathbf{u}_0(\mathbf{x}) & \text{for } \mathbf{x} = \mathbf{X}(\mathbf{s}, 0). \end{cases}$$

The elastic stress is written by means of the *first Piola-Kirchhoff stress tensor* $\mathbb{P}^{\mathrm{e}}(\mathbb{F})$ where $\mathbb{F} = \boldsymbol{\nabla}_{\mathbf{s}} \mathbf{X}$ is the deformation gradient. It is important to define the correct constitutive equation in relation to the case we are facing, for example, the *linear elastic model*, the *St. Venant-Kirchhoff model* and the *Neo-Hookean model*.

2.1. Time advancing scheme

Let us introduce the time-discretized formulation of Problem 1. The time interval $(0, T]$ is partitioned into N intervals of length Δt and denoting the time-discrete variables as $\mathbf{u}^{n+1}(\mathbf{x}) \approx \mathbf{u}(\mathbf{x}, t^{n+1})$, $p^{n+1}(\mathbf{x}) \approx p(\mathbf{x}, t^{n+1})$, $\mathbf{X}^{n+1}(\mathbf{s}) \approx \mathbf{X}(\mathbf{s}, t^{n+1})$ and $\boldsymbol{\lambda}^{n+1}(\mathbf{s}) \approx \boldsymbol{\lambda}(\mathbf{s}, t^{n+1})$. Two choices for the time discretization scheme are presented: the *semi-implicit* and the *fully-implicit* schemes. The time-discrete problem is:

Problem 2. Given $\mathbf{u}_0 \in \left[H^1_0(\Omega)\right]^d$ and $\mathbf{X}_0 \in \left[W^{1,\infty}(\mathcal{B})\right]^d$, for $n = 0, \ldots, N-1$ find $(\mathbf{u}^{n+1}, p^{n+1}) \in \left[H^1_0(\Omega)\right]^d \times L^2_0(\Omega)$, $\mathbf{X}^{n+1} \in \left[H^1(\mathcal{B})\right]^d$ and $\boldsymbol{\lambda}^{n+1} \in \mathbf{\Lambda}$ such that it holds:

$$\begin{cases} \rho_{\mathrm{f}}\left(\left(\dfrac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t}, \mathbf{v}\right) + ((\mathbf{u}^* \cdot \boldsymbol{\nabla})\, \mathbf{u}^{n+1}, \mathbf{v})\right) \\ \quad +2\mu(\epsilon(\mathbf{u}^{n+1}), \epsilon(\mathbf{v})) - (\boldsymbol{\nabla} \cdot \mathbf{v}, p^{n+1}) \\ \quad +\mathbf{c}_1(\boldsymbol{\lambda}^{n+1}, \mathbf{v}(\mathbf{X}^*)) = 0 & \forall \mathbf{v} \in \left[H^1_0(\Omega)\right]^d, \\ (\boldsymbol{\nabla} \cdot \mathbf{u}^{n+1}, q) = 0 & \forall q \in L^2_0(\Omega), \\ \delta_\rho\left(\dfrac{\mathbf{X}^{n+1} - 2\mathbf{X}^n + \mathbf{X}^{n-1}}{\Delta t^2}, \mathbf{Y}\right)_{\mathcal{B}} - \mathbf{c}_2(\boldsymbol{\lambda}^{n+1}, \mathbf{Y}) \\ \quad +(\mathbb{P}^{\mathrm{e}}(\boldsymbol{\nabla}_{\mathbf{s}} \mathbf{X}^{n+1}), \boldsymbol{\nabla}_{\mathbf{s}} \mathbf{Y})_{\mathcal{B}} = 0 & \forall \mathbf{Y} \in \left[H^1(\mathcal{B})\right]^d, \\ \mathbf{c}_1(\boldsymbol{\nu}, \mathbf{u}^{n+1}(\mathbf{X}^*)) - \mathbf{c}_2\left(\boldsymbol{\nu}, \dfrac{\mathbf{X}^{n+1} - \mathbf{X}^n}{\Delta t}\right) = 0 \ \forall \boldsymbol{\nu} \in \mathbf{\Lambda}, \\ \mathbf{u}^0(\mathbf{x}) = \mathbf{u}_0(\mathbf{x}) & \text{in } \Omega, \\ \mathbf{X}^0(\mathbf{s}) = \mathbf{X}_0(\mathbf{s}) & \text{in } \mathcal{B}. \end{cases}$$

Remark: if semi-implicit scheme is considered $\mathbf{u}^* = \mathbf{u}^n$, $\mathbf{X}^* = \mathbf{X}^n$, or if the fuylly-implicit scheme is considered $\mathbf{u}^* = \mathbf{u}^{n+}$, $\mathbf{X}^* = \mathbf{X}^{n+1}$.

2.2. Finite element discretization

Let us introduce the spatial discretization to recover the fully discretized formulation of the Fictitious Domain method. Let $\mathcal{T}_h$ be a triangulation of Ω and consider two *inf-sup compatible* finite element spaces $\mathbf{V}_h \subset \left[H^1_0(\Omega)\right]^d$ and $Q_h \subset L^2_0(\Omega)$. Similarly, consider a triangulation $\mathcal{T}_h^{\mathcal{B}}$ of $\mathcal{B}$. $\mathbf{S}_h$ is the finite element space of piecewise linear d-vectors defined on $\mathcal{B}$ as follows:

$$\mathbf{S}_h = \{\mathbf{Y} \in C^0(\mathcal{B}; \Omega) : \mathbf{Y}|_{T_k} \in \left[\mathcal{P}^1(T_k)\right]^d, \quad k = 1, \ldots, M_e\},$$

where $\mathcal{P}^1(T_k)$ stands for the space of affine polynomials on the element $T_k \in \mathcal{T}_h^{\mathcal{B}}$. Moreover, we consider $\mathbf{\Lambda}_h \subset \mathbf{\Lambda} = \left[H^1(\mathcal{B})\right]^d$. The discrete solution $\mathbf{u}_h^{n+1} \in \mathbf{V}_h$ is the spatially-discrete approximation of $\mathbf{u}^{n+1}$. Similarly, $p_h^{n+1} \approx p^{n+1}$, $\mathbf{X}_h^{n+1} \approx \mathbf{X}^{n+1}$ and $\boldsymbol{\lambda}_h^{n+1} \approx \boldsymbol{\lambda}^{n+1}$. Then, the fully discretized formulation of Problem 1 is the following:

Problem 3. Given $\mathbf{u}_{0,h} \in \mathbf{V}_h$ and $\mathbf{X}_0 \in \left[W^{1,\infty}(\mathcal{B})\right]^d$, for $n = 0, \dots, N-1$ find $(\mathbf{u}_h^{n+1}, p_h^{n+1}) \in \mathbf{V}_h \times Q_h$, $\mathbf{X}_h^{n+1} \in \mathbf{S}_h$ and $\boldsymbol{\lambda}_h^{n+1} \in \mathbf{\Lambda}_h$ such that:

$$
\begin{cases}
\rho_{\mathrm{f}}\left(\left(\dfrac{\mathbf{u}_h^{n+1}-\mathbf{u}_h^n}{\Delta t}, \mathbf{v}\right) + \left(\left(\mathbf{u}_h^* \cdot \boldsymbol{\nabla}\right)\mathbf{u}_h^{n+1}, \mathbf{v}\right)\right) & \\
\quad +2\mu(\epsilon(\mathbf{u}_h^{n+1}), \epsilon(\mathbf{v})) - (\boldsymbol{\nabla}\cdot\mathbf{v}, p_h^{n+1}) & \\
\quad +\mathbf{c}_1(\boldsymbol{\lambda}_h^{n+1}, \mathbf{v}(\mathbf{X}_h^*)) = 0 & \forall \mathbf{v} \in \mathbf{V}_h, \\
(\boldsymbol{\nabla}\cdot\mathbf{u}_h^{n+1}, q) = 0 & \forall q \in Q_h, \\
\delta_\rho\left(\dfrac{\mathbf{X}_h^{n+1}-2\mathbf{X}_h^n+\mathbf{X}_h^{n-1}}{\Delta t^2}, \mathbf{Y}\right)_{\mathcal{B}} - \mathbf{c}_2(\boldsymbol{\lambda}_h^{n+1}, \mathbf{Y}) & \\
\quad +(\mathbb{P}^{\mathrm{e}}(\boldsymbol{\nabla}_{\mathbf{s}}\mathbf{X}_h^{n+1}), \boldsymbol{\nabla}_{\mathbf{s}}\mathbf{Y})_{\mathcal{B}} = 0 & \forall \mathbf{Y} \in \mathbf{S}_h, \\
\mathbf{c}_1(\boldsymbol{\nu}, \mathbf{u}_h^{n+1}(\mathbf{X}_h^*)) - \mathbf{c}_2\left(\boldsymbol{\nu}, \dfrac{\mathbf{X}_h^{n+1}-\mathbf{X}_h^n}{\Delta t}\right) = 0 & \forall \boldsymbol{\nu} \in \mathbf{\Lambda}_h, \\
\mathbf{u}_h^0 = \mathbf{u}_0 & \text{in } \mathcal{T}_h, \\
\mathbf{X}_h^0 = \mathbf{X}_0 & \text{in } \mathcal{T}_h^{\mathcal{B}}.
\end{cases}
$$

This formulation was proven to be unconditionally stable both for semi-implicit and fully-implicit time discretization schemes.

2.3. Algebraic system

The system is solved using an *iterative algorithm* called *Newton's method*. $\{\boldsymbol{\varphi}_i\}_{i=1}^{N_u}, \{\psi_i\}_{i=1}^{N_p}, \{\boldsymbol{\chi}_i\}_{i=1}^{N_X}$ and $\{\boldsymbol{\xi}_i\}_{i=1}^{N_\lambda}$ denote the basis functions in $\mathbf{V}_h$, Q_h, $\mathbf{S}_h$ and $\mathbf{\Lambda}_h$. The following matrices are defined:

$$
\begin{aligned}
&(M_{\mathrm{f}})_{ji} = \rho_{\mathrm{f}}(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i), && i,j = 1,\dots,N_u,\\
&(K_{\mathrm{f}})_{ji} = 2\mu(\epsilon(\boldsymbol{\varphi}_j), \epsilon(\boldsymbol{\varphi}_i)), && i,j = 1,\dots,N_u,\\
&C(\mathbf{u})_i = \rho_{\mathrm{f}}((\mathbf{u}\cdot\boldsymbol{\nabla})\,\mathbf{u}, \boldsymbol{\varphi}_i), && i = 1,\dots,N_u,\\
&B_{ki} = -(\boldsymbol{\nabla}\cdot\boldsymbol{\varphi}_i, \psi_k), && i = 1,\dots,N_u,\ k = 1,\dots,N_p,\\
&(M_{\mathrm{s}})_{ji} = \delta_\rho(\boldsymbol{\chi}_j, \boldsymbol{\chi}_i)_{\mathcal{B}}, && i,j = 1,\dots,N_X,\\
&K_{\mathrm{s}}(\mathbf{X})_i = (\mathbb{P}^{\mathrm{e}}(\boldsymbol{\nabla}_{\mathbf{s}}\mathbf{X}), \boldsymbol{\nabla}_{\mathbf{s}}\boldsymbol{\chi}_i)_{\mathcal{B}}, && i = 1,\dots,N_X,\\
&L_{\mathrm{f}}(\mathbf{X})_{lj} = \mathbf{c}_1(\boldsymbol{\xi}_l, \boldsymbol{\varphi}_j(\boldsymbol{X})), && j = 1,\dots,N_u,\ l = 1,\dots,N_\lambda,\\
&(L_{\mathrm{s}})_{lj} = \mathbf{c}_2(\boldsymbol{\xi}_l, \boldsymbol{\chi}_j) && j = 1,\dots,N_X,\ l = 1,\dots,N_\lambda.
\end{aligned}
$$

$\mathbf{u}^{n+1} \in \mathbb{R}^{N_u}$, $p^{n+1} \in \mathbb{R}^{N_p}$, $\mathbf{X}^{n+1} \in \mathbb{R}^{N_X}$ and $\boldsymbol{\lambda}^{n+1} \in \mathbb{R}^{N_\lambda}$ denote the vectors of coefficients in the expansion of the discrete solution with respect to the Finite Element basis. In particular, let us name the solution vector for the velocity variable $\mathbf{u}^{n,k}$ for a specific n iteration in time and k iteration number of the Newton's algorithm. The correspondent Jacobian matrix of the system is:

$$
J^{n+1,k} = \begin{bmatrix}
A_{\mathrm{f}}(\mathbf{u}^{n+1,k}) & B^T & 0 & L_{\mathrm{f}}(\mathbf{X}^{n+1,k})^T \\
B & 0 & 0 & 0 \\
0 & 0 & A_{\mathrm{s}}(\mathbf{X}^{n+1,k}) & -L_{\mathrm{s}}^T \\
L_{\mathrm{f}}(\mathbf{X}^{n+1,k}) & 0 & -L_{\mathrm{s}} & 0
\end{bmatrix},
$$

where:

$$
\begin{aligned}
A_{\mathrm{f}}(\mathbf{u}^{n+1,k}) &= M_{\mathrm{f}} + K_{\mathrm{f}} + \frac{\mathrm{d}C}{d\mathbf{u}}(\mathbf{u}^{n+1,k}),\\
A_{\mathrm{s}}(\mathbf{X}^{n+1,k}) &= M_{\mathrm{s}} + \frac{\mathrm{d}K_{\mathrm{s}}}{d\mathbf{X}}(\mathbf{X}^{n+1,k}).
\end{aligned}
$$

Consequently, the associated residual vector $\mathbf{R}^{n+1,k}$ is defined. In conclusion, the formulation of the solution of Problem 3 by means of Newton's method is:

Problem 4. Given $\mathbf{u}_0 \in \mathbb{R}^{N_u}$ and $\mathbf{X}_0 \in \mathbb{R}^{N_X}$. For $n = 0, \dots, N-1$, find $\mathbf{U}^{n+1} = [\mathbf{u}^{n+1}, p^{n+1}, \mathbf{X}^{n+1}, \boldsymbol{\lambda}^{n+1}]^T \in \mathbb{R}^{N_u} \times \mathbb{R}^{N_p} \times \mathbb{R}^{N_X} \times \mathbb{R}^{N_\lambda}$ as result of the following problem:

Given initial guess $\mathbf{U}^{n+1,0}$, find $\mathbf{U}^{n+1,k+1} \in \mathbb{R}^{N_u} \times \mathbb{R}^{N_p} \times \mathbb{R}^{N_X} \times \mathbb{R}^{N_\lambda}$ solving the following system for $k = 0, \dots, N_{\max} - 1$ until convergence:

$$
\begin{cases}
J^{n+1,k}\,\delta\mathbf{U}^{n+1,k+1} = \mathbf{R}^{n+1,k}, \\
\mathbf{U}^{n+1,k+1} = \mathbf{U}^{n+1,k} - \delta\mathbf{U}^{n+1,k+1}.
\end{cases}
\tag{7a}
$$

2.4. FSI-SIMPLE preconditioner

The *FSI-SIMPLE* preconditioner is proposed in [4]. Its purpose is to accelerate the solution of the system (7a). In [4], the preconditioner is used to solve FSI problems while considering the movement of the fluid domain. Therefore, a modification of the FSI-SIMPLE preconditioner P_{FS} for use with the immersed methods is presented.

Given the residual $\mathbf{R} = [\mathbf{R_u}, \mathbf{R}_p, \mathbf{R_X}, \mathbf{R_\lambda}]$, the steps needed to compute $\mathbf{U} = P_{\mathrm{FS}}^{-1}\mathbf{R}$ are the following:

1. $\mathbf{X} = A_{\mathrm{s}}^{-1}\mathbf{R_X}$,
2. $b_{\boldsymbol{\lambda}} = L_{\mathrm{s}}\mathbf{X} + \mathbf{R_\lambda}$,
3. $p = \tilde{S}^{-1}(BA_{\mathrm{f}}^{-1}\mathbf{R_u} - \mathbf{R}_p)$,
4. $\boldsymbol{\lambda} = \tilde{S}_c^{-1}(L_{\mathrm{f}}A_{\mathrm{f}}^{-1}\mathbf{R_u} - b_{\boldsymbol{\lambda}})$,
5. $\mathbf{u} = A_{\mathrm{f}}^{-1}\mathbf{R_u} - D^{-1}B^T p - D^{-1}L_{\mathrm{f}}^T\boldsymbol{\lambda}$.

According to [4], the FSI-SIMPLE preconditioner is not optimal as it requires a high number of iterations to converge. However, it is still employed in this thesis as it results in fewer iterations than running without a preconditioner. It is noteworthy that as the number of Degrees of Freedom of the solid domain increases, the number of iterations required also increases, as well reaching several hundred. Therefore, we cannot choose a solid mesh size too small to avoid a high running time, but at the same time, we limit the accuracy of the solution.

3. Implementation in the software library `life`^{x}

Before the approach of this thesis, `life`^{x} had only implemented the *Arbitrary Lagrangian-Eulerian method* to deal with Fluid-Structure interaction problems. Hence, the implementation of the Fictitious Domain approach is an innovation. The `FictitiousDomain` class is the primary class and defines the members and methods required to solve Problem 7a. Many components were introduced, such as a new algorithm called *assembling without intersection* to compute the fluid-solid coupling block, the implementation of the *shared parallelization*, which allows for the simultaneous processing of the Jacobian matrix assembling, and the implementation of the FSI-SIMPLE preconditioner as the `FSIPreconditioners::FSI_SIMPLE` class.

4. Numerical results

At first, we verify the implemented solver by simulating physical situations and comparing the expected behaviour with the solver's result.

Floating or sinking of a solid object: we reproduce the effect of the Buoyancy's force in a scenario where fluid and solid have different densities and both are subjected to the gravity force. The results are the floating or sinking of the solid object accordingly with being less or more dense than the fluid. Simultaneously, the solid is subjected to the effect of the interaction forces with the fluid and deforms itself. The resulting pressure of the fluid domain increases towards the bottom due to the gravity force. On the other hand, the velocity magnitude is conditioned by the presence of the solid object inside, which causes an increase in velocity. The glyph of the fluid domain shows the appearance of vortexes which cause the solid to rise or sink. Hence, in the two cases, their direction is opposite.

Solid object dragged along the channel: we reproduce a solid object dragged along the channel driven by the fluid flow. The expected behaviour is confirmed by moving from the inlet to the outlet side. Unlike the previous tests, the fluid-structure interaction forces do not significantly alter the solid shape. The resulting pressure decreases from the inlet side to the outlet side, accordingly to the Neumann boundary conditions imposed. Meanwhile, the velocity magnitude increases over time, accelerating the movement of the solid object. In conclusion, the presence of the solid does not produce a relevant change in the velocity.

Vesicle in a lid-driven cavity flow: we simulate the movement of a vesicle in a lid-driven cavity flow. It is worth noting that this is the only verification test inspired by the literature. The results obtained are compared with those in [3] and found to be consistent. Specifically, the vesicle moves clockwise, reaching the top side while undergoing a deformation. The glyph of the fluid domain reveals the presence of a vortex created by the non-homogeneous Dirichlet boundary condition applied at the top, which causes the solid to move.

The previous simulations are reproduced both in two and three-dimensional spaces, except for the vesicle benchmark, which is only in two-dimensional space. These results show the effectiveness of the implementation in predicting particular behaviours. Now, we are going to reproduce a more complex scenario: the cardiac valves within a blood vessel. This is the main goal of this thesis. The test case is inspired from [3]. At first, we compared the results using the same physical parameters of the literature and found them coherent. Then, we reproduce a more realistic scenario using the blood's physical parameters. The configuration is an *open valve* setting, where the fluid domain is modeled as a blood vessel with two valves attached at the fluid's boundary. This configuration allows a space between the valves, ensuring that the valves never touch each other and that the blood flow is never entirely restricted. The fluid flow at the inlet is parabolic and sinusoidal in time, creating an oscillation of the opening and closing of the blood vessel by the cardiac valves, trying to represent the physiological process as much as possible. Moreover, to decrease the number of necessary Degrees of Freedom for the fluid domain, we divide the configuration into two symmetric parts, each with one cardiac valve. For mirroring the result obtained on a halve, we impose a symmetric boundary condition on the axis or surface of symmetry.

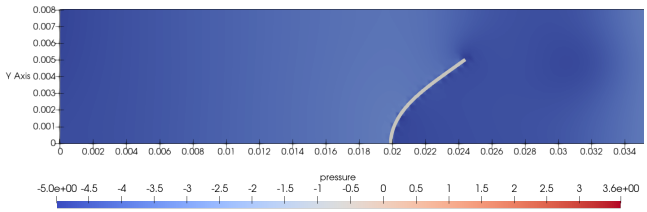
Open valve in two-dimensional space: The physical parameters used in this case are written in Table 1.

Physical parameter	Value
Density of the fluid	$1000\ \mathrm{kg/m^2}$
Density of the solid	$1000\ \mathrm{kg/m^2}$
Viscosity	$0.004\ \mathrm{Pa \cdot s}$
Lamé's first coefficient	$0\ \mathrm{Pa}$
Lamé's second coefficient	$3.57 \cdot 10^7\ \mathrm{Pa}$

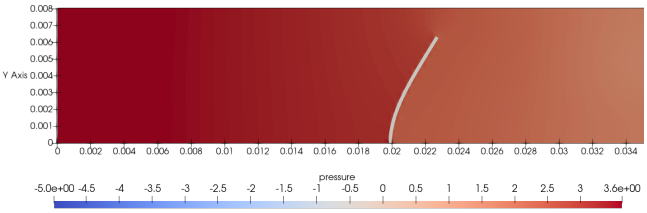
Table 1: Physical parameters for the two-dimensional open valve benchmark.

The constitutive equation used to model the elastic first Piola-Kirchhoff stress tensor of the solid is the Neo-Hookean model. However, we impose Lamé's first coefficient as null to minimize the compressibility of the solid material. The *Reynolds number* of the problem is $\mathrm{Re} = 142$. Despite not being very high in normal physiology, it is suitable for a two-

dimensional space. Figures 1 and 2 depict the oscillation of the cardiac valve between the opening of the blood vessel and its closure. The results obtained are similar to the processes in normal physiology. Moreover, we can observe in Figure 2 how the velocity affects the movement of the cardiac valve: the blood vessel is completely open when the inlet velocity is at its maximum and closed when the inlet velocity is null. An increase in velocity is observed between the two valves when the blood vessel is completely open. On the other hand, Figure 1 depicts the pressure in the fluid domain, which varies in relation to the inlet velocity and the position of the solid. In particular, the pressure is low when the blood vessel is open. On the contrary, when it is closed, the pressure is high.

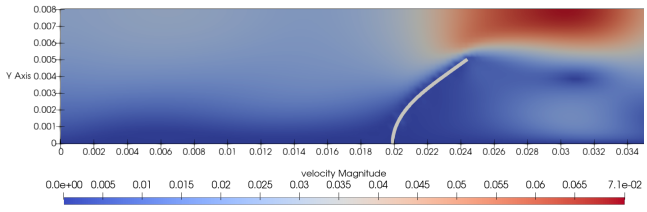


(a) $t = 0.45$ s.

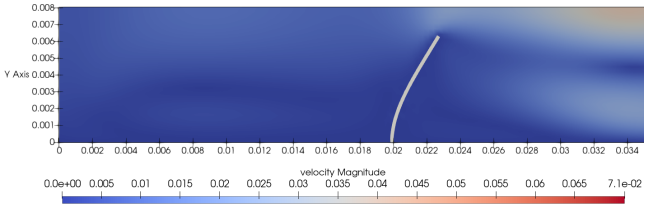


(b) $t = 0.85$ s.

Figure 1: Resulting pressure and solid position.



(a) $t = 0.45$ s.



(b) $t = 0.85$ s.

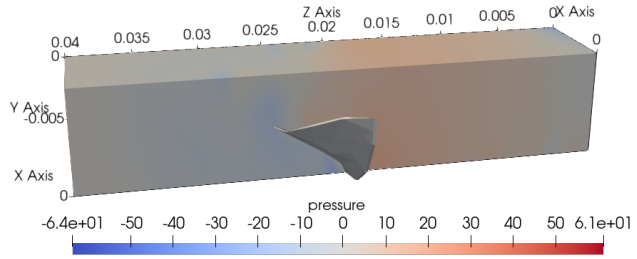
Figure 2: Resulting velocity magnitude and solid position.

Open valve in three-dimensional space: The physical parameters used in this benchmark are written in Table 2.

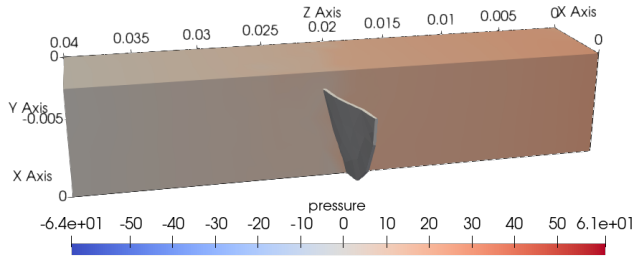
Physical parameter	Value
Density of the fluid	$1000\ \text{kg/m}^3$
Density of the solid	$1000\ \text{kg/m}^3$
Viscosity	$0.004\ \text{Pa} \cdot \text{s}$
Lamé's first coefficient	$4.8 \cdot 10^6\ \text{Pa}$
Lamé's second coefficient	$1.2 \cdot 10^6\ \text{Pa}$

Table 2: Physical parameters for the three-dimensional open valve benchmark.

The constitutive equation used to model the elastic first Piola-Kirchhoff stress tensor of the solid is the Neo-Hookean model. The Reynolds number of the problem is $\text{Re} = 435$, a typical value found in blood vessels like arteries and veins. Figures 3 and 4 show the movement of the cardiac valve in response to the blood flow. We can observe that as the blood flows, the valve is pushed along to the z-axis and lowers in the y-direction, deforming itself to allow unimpeded blood flow, representing the opening process of a blood vessel. As the inlet velocity decreases, the valve tends to return to its initial position, limiting the blood flow and representing the closing process of a blood vessel. In particular, Figure 4 represents the velocity magnitude of the fluid domain. We observe that the blood vessel is open when the fluid flows in and closed when it stops. Moreover, an increase in velocity is observed between the two valves when the blood vessel is open. Finally, Figure 3 depicts the pressure of the fluid domain. The pressure is higher in the half domain where fluid enters due to the cardiac valves restricting blood flow. Moreover, the pressure varies in time with the inlet velocity and the solid position.

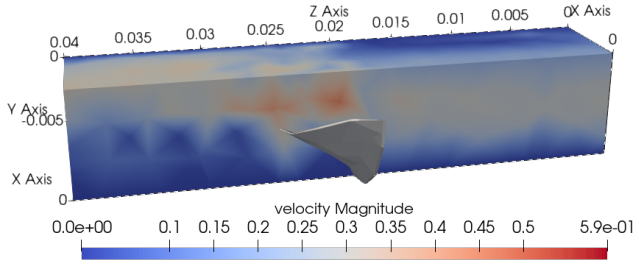


(a) $t = 0.45$ s.

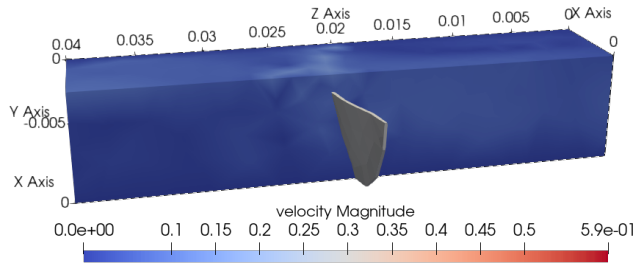


(b) $t = 0.85$ s.

Figure 3: Resulting pressure sliced along the x-axis and solid position.



(a) $t = 0.45$ s.



(b) $t = 0.85$ s.

Figure 4: Resulting velocity magnitude sliced along the x-axis and solid position.

In conclusion, the realistic simulation of the cardiac valves in blood flow is able to simulate correctly the physiological processes.

5. Conclusions

In this work, we realistically reproduced the interaction between cardiac valves and the blood flow by means of the Fictitious Domain method. To the best of our knowledge, the newest formulation of the immersed methods is applied in this framework for the first time in literature. To achieve this, the method was implemented in `life`^{x} . This was the first time this method was used in the library, and it expanded its capabilities for conducting realistic simulations. We simulated the open valve condition in two and three-dimensional spaces with an inlet velocity which was parabolic and sinusoidal in time. The simulation results in the blood vessel's opening and closing due to the cardiac valves' oscillation driven by the blood flow. This simple approximation of the physiological process represents the first step towards a more accurate simulation. For future improvements, a more accurate simulation can be obtained by optimizing the implemented solver and, consequently, decreasing the mesh size of both domains. Moreover, other benchmark tests can be simulated as the closed valve condition or two valves in contact. The method could also be extended to consider the case of a compressible solid or with different viscosity than the fluid one. Moreover, the union with another method, such as the Arbitrary Lagrangian-Eulerian method, can be implemented to simulate the blood vessel's elastic boundary.

References

[1] Pasquale Africa. lifex: a flexible, high performance library for the numerical solution of complex finite element problems. 07 2022.

[2] Daniele Boffi, Nicola Cavallini, and Lucia Gastaldi. The finite element immersed boundary method with distributed lagrange multiplier. *SIAM Journal on Numerical Analysis*, 53, 07 2014.

[3] Ludovic Boilevin-Kayl, Miguel A. Fernández, and Jean-Frédéric Gerbeau. Numerical methods for immersed fsi with thin-walled structures. *Computers & Fluids*, 179:744–763, 2019.

[4] Gwenol Grandperrin. Parallel preconditioners for navier-stokes equations and fluid-structure interaction problems application to hemodynamics. 2013.

[5] Charles Peskin. The immersed boundary method. *Acta Numerica*, 11:479 – 517, 01 2002.