# An Eulerian finite-volume approach of fluid-structure interaction problems on quadtree meshes

Michel Bergmann, Antoine Fondanèche, Angelo Iollo

# An Eulerian finite-volume approach of fluid-structure interaction problems on quadtree meshes


## Abstract

A quadtree-based fully Eulerian finite volume approach for the simulation of fluid-structure interaction problems is presented. Both fluid and structure phases are solved monolithically on the whole computational domain. The discretization stencils are limited to the first layer of neighbors thus enhancing the efficiency of the parallel computations while limiting the numerical order of the finite volume discretizations that can be reached. The behavior of hyperelastic structures is described with the non-linear Mooney-Rivlin model. The simulation of several two dimensional test cases is performed on uniform and quadtree grids and results are compared with the literature. To illustrate the versatility of the numerical model presented, a biomedical application, the axisymmetric simulation of a blood flow in a cardiac pump, is presented.


## 1 Introduction

The numerical simulation of fluid-structure interactions (FSI) with large deformations has several significant applications in engineering, biology and medecine. For instance, the flow around offshore wind turbines involves large displacements and large deformations caused by the slender structure of the blades and by the strong wind-floating-mooring coupling. In biology, there is more and more interest in studying flight or swim strokes at all scales from cyliated bacteria to whales. In medicine, besides the classical application of the pulsatile flow inside large vessels, there is now increasing concern about bio-mimetic, low-impact, blood-pumping devices exploiting large deformations of oscillating membranes. In all these cases, accurate FSI simulations provide an essential enabler to understand these complex phenomena from a quantitative point of view.

Classically, the FSI numerical modeling is based on Lagrangian approaches. The Lagrangian description is essentially based on interface-tracking and body-fitted grids that follow the interface that separates two specific models, one for the fluid and one for the elastic medium. Models such as Arbitrary Lagrangian-Eulerian (ALE) methods [26, 44] or Deforming Spatial-Domain/Space-Time (DSD/ST) [41, 42] methods belong to this class of approaches. Lagrangian methods are known to be very efficient to describe small/medium size deformations. They ensure a precise representation of the geometry and the imposition of transmission conditions at the interface is straightforward since the fluid-structure interface is explicitly tracked. However, when the deformations become large, or there are topological variations (splitting, fusion) or surface contacts (collisions), Lagrangian methods become challenging because of grid deformation and distortions that often result in remeshing.

In this work, we consider a fully Eulerian approach to FSI modeling. The fluid and the structure are considered as a multi-phase incompressible material governed by the same set of equations. The constitutive law differs nevertheless in the fluid and in the elastic medium and the transmission conditions are imposed in a weak form. The novelty of this paper is the discretization of this model which is performed on an hierarchical adaptive grid that is not

necessarily body-fitted. Hierarchical Cartesian schemes allow the multi-scale resolution of PDEs on non body-fitted meshes with a drastic reduction of the computational setup overhead since the mesh generation step is simplified. These methods are easily parallelizable and they can efficiently be mapped to high-performance computer architectures. Moreover, coupled with an Eulerian model, they avoid dealing with grid remeshing, a prohibitive task when the boundaries are moving and the topology is complex and unsteady. In turn, transmission conditions require more care. In order to simplify this step, we employ a diffuse-interface method in which the discontinuous properties of the medium are smoothed across the interface and the boundary conditions are satisfied by continuity of the fluxes and of the solution. The fluid-structure interface itself is captured using an Eulerian level-set function defined as the signed Euclidean distance to the interface.

In the past decades, Eulerian FSI models have been proposed by various authors [8, 14, 30, 32, 38, 45]. In this approach, the deformation of the material is generally followed using a vector field called backward characteristics, representing the inverse mapping of the deformation vector. These functions, connecting the current position of a particle to its initial position in the undeformed configuration of the material, are transported over time according to the velocity, and are used to determine the elastic stress tensor. With a different perspective, in Sugiyama et al. [38] it is suggested to follow the evolution of the symmetrical left Cauchy-Green tensor via an inhomogeneous advection equation which explicitly depends on the velocity gradient.

In this paper, we consider the unified Eulerian model proposed by Cottet et al. [8]. It has been more recently employed by Deborde et al. [11] for the simulation of the interaction between a solitary wave and series of immersed elastic structures. Using a fully uniform Cartesian method, the authors highlight the phenomenon of wave damping in coastal areas. For the resolution of this Eulerian model, we propose a fully finite-volume method on hierarchical Cartesian grids, through the open-source Bitpit/PABLO[1] library. This library gives the user the opportunity to perform massively parallel computations, while maintaining an optimal scalability of the parallelism. However, the numerical order of accuracy is limited since only compact stencils, *i.e.* the first layer of neighbours, are involved in the discretization schemes.

The resolution of the model equations is performed in the whole computational domain, *i.e.* both inside the materials and inside the fluid. When the fluid is subjected to high shear deformations, the elongation of the backward characteristics can become critical. It leads to a steep growth of some components of the stress tensor in the vicinity of the interface, which can induce large oscillations, especially when the material is stiff or when the deformation is very large. To overcome this problem, authors in the literature adopt different strategies. In the work of Sugiyama et al. [38], the Cauchy-Green tensor is continuously reset to the zero tensor inside the fluid in order to avoid the generation of instabilities close to the interface. A second approach is based on the Initial Position Set (IPS) method. On the one hand, Dunne and Rannacher [14] propose to define the advection velocity inside the fluid by harmonic continuation of the structure via an IPS function. On the other hand, Rannacher and Richter [30] use a technique that is closer to ALE approaches, for which the fluid velocity is updated by solving a Laplace problem in the fluid, with respect to the boundary conditions at the interface on which the velocity is well-defined. In the idea of generalizing these IPS methods, Richter [32] introduce a new approach, based on an extension of the solid deformation on a small layer of cells outside the elastic structure. To this end, a constant extension is performed. Recently, Deborde et al. [11] have proposed an improvement of this method by using a linear extrapolation [3] of the backward characteristics inside the fluid. In our study, this extrapolation technique is also employed, coupled with a Fast-Marching method to update the level-set function in order to recover a suitable computation of the elastic forces.

The paper is organized as follows: in Section 2, the Eulerian FSI model is described.

[1] https://optimad.github.io/bitpit

Section 3 is devoted to the numerical method employed to simulate FSI problems. To validate this FSI approach, numerical tests are then outlined in Section 4. In particular, we evaluate the accuracy and efficiency of the method with respect to the literature. Finally, we conclude and give some possible perspectives of this work in Section 5.

## 2 Flow configuration and governing equations

The flow configuration under consideration is depicted in Figure 1. The computational domain $\Omega \subset \mathbb{R}^2$ is divided into tree distinct subdomains related to each medium such that $\Omega = \Omega_f \cup \Omega_s \cup \Omega_e$. The domain filled by the fluid is $\Omega_f$, the domain covered by the elastic material is $\Omega_e$, and the non-deformable solid domains are $\Omega_s$. In that multi-material system, the elastic materials can move and deform according to some deformation laws, whereas for a non-deformable rigid structure, the position and motion are known beforehand. In what follows, the rigid domains $\Omega_s$ can be either included in an elastic material or included in the fluid. The inclusion in the elastic material can be used to induce a certain deformation, it is hence called a holder. The inclusions in the fluid are obstacles and can represent immersed boundary conditions defining complex flow geometries. The interface between the elastic material and the fluid is $\Gamma_e$, and the interfaces between the solid and another phase (elastic or fluid) is $\Gamma_s$. Except the whole domain $\Omega$, all other subdomains and interfaces can evolve in time.

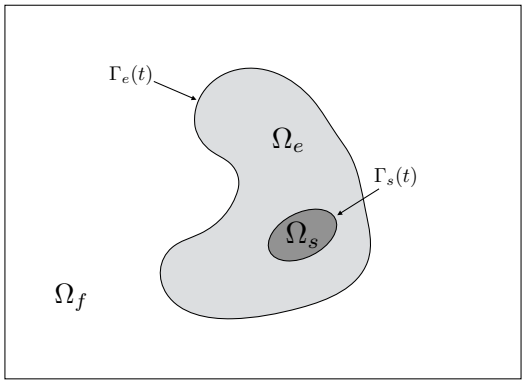



**Figure 1:** *Sketch of the fluid-structure interaction configuration.*

We consider an incompressible fluid flow and a material with incompressible elasticity. All material properties like dynamic viscosity or density are denoted with appropriate subscripts ($f$, $e$ or $s$) corresponding to each material. The unknown velocity and pressure fields are defined by $\mathbf{u}$ and $p$ indifferently in each phase. An Eulerian description is adopted to model both the fluid and the elastic material.

- The fluid flow is governed by the incompressible Navier-Stokes equations in domain $\Omega_f(t)$:

$$\begin{aligned} \rho_f\Big(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\Big) &= -\nabla p + \nabla\cdot\boldsymbol{\sigma}_f, \\ \nabla\cdot\mathbf{u} &= 0, \end{aligned} \tag{2.1}$$

  with $\boldsymbol{\sigma}_f = \mu_f(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ the viscous stress tensor for a Newtonian fluid.

- The incompressible elastic material is governed by similar equations which derive from

the Navier equation in domain $\Omega_e(t)$:

$$\begin{aligned} \rho_e\Big(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\Big) &= -\nabla p + \nabla\cdot\boldsymbol{\sigma}_e, \\ \nabla\cdot\mathbf{u} &= 0, \end{aligned} \tag{2.2}$$

with $\boldsymbol{\sigma}_e$ the elastic stress tensor which will be defined later.

- Both velocity and normal components of the stress tensor are continuous across the interface $\Gamma_e(t)$, *i.e.* the jumps $[\cdot]_{\Gamma_e}$ across the interface are equal to zero:

$$\begin{aligned} [\mathbf{u}]_{\Gamma_e} &= 0, \\ [\sigma]_{\Gamma_e}\cdot\mathbf{n}_e = \sigma_f\cdot\mathbf{n}_e - \sigma_e\cdot\mathbf{n}_e &= 0, \end{aligned} \tag{2.3}$$

  where $\mathbf{n}_e$ is the normal vector to the interface $\Gamma_e$, arbitrarily pointing towards the fluid.

- Finally, the effect of rigid bodies is taken through no-slip boundary conditions on $\Gamma_s(t)$:

$$\mathbf{u} = \mathbf{u}_s. \tag{2.4}$$

  Note that several rigid bodies can be considered and we have thus $\mathbf{u} = \mathbf{u}_s^{(i)}$ on $\Gamma_s^{(i)}$ for the $i^{th}$ body. In this study, the rigid body velocity on $\Gamma_s$ will be imposed, and is thus defined as an external data.

One difficulty in the numerical resolution of system (2.1)-(2.4) is the imposition of boundary conditions (2.3) and (2.4). One way would be to use a computational mesh that fits the interfaces $\Gamma_e$ and $\Gamma_s$. The main advantage of this approach is that conditions (2.3) and (2.4) can be imposed in an accurate way on the interface-fitted mesh nodes. The main drawback, however, is that lot of remeshing steps has to be performed to follow the interface deformation, especially for soft materials inducing large deformations.

The approach considered in this study is to employ non body-fitted grids. In this formulation, the interfaces are represented in an implicit way using level-set functions. The main advantage is that no remeshing steps are needed, but special treatments have to be done since no grid points are located on the interfaces.

## 2.1 Fluid-elastic structure coupling

Equations (2.1) and (2.2) differ only with material quantities such as density $\rho$ and stress tensor $\boldsymbol{\sigma}$. Let $\chi_e$ be the mask function such that $\chi_e(\mathbf{x}) = 1$ if $\mathbf{x}\in\Omega_e$ and $\chi_e(\mathbf{x}) = 0$ elsewhere. A multi-material quantity $\eta$ can generally be written as $\eta = (1-\chi_e)\eta_f + \chi_e\eta_e$. Thus, we have the following mixed quantities:

$$\begin{aligned} \rho &= (1-\chi_e)\rho_f + \chi_e\rho_e, \\ \boldsymbol{\sigma} &= (1-\chi_e)\boldsymbol{\sigma}_f + \chi_e\boldsymbol{\sigma}_e. \end{aligned} \tag{2.5}$$

According to this monolithic formulation (2.5), Equations (2.1) and (2.2) can be recasted as a single set of equations on the domain $\Omega_f\cup\Omega_e$:

$$\begin{aligned} \rho\Big(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\Big) &= -\nabla p + \nabla\cdot\boldsymbol{\sigma}, \\ \nabla\cdot\mathbf{u} &= 0. \end{aligned} \tag{2.6}$$

This system is then completed with conditions (2.3). These conditions could be imposed in a sharp way by modifying the stencil and algorithms near the interface. However, our goal is

to keep as far as possible the same stencils and discretizations inside the whole computational domain. In the diffuse-interface approach we choose, the interface is then smoothed by regularizing the mask function $\chi_e$. While decreasing the accuracy at the interface, this approach gives rise to a continuous problem across the interface, and thus conditions (2.3) are naturally satisfied. The regularization of the mask function is performed using a regular level-set function. Here we chose the signed distance function $\phi$ arbitrarily defined to be negative in $\Omega_e$. The mask function is hence smoothed on a narrowband of width $2\delta$:

$$\chi_e(\phi) = \begin{cases} 1 & \text{if } \phi < -\delta \\ 0 & \text{if } \phi > +\delta \\ 1 - \frac{1}{2}\big(1 + \frac{\phi}{\delta} + \frac{1}{\pi}\sin(\pi\frac{\phi}{\delta})\big) & \text{else.} \end{cases} \tag{2.7}$$

The width of the narrowband is usually limited to very few grid points, and we will choose $\delta = 3h_{min}$, $h_{min}$ being the minimum characteristic size of the grid cells located near the interface $\Gamma_e$.
The level-set function $\phi$ is transported using the following equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \tag{2.8}$$

Since the velocity is solenoidal in the whole domain $\Omega_f \cup \Omega_e$, we can use the conservative form of Equation (2.8):

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0. \tag{2.9}$$

The regularization (2.7) is based on Euclidian distance informations. Since we have no guarantees that the level-set function $\phi$ is still a sign distance function after applying (2.9), reinitialization iterations have to be used (see [39]). Extension to several elastic bodies is straightforward introducing multiple level-set functions.

## 2.2 Interactions with rigid bodies

The previous section have described a numerical modeling of system (2.1)-(2.3). These equations have been recasted in a simpler set of equations, based on the monolithic formulation (2.6). The last condition to close system (2.1)-(2.3) is thus the interaction of fluid and elastic materials with rigid bodies (2.4).

As previously done for the elastic structure, we aim to avoid meshing and remeshing the whole computational domain when the rigid bodies move across this domain. The rigid bodies will be then represented in an implicit way using a mask function. In this study, we chose the Volume Penalization originally introduced by Angot et al. [2], or more precisely its second order extension proposed in [5]. This method has been validated in several previous studies [4, 6].

The main idea of this approach is to consider the whole system as a porous medium, with a variable permeability $\varepsilon$. In that regard, a rigid structure has a very low permeability, $\varepsilon \ll 1$. The system (2.1)-(2.3) can be solved adding extra penalization terms to the momentum equation in (2.6). We obtain the following equations in the whole domain $\Omega$, written the conservation form:

$$\begin{gathered} \rho\Big(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u})\Big) = -\nabla p + \nabla \cdot \sigma + \frac{\chi_s}{\varepsilon}(\mathbf{u}_s - \mathbf{u}), \\ \nabla \cdot \mathbf{u} = 0. \end{gathered} \tag{2.10}$$

where the mask function $\chi_s$ related to the rigid solid is

$$\chi_s(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_s, \\ 0 & \text{else.} \end{cases}$$

In case of multiple bodies, the last term of equation (2.10) is $\sum_{i=1}^{N_b} \frac{\chi_s^{(i)}}{\varepsilon}(\mathbf{u}_s^{(i)} - \mathbf{u})$ with $N_b$ bodies. The solution of the system (2.10) converges towards the solution of the original decoupled system (see [2]) as $\sqrt{\varepsilon}$ tends to zero [20]. In this study we set $\varepsilon = 10^{-10}h^2$, where $h$ refers to a characteristic length of the mesh which will be introduced later.

## 2.3 Definition of the elastic stress tensor

The last point is the definition of the elastic stress tensor $\boldsymbol{\sigma}_e$. Let $\Omega_0$ be the initial or reference configuration of $\Omega_e$ at time $t = 0$ and $\Omega_t$ its deformed configuration at time $t$.

**Backward characteristics**

In a Lagrangian framework, the deformation of a structure is followed according to the reference configuration $\Omega_0$ using the direct (or forward) characteristics

$$\begin{aligned} \mathbf{X} : [0, T_{max}] \times \Omega_0 &\longrightarrow \Omega_t, \\ (t, \boldsymbol{\xi}) &\longmapsto \mathbf{X}(t, \boldsymbol{\xi}). \end{aligned}$$

In this context, at time $t$, a Lagrangian particle initially located at $\boldsymbol{\xi} = \mathbf{X}(0, \boldsymbol{\xi})$ has moved from $\boldsymbol{\xi}$ to $\mathbf{x} = \mathbf{X}(t, \boldsymbol{\xi})$ according to the velocity field. The deformation is hence described through the equation:

$$\frac{\partial \mathbf{X}}{\partial t}(t, \boldsymbol{\xi}) = \mathbf{u}(t, \mathbf{X}(t, \boldsymbol{\xi})), \qquad \mathbf{X}(t = 0, \boldsymbol{\xi}) = \boldsymbol{\xi}. \tag{2.11}$$

In an Eulerian framework, the continuous medium is described through the deformed configuration $\Omega_t$. To find the initial position $\xi \in \Omega_0$ of a particle $x \in \Omega_t$, we introduce the backward characteristics

$$\begin{aligned} \mathbf{Y} : [0, T_{max}] \times \Omega_t &\longrightarrow \Omega_0, \\ (t, \mathbf{x}) &\longmapsto \mathbf{Y}(t, \mathbf{x}). \end{aligned}$$

By definition, the backward characteristics are the inverse of the forward characteristics, *i.e.* $\mathbf{X} = \mathbf{Y}^{-1}$ (see Figure 2) since we have the relations:

$$\mathbf{X}(t, \mathbf{Y}(t, \mathbf{x})) = \mathbf{x} \quad \text{and} \quad \mathbf{Y}(t, \mathbf{X}(t, \boldsymbol{\xi})) = \boldsymbol{\xi}. \tag{2.12}$$

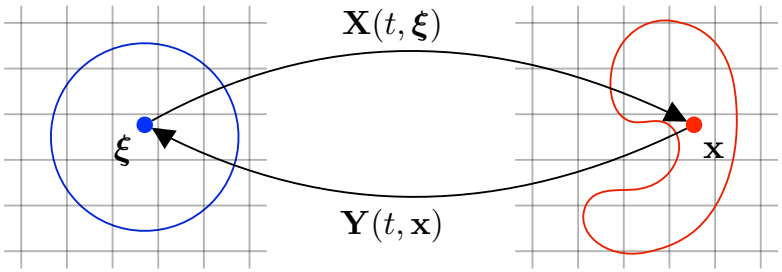



**Figure 2:** *Forward and backward characteristics are used to describe the deformation of a continuous medium.*

Differentiating the second equation of (2.12) with respect to time, we get a transport equation:

$$\begin{cases} \dfrac{\partial \mathbf{Y}}{\partial t}(t, \mathbf{x}) + \mathbf{u}(t, \mathbf{x}) \cdot \nabla_{\mathbf{x}} \mathbf{Y}(t, \mathbf{x}) = 0, \\ \mathbf{Y}(t = 0, \mathbf{x}) = \mathbf{x}. \end{cases} \tag{2.13}$$

This Eulerian relation is equivalent to the Lagrangian formulation (2.11). Moreover, differentiating one of the relations of (2.12) with respect to $\boldsymbol{\xi}$ or $\mathbf{x}$, the deformation tensor $\mathbf{F}$ can be expressed in the sense of both formalisms as

$$\mathbf{F} := [\nabla_{\boldsymbol{\xi}} \mathbf{X}] = [\nabla_{\mathbf{x}} \mathbf{Y}]^{-1}. \tag{2.14}$$

Under the assumption of incompressibility, we have the relation:

$$J := \det(\mathbf{F}) = 1. \tag{2.15}$$

In the Eulerian framework, the backward characteristics $\mathbf{Y}$ are considered to describe the deformation of an elastic material represented by $\Omega_e$. The evolution in time of $\mathbf{Y}$ is followed by solving the transport problem (2.13). The deformation tensor $\mathbf{F}$ can be computed as a function of $\mathbf{Y}$ using relation (2.14).

**Hyper-elastic model**

In the general case, linear elastic models are unable to predict correctly the behavior of materials which undergo from moderate to large deformations. In order to simulate properly the deformation of elastomeric membranes, it is appropriate to opt for a hyper-elastic model. This section is closely inspired by the theory of hyper-elasticity, as stated among others by Ogden [16, 27].

The properties of a hyper-elastic material are traditionally expressed in terms of a strain-energy function, called $W = W(\mathbf{F})$, depending on the deformation tensor. The internal energy per mass unit $\mathcal{E}$ related to the material is defined from this strain-energy function as:

$$\mathcal{E} = \int_{\Omega_0} W(\mathbf{F}(t, \boldsymbol{\xi}))\, d\boldsymbol{\xi}. \tag{2.16}$$

It can be shown that, under the assumptions of objectivity and isotropy, $\mathcal{E}$ can be written as a function of the invariants $\imath_{\mathbf{C}}$ of the right Cauchy-Green tensor $\mathbf{C} = \mathbf{F}^T\mathbf{F}$ (which are also the invariants $\imath_{\mathbf{B}}$ of the left Cauchy-Green tensor $\mathbf{B} = \mathbf{F}\mathbf{F}^T$):

$$\mathcal{E} = \int_{\Omega_0} \tilde{W}(\imath_{\mathbf{C}})\, d\boldsymbol{\xi} = \int_{\Omega_t} \tilde{W}(\imath_{\mathbf{B}})\ J^{-1}\, d\mathbf{x}, \tag{2.17}$$

where the invariants are $\imath_{\mathbf{C}} = \imath_{\mathbf{B}} := (I_1, I_2, I_3) = \big(\mathrm{Tr}(\mathbf{B}), \mathrm{Tr}(\mathrm{Cof}(\mathbf{B})), \det(\mathbf{B})\big)$. For an isotropic incompressible material, the internal energy depends only on the first two invariants, since $I_3 = J^2 = 1$ according to the incompressibility constraint (2.15). In this context, it is hence possible to prove that the elastic stress tensor $\boldsymbol{\sigma}_e$ can be expressed as:

$$\boldsymbol{\sigma}_e = -p^*\mathbf{I} + 2\frac{\partial W}{\partial I_1}\mathbf{B} - \frac{\partial W}{\partial I_2}\mathbf{B}^{-1}, \tag{2.18}$$

where $p^*$ refers to an arbitrary hydrostatic pressure. This Eulerian formulation shows that the elastic stress tensor depends on the left Cauchy-Green tensor $\mathbf{B} = [\nabla\mathbf{Y}]^{-1}[\nabla\mathbf{Y}]^{-T}$ and its invariants.

In this work, we consider the two-parameter Mooney-Rivlin model [24, 33], for which the strain-energy function is defined (in three dimensions) as:

$$W = c_1(I_1 - 3) + c_2(I_2 - 3), \tag{2.19}$$

where $c_1, c_2 \geq 0$ are the empirical parameters of the material related to the shear modulus through the consistancy relation $G = 2(c_1 + c_2)$. From Equations (2.18) and (2.19), the Mooney-Rivlin stress tensor is consequently:

$$\boldsymbol{\sigma}_e = -(2c_1 I_1 - 2c_2 I_2)\mathbf{I} + 2c_1\mathbf{B} - 2c_2\mathbf{B}^{-1}. \tag{2.20}$$

This kind of model is particularly adapted to rubber-like materials which undergo moderate deformations (200%-250%), as reported by [22]. Some more complete hyperelasticity models, such as the Ogden or van der Waals models, could also have been chosen, but they depend generally on more parameters and are thus more difficult to calibrate.

### 2.4 The Eulerian single-continuum model

This unified model has already been studied by [8], and used previously by [11]. We solve the whole fluid-structure system in a monolithic way, using a single-continuum model in the entire domain $\Omega$:

$$\rho\Big(\frac{\partial \mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{u}\otimes\mathbf{u})\Big) = -\nabla p + \nabla\cdot\boldsymbol{\sigma}(\mathbf{u},\mathbf{Y},\phi) + \frac{\chi_s}{\varepsilon}(\mathbf{u}_s - \mathbf{u}), \tag{2.21a}$$

$$\nabla\cdot\mathbf{u} = 0,$$

$$\begin{aligned} \frac{\partial \mathbf{Y}}{\partial t} + \nabla\cdot(\mathbf{u}\otimes\mathbf{Y}) &= 0, \\ \frac{\partial \phi}{\partial t} + \nabla\cdot(\mathbf{u}\phi) &= 0, \end{aligned} \tag{2.21b}$$

where the stress tensor is defined as:

$$\boldsymbol{\sigma}(\mathbf{u},\mathbf{Y},\phi) = \boldsymbol{\sigma}_f(\mathbf{u}) + \chi_e(\phi)\boldsymbol{\sigma}_e(\mathbf{Y}), \tag{2.22}$$

with $\boldsymbol{\sigma}_f = \mu_f(\nabla\mathbf{u}+\nabla\mathbf{u}^T)$ and $\boldsymbol{\sigma}_e = -(2c_1I_1 - 2c_2I_2)\mathbf{I} + 2c_1\mathbf{B} - 2c_2\mathbf{B}^{-1}$. To enhance the stability of the numerical method, both fluid and elastic material are chosen to be viscous. This system is closed using appropriate initial and boundary conditions for $\mathbf{u}$, $p$ $\mathbf{Y}$ and $\phi$.

A mathematical analysis of this Eulerian model has been proposed by Cottet et al. [8]. In this study, they proved notably the local existence of strong solutions, in the case of a membrane immersed in a viscous fluid.

## 3 Numerical method

This section is dedicated to the time and space discretizations of the monolithic model (2.21). This model has already been studied by Deborde et al. [11] in the context of fully uniform Cartesian meshes. The originality of this paper lies in the use of quadtree-based hierarchical Cartesian meshes. Indeed, this kind of mesh is particularly interesting due to the multi-physics nature of the problem.

In order to solve numerically the model (2.21), a versatile cell-center finite volume approach is preferred. This choice is motivated by two main arguments. On the one hand, in order to get an efficient and automatic management of the quadtree data structure, we use the Bitpit/PABLO library, developed by the company Optimad Engineering Srl (Torino). Based on a *Z-order* filling curve, the access to the data related to a grid cell (and its neighbors) is natural and cost-optimized. It also provides a transparent and automatic handling of the connectivity and intersections, which is necessary for the computations of numerical fluxes. On the other hand, a cell-centered finite difference scheme on quadtree grids has been proposed recently by Raeli et al. [29] for the resolution of anisotropic elliptic problems. In that work, a local study of the configuration of each stencil is performed to consistently discretize the differential operators. The finite-volume approach proposed in the present work does not need

a study of the local configuration of the stencil, at the price of being less accurate at the level jumps although the order of convergence is preserved.

## 3.1 Time integration

Let $\Delta t := t^{n+1} - t^n$ be the time step and $\varphi^n := \varphi(t^n)$ be the discrete value of a function $\varphi$ at time $t^n$. For sake of simplicity, the time step is assumed to be constant, but the generalization to a variable time step is straightforward. This time step is chosen in order to respect the stability conditions, relative to the chosen time discretization, such as the classical Courant-Lax-Friedrichs (CFL) condition induced by explicit convective schemes. It is based on the sum of the maximum convection speed $||\mathbf{u}||_{L^\infty}$ and the speed of linear elastic waves $c_{wave}$ propagating inside the structure. In the incompressible regime, this former speed satisfies the relation $c_{wave}^2 = E/\rho = 3G/\rho$ where $E$ and $G$ refer to the Young and shear modulus respectively. In our simulations, the CFL number is set to 0.25.

One of the main difficulty in solving system (2.21) lies in the numerical discretization of the elastic part of the stress tensor in (2.21a). Indeed, $\boldsymbol{\sigma}_e$ depends on $\mathbf{B} = [\nabla \mathbf{Y}]^{-1}[\nabla \mathbf{Y}]^{-T}$ and is thus nonlinear in $\mathbf{Y}$. The system (2.21) could be solved implicitely in a coupled way using Newton's iterations. An implicit formulation would be particularly valuable to get rid-off the time step restriction due to elastic materials. The drawback is however the large computational cost to build and solve the non-linear system. In this work, we chose an iterate approach to couple the fluid and the elastic material by solving alternatively in time Equations (2.21a) and (2.21b).

Provided that the velocity field $\mathbf{u}$ is solenoidal at time iteration $n$ (to be discussed in Section 3.2), we solve the transport equations (2.21b) as a first step. The level-set function $\phi$ is transported from time $t^n$ to $t^{n+1}$ using the classical second-order TVD Runge-Kutta scheme as:

$$\begin{aligned} \phi^* &= \phi^n - \Delta t \nabla \cdot (\mathbf{u}\phi)^n, \\ \phi^{n+1} &= \frac{1}{2}\phi^n + \frac{1}{2}\phi^* - \frac{\Delta t}{2}\nabla \cdot (\mathbf{u}\phi)^*. \end{aligned} \tag{3.1}$$

The same numerical scheme is used for the transport of the backward characteristics.

Then, system (2.21a) is solved using the fractional time step method introduced by Chorin [7] and Temam [40]. We first compute a velocity field $\mathbf{u}^*$ from a pressure guess $q$ with a second-order Gear scheme applied to the momentum equation:

$$\begin{aligned} \frac{3\mathbf{u}^* - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\Delta t} &+ \big(2\nabla \cdot (\mathbf{u} \otimes \mathbf{u})^n - \nabla \cdot (\mathbf{u} \otimes \mathbf{u})^{n-1}\big) \\ &= \frac{1}{\rho}\Big( - \nabla q + \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}^*, \mathbf{Y}^{n+1}, \phi^{n+1}) + \frac{\chi_s^{n+1}}{\epsilon}(\mathbf{u}_s^{n+1} - \mathbf{u}^*)\Big) \end{aligned} \tag{3.2}$$

Since the backward characteristics and the level-set function have already been advected in time, the quantities $\mathbf{Y}^{n+1}$ and $\phi^{n+1}$ are known and the elastic part of the stress tensor can be finally computed in an explicit way. Also, since $\mathbf{u}_s^{n+1}$ and thus $\chi_s^{n+1}$ are imposed, they can be considered as source terms. The only implicit terms related to $\mathbf{u}^*$ are in the viscous part of stress tensor and in the penalization term, resulting in a classical linear system to be solved.

The velocity field $\mathbf{u}^*$ is *a priori* not solenoidal. A projection onto the divergence-free subspace is hence performed. Applying the divergence operator to equation

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\big(\nabla p^{n+1} - \nabla q\big), \tag{3.3}$$

and imposing that $\nabla \cdot \mathbf{u}^{n+1} = 0$, we obtain a Poisson Equation:

$$\Delta p' = \nabla \cdot \mathbf{u}^*, \tag{3.4}$$

where $p' := (\Delta t/\rho)(p^{n+1} - q)$ is the pressure increment. Homogenous Neumann boundary conditions are imposed in order to ensure that there is no perturbation at the boundaries for the normal component of the velocity. Once the pressure increment $p'$ is computed, a correction step is performed:

$$\begin{cases} p^{n+1} = q + \dfrac{\rho}{\Delta t}\, p', & \text{(3.5a)} \\ \mathbf{u}^{n+1} = \mathbf{u}^* - \nabla p'. & \text{(3.5b)} \end{cases}$$

In this study, we use the incremental version of the projection method proposed by Goda [17] for which $q := p^n$.

## 3.2 Spatial discretizations

In this work, the monolithic model (2.21) is solved numerically using quadtree-based cell-centered finite volume schemes.

### 3.2.1 Quadtree

We use a Cartesian hierarchical data structure, called quadtree, to represent the spatial partitioning of the physical domain. As depicted in Figure 3, a quadtree grid is composed of square cells with different levels of refinement. Starting from the *root* of the tree (level 0), which represents the whole computational domain, the grid is recursively refined to a desired level of refinement. When refining, a parent cell of level $p$ is divided into four children of level $(p+1)$. Inversely, four children can merge into a single parent when coarsening. If a cell has no child, which means it is located at the treetop, the cell is called *leaf*.

Thanks to the library PABLO, which is a part of Bitpit library, we get use of an efficient tool to store the data structure. The library relies on the linear Z-order indexing proposed by Morton [25] in 1966, which enables the manipulation of the 2D grid data from a 1D vector structure. We can thus access all the informations coming from neighboring cells in a cost-optimized way, from the point of view of computational time and memory. Here, the quadtree grid is defined graded, which means that the level difference between a cell and all its adjacent cells (by face) is at most one; and linear, which means that only the data related to the leaves are kept in memory.

This Adaptative Mesh Refinement (AMR) procedure offers the opportunity to dynamically adapt the mesh to the flow configuration. By refining locally where the solution varies significantly (*e.g.* in the vicinity of interfaces and in the wake of structures), and by coarsening where the solution is more regular, the number of degrees of freedom can be drastically reduced while preserving a desired accuracy for our simulations. This strategy is even more interesting when it comes to taking into account structures that can move or be deformed. For the domain decomposition, the number of communications between processors is restricted to a single layer of ghost cells. If this constraint guarantees a high efficiency of the parallelism, the spatial discretizations involve compact stencils only and thus, this constraint limits *a fortiori* the order of accuracy of the numerical method.

In this section, we detail the finite volume discretizations of the differential operators involved in the multi-material model. To describe these discretizations, we get use of arbitrary

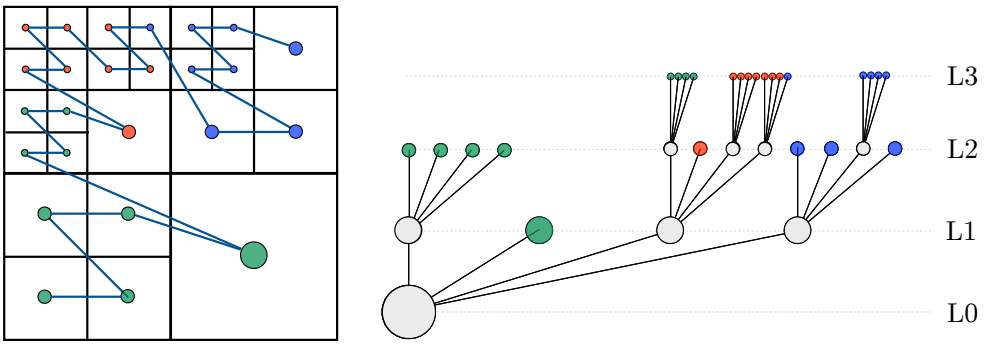



**Figure 3:** *Global Z-order indexing for a two-dimensional graded quadtree grid. The domain decomposition is depicted with colors.*

quantities, namely a scalar function $\varphi$ and a vector field $\mathbf{v} \in \mathbb{R}^2$. The square domain $\Omega$ is decomposed into a quadtree partition of $N_{cells}$ small square cells $\Omega_i$ of level $L_i$ (being the *leaves* of the tree) such that $\Omega = \bigcup_i \Omega_i$. By convention, the grid configuration is identified by its minimum and maximum levels of refinement $L_{min}$ and $L_{max}$. In other words, for a $L_{min} - L_{max}$ grid, the characteristic length $h_i$ of $\Omega_i$ is between $h_{max} = D/2^{L_{min}}$ and $h_{min} = D/2^{L_{max}}$, where $D$ refers to the characteristic size of the computational domain $\Omega$. We denote by $\mathbf{x}_i$ the center of the cell $\Omega_i$ and $\varphi_i := \varphi(\mathbf{x}_i)$ the discrete value of a quantity $\varphi$ evaluated at the cell center $\mathbf{x}_i$. Depending on the configuration, the area of the cell is $|\Omega_i| = h_i^2$ for a 2D simulation whereas $|\Omega_i| = 2\pi r_i h_i^2$ for a 2D axi-symmetric simulation, $r_i$ being the radial component of $\mathbf{x}_i$.

For a finite volume method, the discrete operators are computed as face contributions called fluxes. Let $f$ be the intersecting face of two cells called $\Omega_{out}$ and $\Omega_{in}$ and $\mathbf{x}_{fc}$ be the coordinates of the face center. As a convention, the normal vector $\mathbf{n}_f$ of $f$ is pointing from $\Omega_{in}$ to $\Omega_{out}$. The discrete values of $\varphi$ in $\Omega_{in}$ and $\Omega_{out}$ are denoted by $\varphi_{in}$ and $\varphi_{out}$ respectively. The area of the face is $|f| = h_f$ for a 2D simulation while $|f| = 2\pi r_{fc} h_f$ for a 2D axi-symmetric simulation, where $h_f = \min\{h_{in}, h_{out}\}$ refers to the characteristic length of $f$ and $r_{fc}$ is the radial component of the face center.

### 3.2.2 Cell-center and face-center velocities

It is well known that if the collocated cell-center velocity $\mathbf{u}^*$ is used to compute $\nabla \cdot \mathbf{u}^*$ in the Poisson equation (3.4), spurious grid-to-grid oscillations may occur due to the odd-even decoupling between velocity and pressure. This is one of the main drawbacks of non-staggered variable arrangements. As shown by Ferziger and Peric [15], traditional collocated methods cannot simultaneously guarantee pressure regularity and mass conservation. One way to overcome this problem has been proposed by Patankar [28] and consists in adopting a fully staggered arrangement of variables. For this Marker-And-Cell (MAC) method, the prediction step (3.2) and the Poisson equation (3.4) are solved at different locations, leading to distinct spatial discretizations. In this sense, staggered arrangements become more challenging for non-uniform Cartesian methods.

In order to stabilize the method, the collocated approach introduced by Rhie and Chow [31], and employed later by Mittal et al. [23], is chosen. An auxiliary face-center velocity, $U^*_{fc}$,

is introduced:

$$\begin{cases} \tilde{\mathbf{u}} = \mathbf{u}^* + \dfrac{\Delta t}{\rho}(\nabla p^n)_{cc} & \text{(3.6a)} \\ \widetilde{U_{fc}} = \mathcal{F}(\tilde{\mathbf{u}}) & \text{(3.6b)} \\ U^*_{fc} = \widetilde{U_{fc}} - \dfrac{\Delta t}{\rho}(\nabla p^n)_{fc} & \text{(3.6c)} \end{cases}$$

where subscripts $cc$ and $fc$ refer to cell-center and face-center locations respectively, and $\mathcal{F}$ denotes a cell-center to face-center operator. The computation of this operator $\mathcal{F}$ is discussed at the end of the section. This velocity arrangement is illustrated in Figure 4.

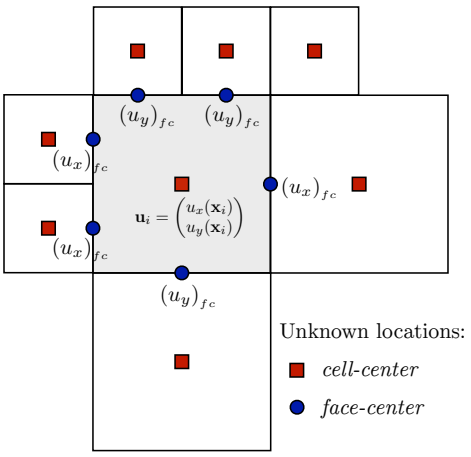



**Figure 4:** *Pattern of the velocity arrangement on quadtree grids.*

The face-center velocity $U^*_{fc}$ is now used to compute the right hand side of the Poisson equation (3.4):

$$\Delta p' = \frac{\rho}{\Delta t}\nabla \cdot U^*_{fc}. \tag{3.7}$$

Once the pressure increment $p'$ is obtained, both cell-center and face-center velocities are finally corrected independently:

$$\begin{cases} \mathbf{u}^{n+1} = \mathbf{u}^* - (\nabla p')_{cc}, & \text{(3.8a)} \\ U^{n+1}_{fc} = U^*_{fc} - (\nabla p')_{fc}. & \text{(3.8b)} \end{cases}$$

If the face-center velocity $U^{n+1}_{fc}$ satisfies the discrete mass conservation within the limit of the Poisson solver tolerence $\epsilon_\Delta$ (which can be set to machine precision), *i.e.*

$$\sum_{f \subset \partial\Omega_i} U^{n+1}_{fc}|f| = \mathcal{O}(\epsilon_\Delta) \ll 1, \quad \forall \Omega_i \subset \Omega,$$

the interpolation $\mathcal{F}(\mathbf{u}^{n+1})$ does not. Consequently, both cell-center $\mathbf{u}$ and face-center $U_{fc}$ velocities need to be stored to ensure mass conservation. However, only one component of the velocity is needed on each face, leading to a limited increase in the memory cost.

In order to compute a face centered quantity $\varphi_{fc}$ from cell-center values, a second order face-center interpolator $\mathcal{I}$ is introduced. Figure 5 illustrates the stencil used for this interpolation. First, if the face configuration is uniform, which means that the two cells $\Omega_{in}$ and $\Omega_{out}$ have a same level of refinement, $\varphi_{fc}$ is naturally defined as the average of $\varphi_{out}$ and $\varphi_{in}$. Otherwise, $\varphi_{fc}$

is computed as the average of $\varphi_T$ and $\varphi_B$ for a level jump configuration. Since these quantities are unknown, they are approximated using a 3-point linear or 4-point bilinear interpolation, depending on the number of neighboring cells located respectively around the lower (B) and upper (T) nodes of the face.

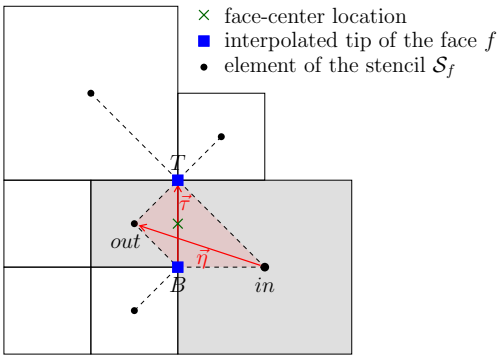


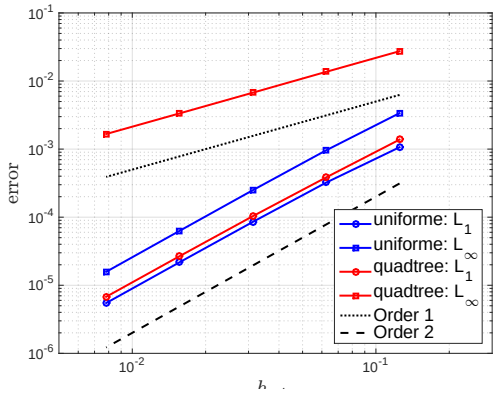

**Figure 5:** *Pattern of the face stencil considered to compute $\varphi_{fc}$ and $(\nabla\varphi)_{fc}$ for a level jump configuration. The red quadrilatera represents the diamond cell used to interpolate $(\nabla\varphi)_{fc}$.*

In the computation of the normal face-center velocity $U_{fc}$ (see Equation (3.6b)), a cell-center to face-center operator $\mathcal{F}$ was mentioned. This operator was firstly defined as the second order interpolator, *i.e.* $\mathcal{F} := \mathcal{I}$. But in practice, we have noticed the apparition of some spurious oscillations at level jumps within the limit of small time steps $\Delta t$. We also observed that, the smaller the time step, the larger the amplitude of these oscillations.

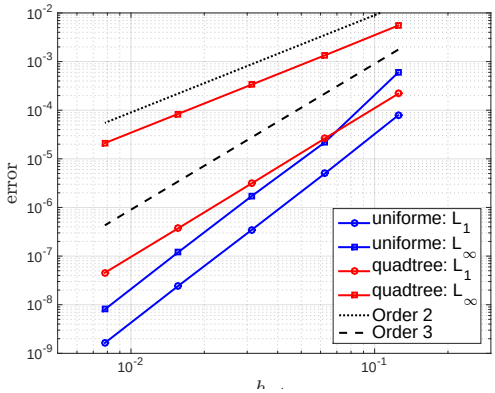
This oscillatory behavior is caused by the inaccurate computation of $\nabla \cdot U^*_{fc}$ at level jumps. Indeed, to construct a discretization scheme for the divergence operator, we have to choose between conservativity and accuracy. Since we expect the schemes to be conservative in this work, we lose *a fortiori* one order of accuracy at level jumps, as illustrated by the convergence study reported in Figure 6.

To limit the impact of these numerical inaccuracies, $\widetilde{U}_{fc}$ is computed as:

$$\widetilde{U}_{fc} = \mathcal{F}(\widetilde{\mathbf{u}}) := \begin{cases} \mathcal{I}(\tilde{\mathbf{u}}) & \text{if the stencils of the two cells are both uniforms,} \\ U^n_{fc} + \mathcal{I}(\tilde{\mathbf{u}} - \mathbf{u}^n) & \text{else.} \end{cases} \tag{3.9}$$




**(a)** *With a linear interpolation of the fluxes + a mid-point quadrature rule.*




**(b)** *With a quadratic interpolation of the fluxes + a two-point Gauss quadrature rule.*

**Figure 6:** *$L_1$ and $L_\infty$ truncation errors obtained for the divergence operator on uniform and quadtree meshes.*

### 3.2.3 Discretization of the involved operators

**Laplacian operator**

We use a simplified version of the Discrete Duality Finite Volume (DDFV) method proposed by [9, 12]. This method, called diamond method, is based on the choice of artificial diamond-like polygons, as depicted in Figure 5. The finite volume formulation of the Laplacian operator is discretized using a mid-point formula:

$$(\Delta\varphi)_i = \frac{1}{|\Omega_i|}\int_{\partial\Omega_i} \nabla\varphi\cdot\mathbf{n}\, ds = \sum_{f\subset\partial\Omega_i} (\nabla\varphi)_{fc}\cdot\mathbf{n}_f \frac{|f|}{|\Omega_i|} + \mathcal{O}(h_i^2). \tag{3.10}$$

The normal face-center derivative $(\nabla\varphi)_{fc}\cdot\mathbf{n}_f$ is computed as a linear combination of two directional derivatives:

$$\nabla\varphi\cdot\boldsymbol{\eta} = \varphi_{out} - \varphi_{in}, \tag{3.11a}$$
$$\nabla\varphi\cdot\boldsymbol{\tau} = \varphi_T - \varphi_B, \tag{3.11b}$$

following the direction of vectors $\boldsymbol{\eta} = \mathbf{x}_{out} - \mathbf{x}_{in}$ and $\boldsymbol{\tau} = \mathbf{x}_T - \mathbf{x}_B$ respectively. For a uniform face configuration, vectors $\boldsymbol{\eta}$ and $\mathbf{n}_f$ are collinear, Equation (3.11a) thus matches the classical second order centered approximation of the normal derivative. For a level jump configuration, the normal vector $\mathbf{n}_f$ can be expressed as a linear combination of vectors $\boldsymbol{\eta}$ and $\boldsymbol{\tau}$, *i.e.* $\exists c_\tau, c_\eta \in \mathbb{R}:\ \mathbf{n}_f = c_\tau\boldsymbol{\tau} + c_\eta\boldsymbol{\eta}$. The face-center normal derivative can be hence approximated as:

$$(\nabla\varphi)_{fc}\cdot\mathbf{n}_f = c_\tau(\varphi_T - \varphi_B) + c_\eta(\varphi_{out} - \varphi_{in}). \tag{3.12}$$

The unknown values $\varphi_T$ and $\varphi_B$ are interpolated from grid points data in a linear or bilinear sense, according to their respective number of neighboring cells.

The Poisson matrix resulting from this diamond discretization is not symmetric and positive definite. Guittet et al. [18] proposed a formulation that gives rise to a symmetric positive definite system. Numerical experiments show that the diamond discretization of Poisson solver is second-order accurate on both uniform and quadtree grids for different types of refinement, see Figure 7. The resolution of all linear systems involved in this method is performed using the PETSc library.

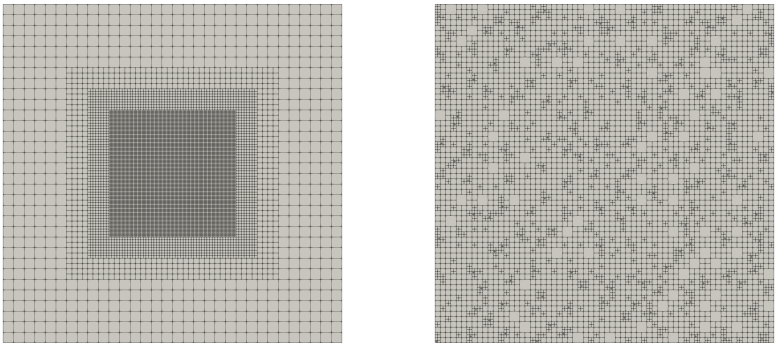
**Discretization of advection terms**

The transport equations for both the level-set function $\phi$ and the backward characteristics $\mathbf{Y}$ (see Equations (2.21b)) are solved using a second-order finite volume scheme. A similar scheme is also used to compute the convective terms during the prediction step (3.2). The conservative form of the transport equations is used since the mass conservation is precisely recovered after the projection step, *i.e.* $(\mathbf{u}\cdot\nabla)\varphi = \nabla\cdot\mathbf{F}(\varphi)$. The second-order finite-volume discretization reads:

$$\int_{\Omega_i} \nabla\cdot\mathbf{F}(\varphi)\, d\mathbf{x} = \int_{\partial\Omega_i} \mathbf{F}(\varphi)\cdot\mathbf{n}\, ds = \sum_{f\subset\partial\Omega_i} \mathbf{F}(\varphi_{fc})\cdot\mathbf{n}_f\ |f|. \tag{3.13}$$

where the flux is defined by $\mathbf{F}(\varphi) := U\varphi$. Since the normal face-center velocity $U_{fc}$ is directly computed from equation (3.8b), it is only required to reconstruct the function $\varphi$ to approximate the numerical flux.

In this paper, we use a second-order extension of the Godunov upwind scheme. The accuracy is limited to the second order since the quadrature rule, related to the velocity arrangement



| grid | uniform | | | | by-boxes AMR | | | | pseudo-random AMR | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $L_{min}$ | $L^1$ | $q_1$ | $L^\infty$ | $q_\infty$ | $L^1$ | $q_1$ | $L^\infty$ | $q_\infty$ | $L^1$ | $q_1$ | $L^\infty$ | $q_\infty$ |
| 5 | 4.96e-3 | / | 1.24e-2 | / | 1.48e-3 | / | 1.45e-2 | / | 7.62e-4 | / | 1.38e-2 | / |
| 6 | 1.17e-3 | 2.08 | 3.12e-3 | 1.99 | 4.05e-4 | 1.87 | 3.48e-3 | 2.06 | 1.83e-4 | 2.06 | 3.44e-3 | 2.00 |
| 7 | 2.85e-4 | 2.04 | 7.88e-4 | 1.99 | 9.52e-5 | 2.09 | 8.56e-4 | 2.02 | 4.53e-5 | 2.01 | 8.53e-4 | 2.01 |
| 8 | 7.01e-5 | 2.02 | 1.98e-4 | 1.99 | 2.29e-5 | 2.06 | 2.12e-4 | 2.01 | 1.11e-5 | 2.03 | 2.15e-4 | 1.99 |
| 9 | 1.74e-5 | 2.01 | 4.96e-5 | 2.00 | 5.77e-6 | 1.99 | 5.27e-5 | 2.01 | 2.79e-6 | 1.99 | 5.47e-5 | 1.97 |

**Figure 7:** *Convergence of the Poisson solver on uniform and quadtree meshes. The Poisson equation $\Delta\varphi = f$ is solved in the domain $\Omega = [0, 2\pi]^2$. The analytical solution is $\varphi(x, y) = \cos(x + y)\sin(x - y)$ and Dirichlet boundary conditions are prescribed on $\partial\Omega$. Two kinds of AMR have been tested, which rely on a refinement by boxes (on the left) and a pseudo-random refinement (on the right). Results have been obtained for a same minimum level of refinement $L_{min}$ (with $L_{max} = L_{min} + 3$) and show that the Poisson solver is second order accurate.*

with $U_{fc}$, involved in the discretization is only second order accurate. To that end, the linear reconstruction of $\varphi$ in each cell $\Omega_i$ is performed using the cell-center gradient as:

$$\varphi\big|_{\Omega_i}(\mathbf{x}) = \varphi_i + (\mathbf{x} - \mathbf{x}_i)^T(\nabla\varphi)_i, \tag{3.14}$$

for any point $\mathbf{x} \in \Omega_i$. The value of the gradient $(\nabla\varphi)_i$ is approximated using a quadratic least-square interpolation in the compact stencil of $\Omega_i$. This interpolation is always possible since the size of this stencil is at least 6 for a graded quadtree grid. To guarantee that the advection scheme is stable under the CFL condition formulated in Section 3.1, the computation of the flux is performed using the Rusanov (or Local Lax-Friedrichs) numerical flux, which reads:

$$\widetilde{\mathbf{F}}_{O2}(\varphi^+, \varphi^-) = \frac{1}{2}U_{fc}(\varphi^+ + \varphi^-) - \frac{1}{2}|U_{fc}|(\varphi^+ - \varphi^-), \tag{3.15}$$

where $\varphi^- = \varphi\big|_{\Omega_{in}}(\mathbf{x}_{fc})$ and $\varphi^+ = \varphi\big|_{\Omega_{out}}(\mathbf{x}_{fc})$ refer to the reconstruction of $\varphi_{fc}$ from both sides of the face $f$.

The discretization scheme outline above, called *FV2*, is not monotone. However, since the solution remains regular enough in our incompressible framework, *a priori* or *a posteriori* limiting techniques, such as WENO or MOOD, are not necessary.

## 3.3 Re-initialization and extrapolation algorithms

In this study, the level-set function $\phi$ is used to both capture the multi-material interface $\phi = 0$ and to define the regularized mask function $\chi_e \equiv \chi_e(\phi)$ involved in (2.22) and introduced in the previous sections. A common and natural choice for the level-set function is the signed distance function. The signed distance behaviour can however be lost when the level-set function is transported using the fluid velocity in the whole domain. Severe stretching of the distance

function near the interface can lead to numerical inaccuracies on the interface position. When dealing with level-set based methods, the mass of each phase cannot be properly conserved, which is a key requirement to get a realistic simulation of multimaterial problems. This loss of mass cannot be fully removed but it can be significantly reduced. A wide range of strategies, called reinitialization algorithms, have been proposed in the literature to recover the distance behaviour. First of all, we can mention the relaxation method proposed by Russo and Smereka [34] and its extensions to high-order precision by [13, 21] among others. A high-order reinitialization scheme would be computationally challenging in this work since we are limited to compact stencils for our discretization schemes. In addition, fast-sweeping methods [43, 46] are not suitable either because the Z-order indexing does not provide an efficient and low-cost sweeping direction to solve the Eikonal equation. The fast-marching methods introduced by Sethian [35, 36] remain more interesting in our case. A third order accurate fast-marching method (see for instance [1]) is employed to limit this variation of mass.

In addition, we can not guarantee the regularity of the backward characteristics $\mathbf{Y}$ near the interface. Indeed, the fluid can be subjected to large shear deformations, which can cause the explosive growth of some components of the mixed stress tensor $\boldsymbol{\sigma}$. These distortions of the backward characteristics can lead to numerical instabilities, especially when dealing with very stiff materials. In the approach of Sugiyama et al. [38], the irregularity of the stress tensor is eliminated in a more direct way by enforcing $\boldsymbol{\sigma}_e = \mathbf{0}$ in the fluid where $\chi_e = 0$. This extrapolation technique is hence not needed but this approach leads to other drawbacks, in particular, the equations to be solved to follow the evolution of the left Cauchy-Green tensor $\mathbf{B}$ are more complex.

Since we want to keep as far as possible the same stencils and discretization schemes in the whole computational domain, the characteristics are frequently reset inside the fluid using the linear extrapolation introduced by Aslam [3] and employed recently by Deborde et al. [11]. The two transport equations are solved using a semi-Lagrangian scheme based on (bi-)linear upwind interpolations.

## 3.4 Low-dispersion finite volume method

Although the upwind discretization scheme *FV2* presented previously is consistent and stable, it turns out to be a source of numerical instabilities when solving the FSI model. In practice, the numerical solution tends to be oscillatory, especially in the vicinity of the interface. These oscillations are critical for highly stiff materials since the computation can become unstable after some time iterations. However, the amplitude of these oscillations remains consistent in the sense that it decreases with a more refined mesh. To give some explanations of this phenomenon, a stability analysis in one dimension of space (1D) has been performed.

To that end, the 2D FSI model is simplified into a system of two nonlinear PDEs:

$$\partial_t u = \mathcal{F}(u, Y) := -u\partial_x u + \partial_x \sigma_e(Y), \tag{3.16a}$$

$$\partial_t Y = \mathcal{G}(u, Y) := -u\partial_x Y, \tag{3.16b}$$

where the unknowns are the velocity $u$ and the backward characteristic $Y$. This simplified model is chosen since it best inherits the characteristics of the 2D FSI model. Without loss of generality, the neo-hookean model is chosen, the elastic stress tensor can hence be written as:

$$\sigma_e(Y) = G[(\partial_x Y)^{-2} - 1]. \tag{3.17}$$

Here, the computational domain is purely elastic, *i.e.* $\Omega = \Omega_e$, the level-set function $\phi$ is thus not taken into account.

### 3.4.1 Numerical test in 1D

A preliminary one-dimensional test is performed. The computational domain $\Omega = [-0.03, 0.03]$ is uniformly discretized into $N = 200$ cells of size $\Delta x$. As depicted in Figure 8a, a discontinuous Heaviside function is prescribed as an initial condition:

$$u(t=0,x) = \begin{cases} 1 & \text{if } |x| < 0.003, \\ 0 & \text{else,} \end{cases} \tag{3.18}$$

to enable the apparition of numerical instabilities. The shear modulus is set to $G = 1$ MPa to simulate a stiff material. The 1D model (3.16) is integration in time using a first-order linear multi-step method:

$$\begin{aligned} \frac{Y_i^{n+1} - Y_i^n}{\Delta t} &= \mathcal{G}_i(u^{n+1}, Y^n), \quad \text{with } u^{n+1} = 2u^n - u^{n-1}, \\ \frac{u_i^{n+1} - u_i^n}{\Delta t} &= \mathcal{F}_i(u^n, Y^{n+1}). \end{aligned} \tag{3.19}$$

The advection operator $A(Y) = u\partial_x Y = \partial_x(uY) - Y\partial_x u$ is discretized in space using both finite volume and finite difference methods. A first test is performed using the analogous 1D version of the *FV2* scheme presented in Section 3.2.3, which can be written as (provided $u > 0$):

$$\left(\tilde{A}_{FV2}(Y)\right)_i = \frac{u_{i+\frac{1}{2}}}{4\Delta x}(Y_{i+1} - Y_{i-1}) - \frac{u_{i-\frac{1}{2}}}{4\Delta x}(-3Y_i + 4Y_{i-1} - Y_{i-2}). \tag{3.20}$$

As a comparison, a second test is proposed by considering a second order finite difference scheme, called *FD2*, which reads (provided $u > 0$):

$$\left(\tilde{A}_{FD2}(Y)\right)_i = \frac{u_i}{4\Delta x}(Y_{i+1} + 3Y_i - 5Y_{i-1} + Y_{i-2}). \tag{3.21}$$

For both numerical tests, the convective term $A(u)$ is discretized using the *FV2* scheme. The solutions obtained at time $t = 0.5$ ms are displayed in Figure 8b. The oscillatory behavior seems thus to be caused by the finite volume discretization scheme we employ to advect the backward characteristics. It is however possible to get rid of these instabilities with a finite difference method.

### 3.4.2 Asymptotic behavior of advection schemes

In what follows, the asymptotic convergence of the numerical schemes *FV2* and *FD2* in the limit of small perturbations is investigated. In this context, a scheme is called Linearity Preserving (LP) if the limit discrete scheme, obtained from linearization, is consistent with the linearized continuous operator.

As a linearization, the solution $(u, Y)$ is decomposed into the sum of a non-perturbed solution $(u_0, Y_0)$ and a small perturbation homogeneous to $(\widetilde{u}, \widetilde{Y})$ as:

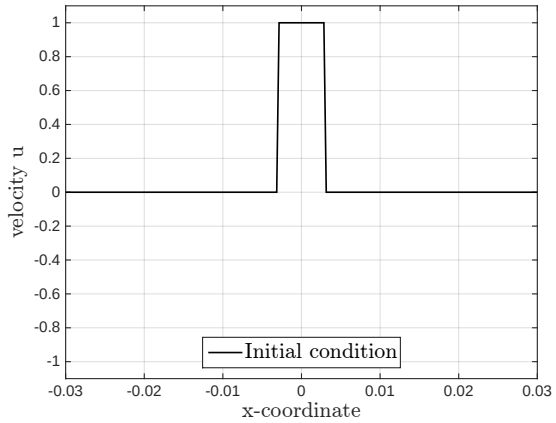
$$u(t,x) = u_0(t,x) + \varepsilon\widetilde{u}(t,x), \tag{3.22a}$$

$$Y(t,x) = Y_0(t,x) + \varepsilon\widetilde{Y}(t,x). \tag{3.22b}$$

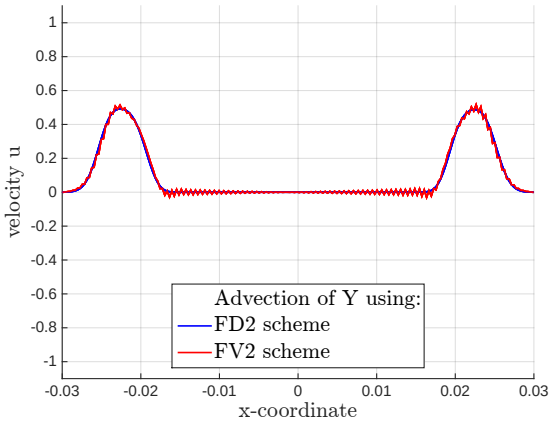
Since $Y(t=0,x) = x$, $Y$ can be interpreted as a perturbation of the identity by imposing $Y_0(t,x) := x$. Introducing the decomposition (3.22) into the 1D model (3.16), we have $u_0 \equiv 0$ and we obtain the following linearized model:

$$\partial_t \widetilde{u} = -2G\partial^2_{xx}\widetilde{Y}, \tag{3.23a}$$

$$\partial_t \widetilde{Y} = -\widetilde{u}. \tag{3.23b}$$




**(a)** *Initial condition* $u(t=0,\cdot)$.




**(b)** *Solution at time* $t = 0.5ms$.

**Figure 8:** *One-dimensional numerical test : initial condition and solutions obtained using FD2 and FV2 schemes as a discretization of the advective term* $A(Y)$.

Differentiating Equations (3.23) with respect to time, these equations can be recasted into wave equations for which the propagation speed is $a = \sqrt{2G}$.

We now compare the asymptotic behavior of the two advection schemes *FV2* and *FD2*. From the discrete formulations (3.20) and (3.21), the linearization leads to:

$$\textit{(FV2):} \quad (\partial_t \widetilde{Y})_i = -\widetilde{u}_i - \frac{\Delta x^2}{8}(\partial^2_{xx}\widetilde{u})_i + \mathcal{O}(\varepsilon), \tag{3.24a}$$

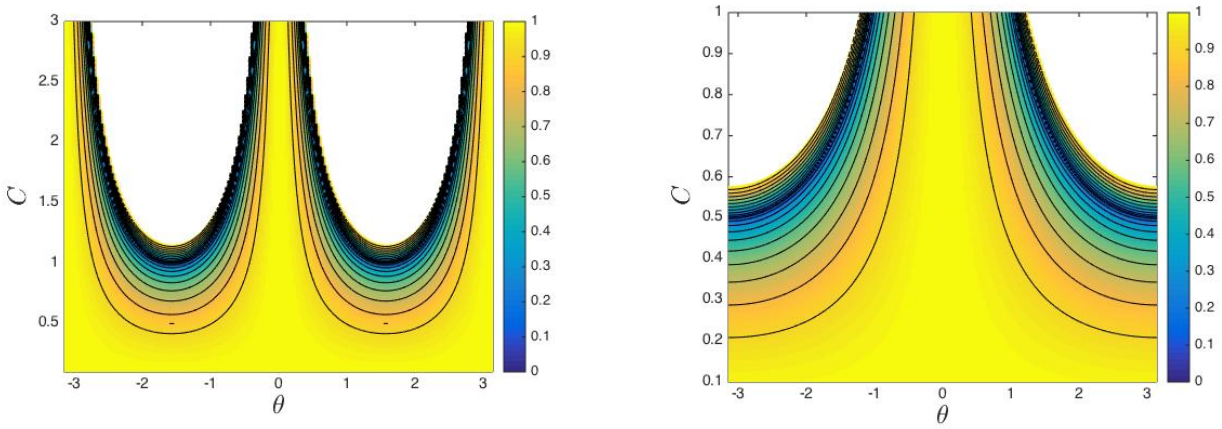
$$\textit{(FD2):} \quad (\partial_t \widetilde{Y})_i = -\widetilde{u}_i + \mathcal{O}(\varepsilon). \tag{3.24b}$$

On one hand, according to Equation (3.23b), we observe that the asymptotic behavior is preserved exactly using the finite difference scheme *FD2*, the scheme is then said to be Exactly Linearity Preserving (*ELP*). On the other hand, the *FV2* formulation is only second-order LP. Differentiating Equation (3.24a) with respect to time, we then obtain:

$$(\partial^2_{tt}\widetilde{Y})_i = a^2(\partial^2_{xx}\widetilde{Y})_i + \frac{a^2\Delta x^2}{8}(\partial^4_{xxxx}\widetilde{Y})_i + \mathcal{O}(\varepsilon). \tag{3.25}$$

In the limit of small perturbations $\varepsilon \to 0$, a dispersive term is consequently introduced by the finite volume formulation. This term depends both on the rigidity of the material ($a^2 = 2G$) and the characteristic size of the mesh $\Delta x$. This result explains partially the oscillatory phenomenon previously stated.

To complete these observations, a Fourier analysis is performed. For more details related to this analysis, the reader can refer to the Appendix A. The amplification factors obtained from both discretizations of the advective operator $A(Y)$ are presented in Figure 9. The high frequencies (for $\theta = \pm\pi$) are not damped using the finite volume discretization of the advective term $A(Y)$. This study confirms the oscillatory behavior of the solution (see Figure 8b). For stiff problems, some spurious instabilities can also appear computing the steep stress tensor $\chi_e \boldsymbol{\sigma}_e$. To stabilize the finite volume approach, it is appropriate to enforce the ELP property in order to enhance the damping of high frequencies, and therefore prevent the propagation of these oscillations.




**(a)** *Using the non-ELP FV2 scheme.*

**(b)** *Using the ELP FD2 scheme.*

**Figure 9:** *Asymptotic analysis of the model (3.16): amplification factors as a function of the frequency of a Fourier mode $\theta$ and the CFL number $C = a\Delta t/\Delta x$ obtained using ELP and non-ELP discretizations of the advective term $A(Y)$. Inside the white regions, the amplification factor is greater than 1.*

### 3.4.3 An *ELP FV2* scheme.

The robustness of the numerical method can be enhanced identifying the difference between the discretizations of *FD2* and *FV2*. By comparing their truncation errors, it is possible to highlight the dispersive term involved in Equation (3.25).

Under the assumption $u > 0$, the two discretization schemes for the advective operator are defined by Equations (3.20) and (3.21). The truncation errors $e_{FD2}$ and $e_{FV2}$, obtained from *FD2* and *FV2* schemes, can be respectively written as:

$$\big(e_{FD2}\big)_i = \Big[ -\frac{1}{12} u_i (\partial^3_{xxx} Y)_i \Big] \Delta x^2 + \mathcal{O}(\Delta x^3), \tag{3.26a}$$

$$\big(e_{FV2}\big)_i = \Big[ -\frac{1}{12} u_i (\partial^3_{xxx} Y)_i + \frac{1}{8} (\partial^2_{xx} u)_i (\partial_x Y)_i \Big] \Delta x^2 + \mathcal{O}(\Delta x^3). \tag{3.26b}$$

We thus define a source term, called $S_{FV2}$, which corresponds to the difference between the two truncation errors, namely:

$$\big(S_{FV2}\big)_i = \big(e_{FV2}\big)_i - \big(e_{FD2}\big)_i + \mathcal{O}(\Delta x^3) = \frac{\Delta x^2}{8} (\partial^2_{xx} u)_i (\partial_x Y)_i. \tag{3.27}$$

Hence, to describe the time evolution of backward characteristics, an inhomogeneous advection equation is considered:

$$(\partial_t Y)_i + (\widetilde{A}_{FV2}(Y))_i = (S_{FV2})_i. \tag{3.28}$$

A consistent discretization of the source term (3.27) for the advection of backward characteristics help thus enhancing the robustness of the numerical method since the non-oscillatory property is recovered.

The generalization in two dimensions is straightforward in the context of Cartesian meshes. As carried out in one dimension, the truncation errors are computed for both *FD2* and *FV2* schemes. We consider an uniform Cartesian mesh where the characteristic length of each cell $\Omega_{ij}$ is $h$. Under the assumption $u > 0$, the second order *FD2* and *FV2* schemes in two dimensions

can be respectively written as:

$$\begin{aligned}\big(\mathbf{u}\cdot\nabla Y\big)_{i,j} \approx \big(\tilde{A}^{2D}_{FD2}(Y)\big)_{i,j} = \frac{(\mathbf{u}_x)_{i,j}}{4h}(Y_{i+1,j}+3Y_{i,j}-5Y_{i-1,j}+Y_{i-2,j}) \\ + \frac{(\mathbf{u}_y)_{i,j}}{4h}(Y_{i,j+1}+3Y_{i,j}-5Y_{i,j-1}+Y_{i,j-2}),\end{aligned} \tag{3.29}$$

and:

$$\begin{aligned}\big(\nabla\cdot(\mathbf{u}Y)\big)_{i,j} &\approx \big(\tilde{A}^{2D}_{FV2}(Y)\big)_{i,j} \\ &= \frac{1}{h}\Big[(\mathbf{u}_x)_{i+\frac{1}{2},j}\big(Y_{i,j}+\frac{1}{4}(Y_{i+1,j}-Y_{i-1,j})\big) - (\mathbf{u}_x)_{i-\frac{1}{2},j}\big(Y_{i-1,j}+\frac{1}{4}(Y_{i,j}-Y_{i-2,j})\big) \\ &+ (\mathbf{u}_y)_{i,j+\frac{1}{2}}\big(Y_{i,j}+\frac{1}{4}(Y_{i,j+1}-Y_{i,j-1})\big) - (\mathbf{u}_y)_{i,j-\frac{1}{2}}\big(Y_{i,j-1}+\frac{1}{4}(Y_{i,j}-Y_{i,j-2})\big)\Big],\end{aligned} \tag{3.30}$$

for each component of the backward characteristics denoted by $Y$ ($Y=\mathbf{Y}_x$ or $Y=\mathbf{Y}_y$). The truncation errors $e^{2D}_{FD2}$ and $e^{2D}_{FV2}$, obtained from Equations (3.29) and (3.30), are:

$$\big(e^{2D}_{FD2}\big)_{i,j} = -\frac{h^2}{12}\Big((\mathbf{u}_x)_{i,j}(\partial^3_{xxx}Y)_{i,j}+(\mathbf{u}_y)_{i,j}(\partial^3_{yyy}Y)_{i,j}\Big)+\mathcal{O}(h^3), \tag{3.31}$$

and:

$$\begin{aligned}\big(e^{2D}_{FV2}\big)_{i,j} = \big(e^{2D}_{FD2}\big)_{i,j} + \underbrace{\frac{h^2}{8}\Big((\partial^2_{xx}\mathbf{u}_x)_{i,j}(\partial_x Y)_{i,j}+(\partial^2_{yy}\mathbf{u}_y)_{i,j}(\partial_y Y)_{i,j}\Big)}_{(S_1)_{i,j}} \\ + \underbrace{\frac{h^2}{24}Y_{i,j}\Big((\partial^3_{xxx}\mathbf{u}_x)_{i,j}+(\partial^3_{yyy}\mathbf{u}_y)_{i,j}\Big)}_{(S_2)_{i,j}}+\mathcal{O}(h^3),\end{aligned} \tag{3.32}$$

The truncation error $e^{2D}_{FD2}$ and the term $S_1$ correspond to the two-dimensional extension of truncation errors obtained previously in 1D (see Equations (3.26) and (3.27)). It can be proved that the last term $S_2$ of Equation (3.32) is not responsible for the oscillatory behavior of the solution. Since the term $S_1$ is the origin of the dispersive property, the evolution of the backward characteristics $\mathbf{Y}$ is carried out by means of the non-homogeneous advection equation

$$\frac{\partial\mathbf{Y}}{\partial t}+\nabla\cdot(\mathbf{u}\otimes\mathbf{Y}) = \mathbf{S}^{2D}_{FV2}, \tag{3.33}$$

where $\mathbf{S}_{FV2}$ refers to a source term defined by:

$$\mathbf{S}^{2D}_{FV2} = \frac{h^2}{8}(\partial^2_{xx}\mathbf{u}_x,\partial^2_{yy}\mathbf{u}_y)^T[\nabla\mathbf{Y}] =: \frac{h^2}{8}(\nabla\cdot[\nabla^{diag}\mathbf{u}])[\nabla\mathbf{Y}]. \tag{3.34}$$

Here, $[\nabla^{diag}\mathbf{u}]$ correspond to the diagonal matrix obtained from the velocity gradient $[\nabla\mathbf{u}]$, *i.e.* $[\nabla^{diag}\mathbf{u}]_{ij} := [\nabla\mathbf{u}]_{ij}\delta_{ij}$. By adding this source term $\mathbf{S}^{2D}_{FV2}$, the finite volume formulation becomes ELP and the numerical method hence benefits from the non-oscillation property.

# 4 Numerical Validation and Simulations

In what follows, we consider three test cases. First, the numerical method is validated over a reference test case on uniform grids for soft and stiff materials. In this configuration, the structure is driven by the fluid inside a cavity. The soft structure undergoes large deformation due to wall effects. In a second test case, we compare the results obtained on uniform and

AMR grids, prescribing an identical mesh resolution close to the interface. This test involves the interaction between a rigid body immersed in an elastic structure. Finally, a realistic biomedical simulation is proposed. In a 2D-axisymmetric complex geometry, mimicking a cardiac pump, a stiff elastomeric membrane is excited at the tip, at a given frequency, in order to reproduce a pulsatile blood flow.

## 4.1 A solid deformation in a lid-driven cavity flow

We first study the deformation of an elastic cylinder immersed in a lid-driven cavity flow. This test has been studied in the literature using both Lagrangian and Eulerian formalims [10, 38, 47]. An elastic cylinder is immersed in a cavity $\Omega = [0,1]^2$m filled with a fluid of density $\rho_f = 1$ kg.m$^{-3}$ and viscosity $\mu_f = 10^{-2}$ Pa.s. Initially, the center of mass of the cylinder is $\mathbf{x}_c(t=0) = (0.6, 0.5)^T$m and its diameter is set to 0.4m. The set-up is shown in Figure 10. The properties of the elastic structure and the fluid are identical, *i.e.* $\rho_e = \rho_f$ and $\mu_e = \mu_f$. The deformation of the neo-Hookean material ($c_1 = G/2$, $c_2 = 0$) is tested for two different shear modulus in order to consider both soft ($G = 0.1$ Pa) and stiff ($G = 10$ Pa) materials.

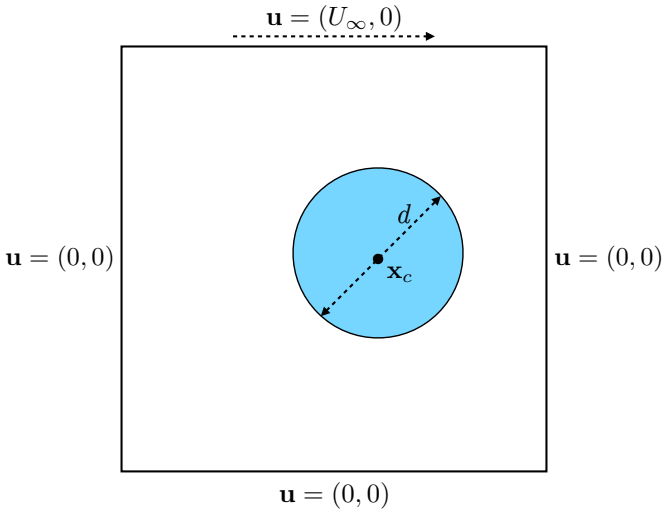



**Figure 10:** *Deformable cylinder in a lid-driven cavity flow: fluid-structure set-up.*

At time $t = 0$, the fluid and structure are at rest. The flow is then generated by imposing a horizontal velocity $\mathbf{u} = (1,0)^T$m on the upper wall while imposing no-slip boundary conditions on the other external boundaries. Numerical simulations are performed here using uniform grids. The level-set function $\phi$ and the backward characteristics $\mathbf{Y}$ are updated every 0.01s, up to a maximum distance of $d_{max} = 0.2$m.

The y-component of the velocity and z-component of the vorticity at time $t = 4.69$s are presented in Figure 11 for the soft material. The interface deformation, for both soft and stiff materials, is presented in Figure 12. The interface evolution is in good agreements with reference results obtained by Zhao et al. [47].

Figure 13 shows the temporal evolution of the centroid $\mathbf{x}_c(t)$ of the material. The location of the centroid is approximated as:

$$\mathbf{x}_c(t) \approx \frac{\sum_{i=1}^{N_{cells}} \widetilde{\Omega_i^e}(t)\ \mathbf{x}_i}{\sum_{i=1}^{N_{cells}} \widetilde{\Omega_i^e}(t)}, \tag{4.1}$$

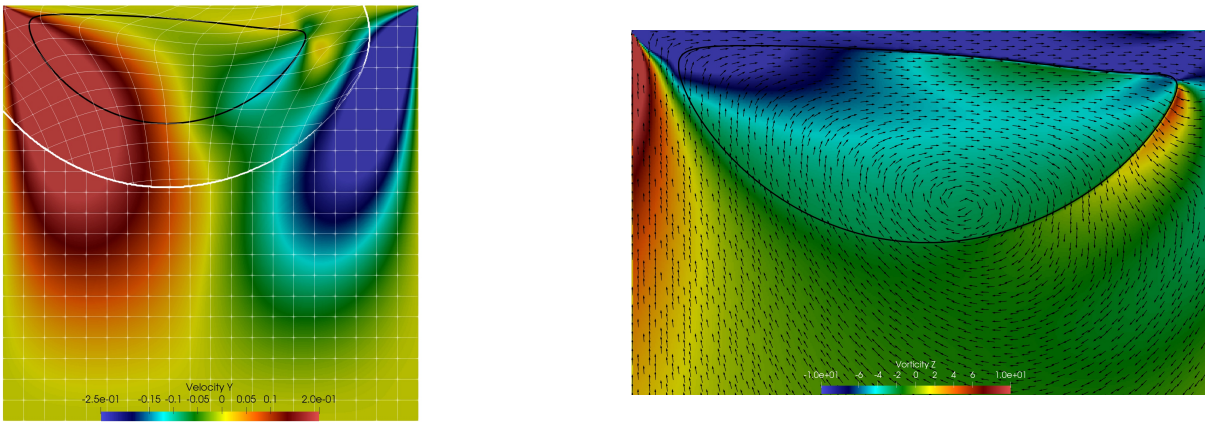



**Figure 11:** *Deformable soft cylinder ($G = 0.1Pa$) in a lid-driven cavity flow: snapshots at time $t = 4.69s$. Left: y-component of the velocity with* $\mathbf{Y}$*-contours in white; right: z-component of the vorticity with velocity field. The black line represents the interface* $\Gamma_e$*.*

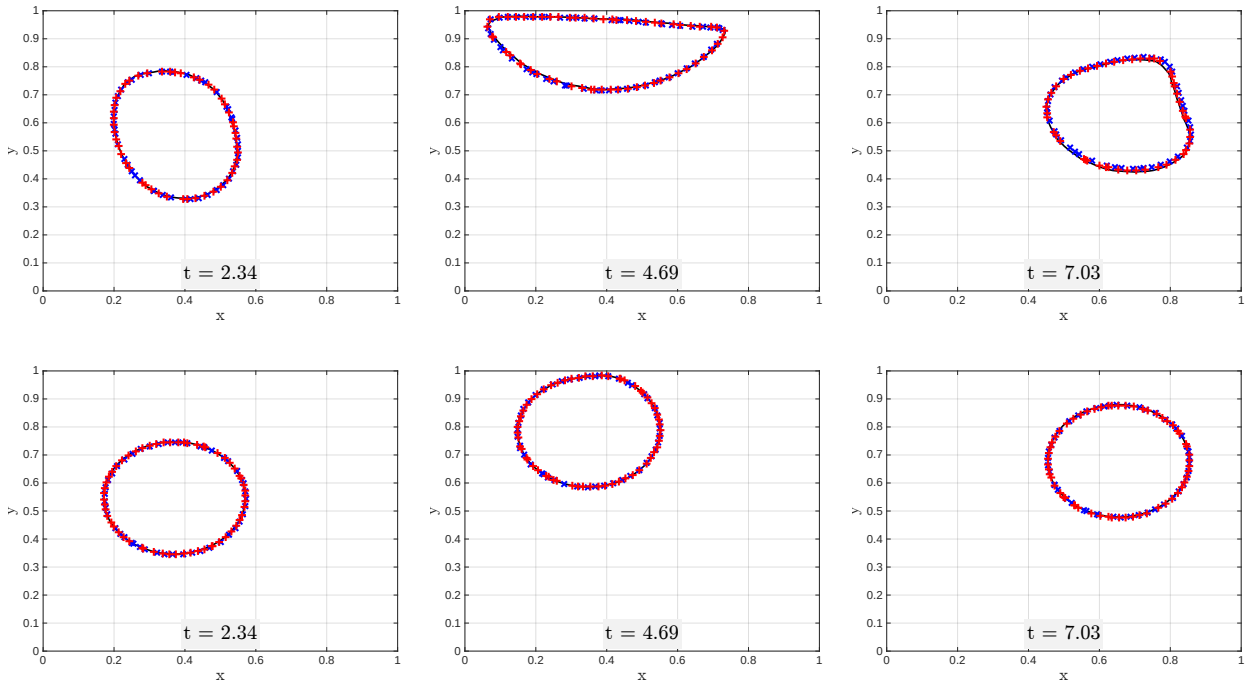



**Figure 12:** *Deformation of soft (top range) and stiff (bottom range) cylinders immersed in a lid-driven cavity flow: interface location* $\Gamma_e$ *for various time instants, comparison with the work of Zhao [47] (black line). The blue-plus and red-cross markers refer to the results obtained on uniform* $L_9$ *and* $L_{10}$ *grids respectively.*

where $\widetilde{\Omega_i^e}$ is an approximation of the volume fraction of solid in cell $\Omega_i$, *i.e.* $\widetilde{\Omega_i^e} = \chi_e(\phi_i)|\Omega_i|$. The results obtained on uniform grids are in good agreement with the results of Sugiyama et al. [38] and Deborde [10].

Finally, Figure 14 presents the temporal evolution of the material volume

$$|\Omega_e(t)| \approx \sum_{i=1}^{N_{cells}} \widetilde{\Omega_i^e}(t). \tag{4.2}$$

These computation are performed to quantify the mass-conservation error usually observed [37] when dealing with level-set methods. For both rigidities, the relative loss of volume remains

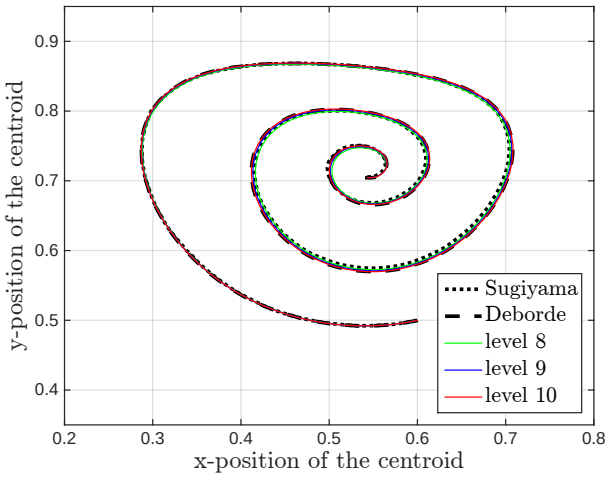
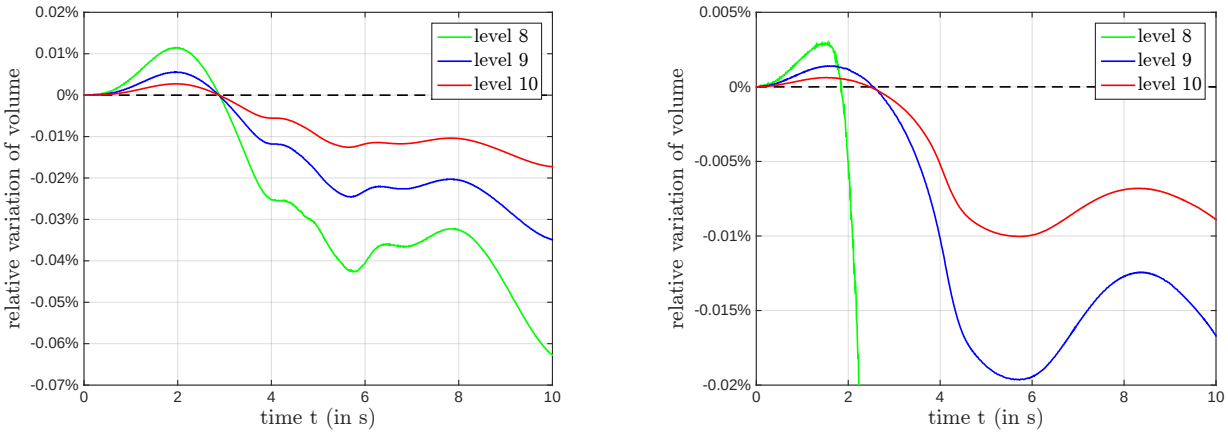



**Figure 13:** *Deformable soft cylinder ($G = 0.1Pa$) in a lid-driven cavity flow: approximated centroid position over time until $t = 20s$. The simulations are run on $L_8$ to $L_{10}$ uniform grids. The centroid position is computed following Eq. 4.1.*

below 0.02%. As expected, using interface regularization, this error decreases with first order accuracy.




**(a)** *Soft material ($G = 0.1Pa$).* **(b)** *Stiff material ($G = 10Pa$).*

**Figure 14:** *Two deformable cylinders in a lid-driven cavity flow: estimation of the relative mass-conservation error over time. The simulations are run on $L_8$ to $L_{10}$ uniform grids. The volume of the structure is computed following Eq. 4.2.*

## 4.2 Solid deformation due to an immersed actuator

We are now interested in simulating the elastic deformation induced by a rigid displacement imposed inside the structure. The goal is here to compare the accuracy and the computational costs for simulations on uniform and dynamic AMR grids.

An elastic cylinder is immersed in a in a cavity $\Omega = [0, 1]^2$m filled with viscous fluid as depicted in Figure 10. The fluid and structure densities and viscosities are $\rho_e = \rho_f = 1$ kg.m$^{-3}$ and $\mu_e = \mu_f = 10^{-2}$ Pa.s. The deformable cylinder is defined as a neo-Hookean moderately

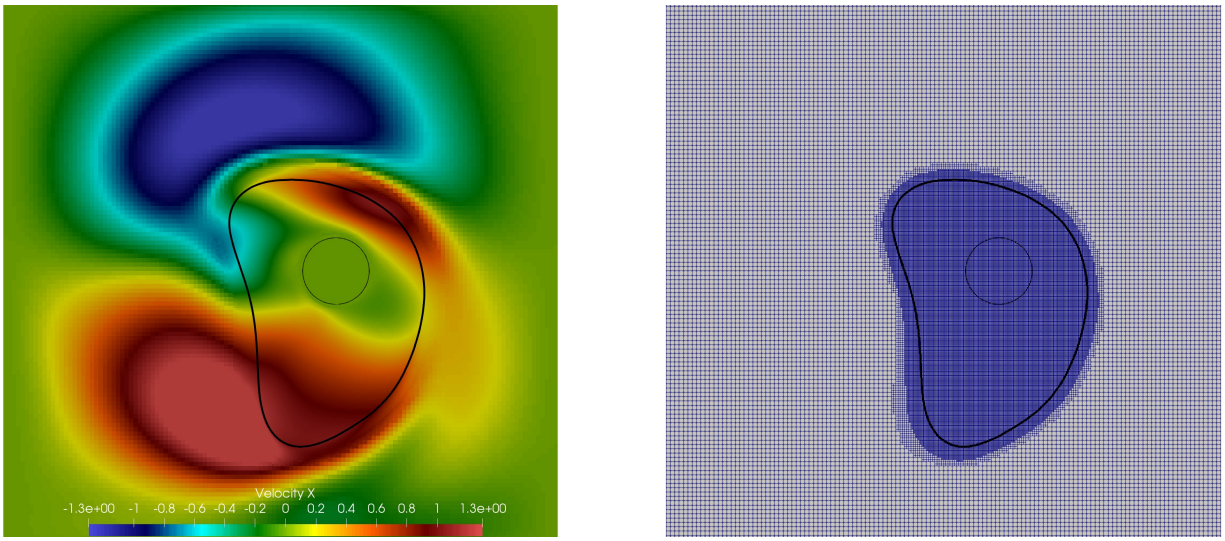
stiff material with $G = 1\text{Pa}$.
The fluid and structure are initially at rest and no-slip boundary conditions are apply on all external boundaries. A holder, defined by a small rigid cylinder with diameter equal to 0.12m, is immersed inside the elastic cylinder. The imposed motion of the holder describes a circle around the center of the domain, with a radius $r_s = 0.15\text{m}$. The imposed velocity $\mathbf{u}_s$ is defined analytically as:

$$\mathbf{u}_s(t) = \begin{cases} \pi r_s t\big(-\sin(\pi t^2/2), \cos(\pi t^2/2)\big)^T & \text{if } t < 2s, \\ 2\pi r_s\big(-\sin(2\pi t), \cos(2\pi t)\big)^T & \text{it } t \geq 2s. \end{cases}$$

A transient velocity of the rigid cylinder is imposed until $t = 2\text{s}$ to avoid heavy compression effects.

Numerical simulations are performed on both uniform and dynamic adaptive grids. Figure 15a displays the x-component of the velocity at time $t = 5\text{s}$ obtained using the dynamic AMR grid with 2 level jumps ($L_7 - L_9$ configuration) shown on Figure 15b.

For fair comparison between uniform and quadtree AMR grids, a same mesh resolution is imposed close to the interface $\Gamma_e$. To this end, a geometrical criterion based on the level-set function $\phi$ is used. Up to a minimum distance $D_{min}$ from the structure, the refinement level is set to the maximum level $L_{max}$. From this distance, the grid is progressively coarsened until the chosen minimum level $L_{min}$. In these simulations, the maximum level and minimum distance are set to $L_{max} = 9$ and $D_{min} = 10h_{min}$ respectively. The mesh adaptation procedure is activated each 0.05s to precisely follow the structure over time. To define the new grid data from the previous mesh, the solution is linearly reconstructed (following Eq. (3.14) in Section 3.2.3) according to the parent/children inheritance.




**(a)** *x-component of the velocity* **(b)** *dynamic AMR $L_7 - L_9$ grid*

**Figure 15:** *Solid deformation due to an immersed actuator at time $t = 5$. The thin and thick black lines represent the interfaces $\Gamma_s$ and $\Gamma_e$ respectively.*

In what follows, we investigate how the accuracy of the numerical solution is impacted by coarsening the grid in the fluid outside the narrow band. We compare the interface position for different numbers of coarsening ($L_{max} - L_{min}$), ranging from 1 to 3. The positions of $\Gamma_e$ at time $t = 2\text{s}$ and $t = 5\text{s}$ are presented in Figure 16. Despite the large difference with regard to the number of grid points, we obtain similar deformations. Moreover, as shown in Figure 17, the relative errors on mass remain of the same order of magnitude in all cases, thus showing that the coarsening in the fluid has a limited influence on the results. However, there is a significant

difference between the four simulations in terms of computational costs. The average number of grid points $N_{cells}$ and the computational time of each simulation $T_{run}$ are reported in Table 1. The execution times presented here were obtained for a same number of processors $N_p = 64$ and on the same supercomputer. The efficiency of the numerical method for different grids is evaluated with respect to the simulation on the $L_9$ uniform mesh. AS expected, the CPU decrease with the number of grid cells. The AMR process is however impacted by the mesh adaptation. Indeed, it is thus necessary to update the solution data on the new grid and to modify the parallel partition.

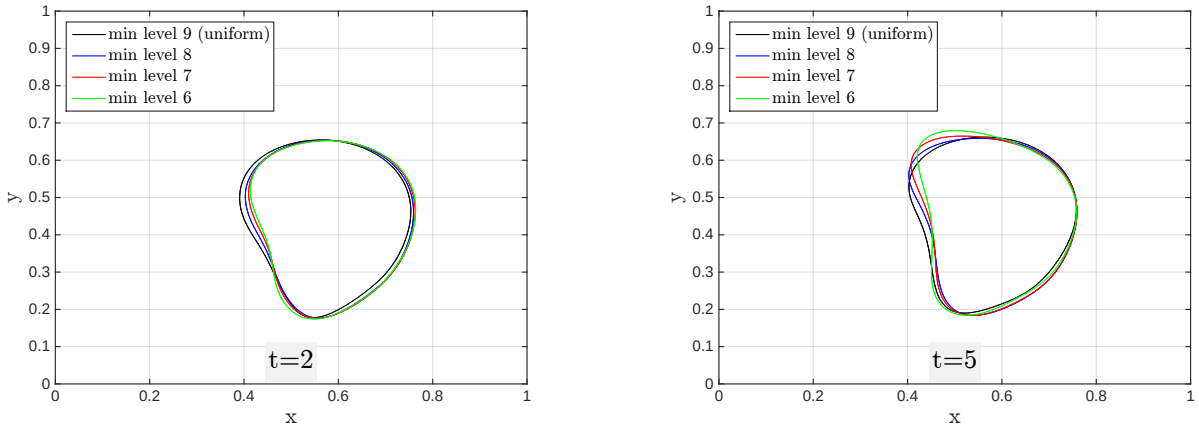



**Figure 16:** *Solid deformation due to an immersed actuator: position of the interface $\Gamma_e(t)$ at time $t = 2$ (left) and $t = 5$ (right). The comparison is performed between the $L_9$ uniform grid and dynamic AMR grids for a same maximum level of refinement $L_{max}$.*

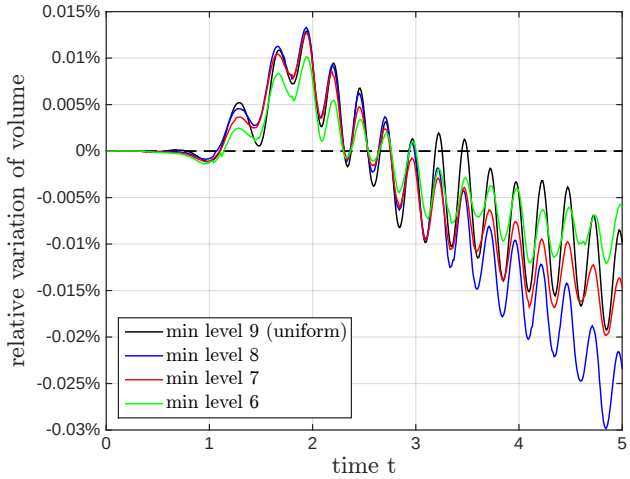



**Figure 17:** *Solid deformation due to an immersed actuator: estimation of the relative mass-conservation error over time. The comparison is performed between the $L_9$ uniform grid and dynamic AMR grids for a same maximum level of refinement $L_{max}$.*

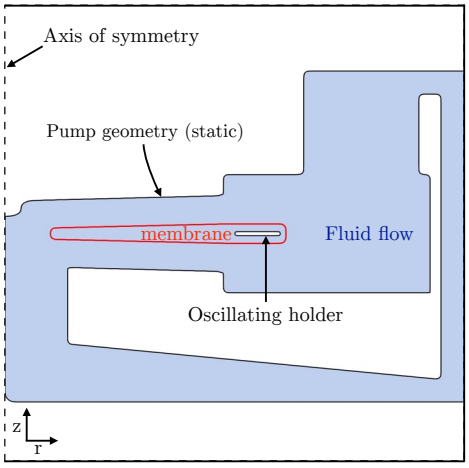
## 4.3 Oscillating membrane immersed in a cardiac pump geometry

A 2D-axi-symmetric test case in now considered. An elastic membrane is immersed in the

| minimum level $L_{min}$ | average $N_{cells}$ | gain | execution time $T_{run}$ (in s) | gain |
|---|---|---|---|---|
| 9 (uniform) | 262144 | / | 13406 | / |
| 8 | 86909 | -67% | 5618 | -58% |
| 7 | 43016 | -84% | 3432 | -74% |
| 6 | 31882 | -88% | 3015 | -78% |

**Table 1:** *Solid deformation due to an immersed actuator: computational cost of simulations for different dynamic AMR grids and gain when compared to the $L_9$ uniform simulation.*

closed circuit filled with a viscous and Newtonian fluid, mimicking blood. The computational domain is $\Omega = [0, 4] \times [-2, 2]$ cm. This test case is designed to mimic a Left Ventricular Assist Device (LVAD) behaviour. The immersed membrane is an elastic disc with variable thickness and a hole in its center. The motion of the membrane is driven by a holder, a rigid ring-shaped support, which oscillates up and down with an imposed amplitude and frequency. This FSI problem is stiff if we refer to the ratio $c_{wave}/c_{max,holder} \approx 168$, where $c_{wave}$ and $c_{max,holder}$ denote respectively the speed of elastic waves and the maximum speed of the holder. All the properties of the membrane and the holder are given in Table 18b.




**(a)** *spatial set-up.*

| Parameters | | Value |
|---|---|---|
| Membrane | $G$ | 1 MPa |
| | $\rho_e$ | $\rho_f = 1.17\times10^3$ kg/m$^3$ |
| | $\mu_e$ | $\mu_f = 1.6\times10^{-3}$ Pa.s |
| | inner radius | 0.5 cm |
| | outer radius | 2.45 cm |
| | inner thickness | 1 mm |
| | outer thickness | 1.7 mm |
| Holder | radial position | 2.2 cm |
| | inner radius | 2.0 cm |
| | outer radius | 2.4 cm |
| | thickness | 0.35 mm |
| | frequency | 100 Hz |
| | amplitude | 0.6 mm |

**(b)** *Material properties.*

**Figure 18:** *Oscillating membrane immersed in a cardiac pump geometry: spatial set-up and properties of solid structures.*

The whole system is initially at rest. Axi-symmetric boundary conditions are imposed on the left boundary (*i.e.* $\mathbf{u}_r = 0$, $\partial \mathbf{u}_z/\partial r = 0$). The geometry of the pump (blank part in figure 18a) are taken into account using Volume Penalization. While the velocity of external boundary (pump) is zero, the motion of the holder is imposed following 18b.

The simulations are performed on AMR grids, with dynamic mesh adaptation based on two criteria. Firstly, a geometrical criterion based on the distance function $\phi$ is used to impose a maximum refinement (level $L_{max}$) inside the membrane and in the vicinity of interfaces $\Gamma_s$ and $\Gamma_e$. Secondly, the Hessian of the velocity $H_{\mathbf{u}}$ is chosen as an *a posteriori* estimator of the solution regularity. The optimal refinement is determined according to two threshold values, denoted by $\eta_{min}$ and $\eta_{max}$. The refinement is imposed to be maximum if $||H_{\mathbf{u}}||_F > \eta_{max}$ and minimum if $||H_{\mathbf{u}}||_F < \eta_{min}$ where $||\cdot||_F$ refers to the Frobenius norm. Between these two threshold values, the refinement is smoothly regularized. The parameters used for this last criterion are set empirically to $\eta_{min} = 10^6$ and $\eta_{max} = 10^7$. Inside the steady pump geometry, the mesh is coarsened as much as possible to decrease the global mesh size. In addition, to

ensure a suitable mesh resolution inside the membrane, the maximum level of refinement is initially set to $L_{max} = 10$. The minimum level of refinement inside the fluid is set to $L_{min} = 8$. The mesh adaptation is performed 100 times per period $T = 1/f$. The level-set function and the backward characteristics are updated 500 times per period.Finally, $c_1 = c_2 = G/4$ are set in the Mooney-Rivlin model.

Snapshots of both the angular component of the vorticity and of the velocity magnitude are presented in Figure 19 after 2, 3, 4 and 5 periods of oscillation.

Figure 20 shows the temporal evolutions of the relative volume variation of the membrane and of the blood flow rate, for different oscillation frequencies. The volume of the membrane is well conserved with an error below 1%. As expected, at least in a given range, the mean flow rate increases with the holder frequency.

# 5 Conclusion and Perspectives

We have presented a fully Eulerian method to solve FSI problems on hierarchical Cartesian meshes. The approach developed here turns out to be robust and versatile since it enables the resolution of a wide range of FSI problems, including different kinds of rigid solids and hyper-elastic materials with arbitrary and complex shapes. For many applications, especially when the deformations are large and when the geometries are regular enough, this Eulerian method is particularly adapted.

We have developed a quadtree-based finite-volume discretization scheme which is second-order accurate in space. By considering only compact discretization stencils, the numerical method provides a suitable accuracy, while preserving a high efficiency of the numerical computations. Moreover, the dynamic AMR procedure is particularly interesting since we intend to guarantee a fine mesh resolution throughout the whole simulation where required, *e.g.* close to interfaces and where the regularity of the solution is low. Nevertheless, we have also highlighted that the precision of the discretization schemes can be degraded at level jumps. To compensate these inaccuracies, some stabilization techniques have been employed.

When dealing with very stiff materials, we have also observed that the resulting solution tends to be oscillatory while employing a standard finite-volume method. According to the stability analysis performed in one dimension, we have proved that the oscillatory behavior is due to the *FV2* scheme we employ to transport the backward characteristics. By enforcing the Exactly Linearity Preserving (ELP) property, we got rid of spurious oscillations that can lead to instabilities in the vicinity of interfaces. However, the computational cost of a FSI simulation remains critical since the formulation is explicit in time, leading to a significant elastic CFL constraint. To make the simulation more affordable, it would be necessary to consider an implicit formulation. For this purpose, one could investigate an iterative Newton method or draw on the implicit formulation introduced by Ii et al. [19].

The validity of the method has been verified firstly on uniform grids through the simulation of solid deformations in a lid-driven cavity flow. Both soft and stiff materials have been considered, which emphasizes the robustness of this method. In addition, mass conservation is effectively preserved, in a consistent way. Further, the interaction between a moderately stiff material actuated by a rigid non-deformable structure has been studied. The low impact of coarsening on the structure deformation has been confirmed and the results suggest that the numerical method offers a valuable trade-off between accuracy and feasibility of the computation. The simulation of a flow in a complex LVAD geometry is finally proposed as a bio-medical application. In particular, this numerical experiment shows the ability to reproduce a pumping

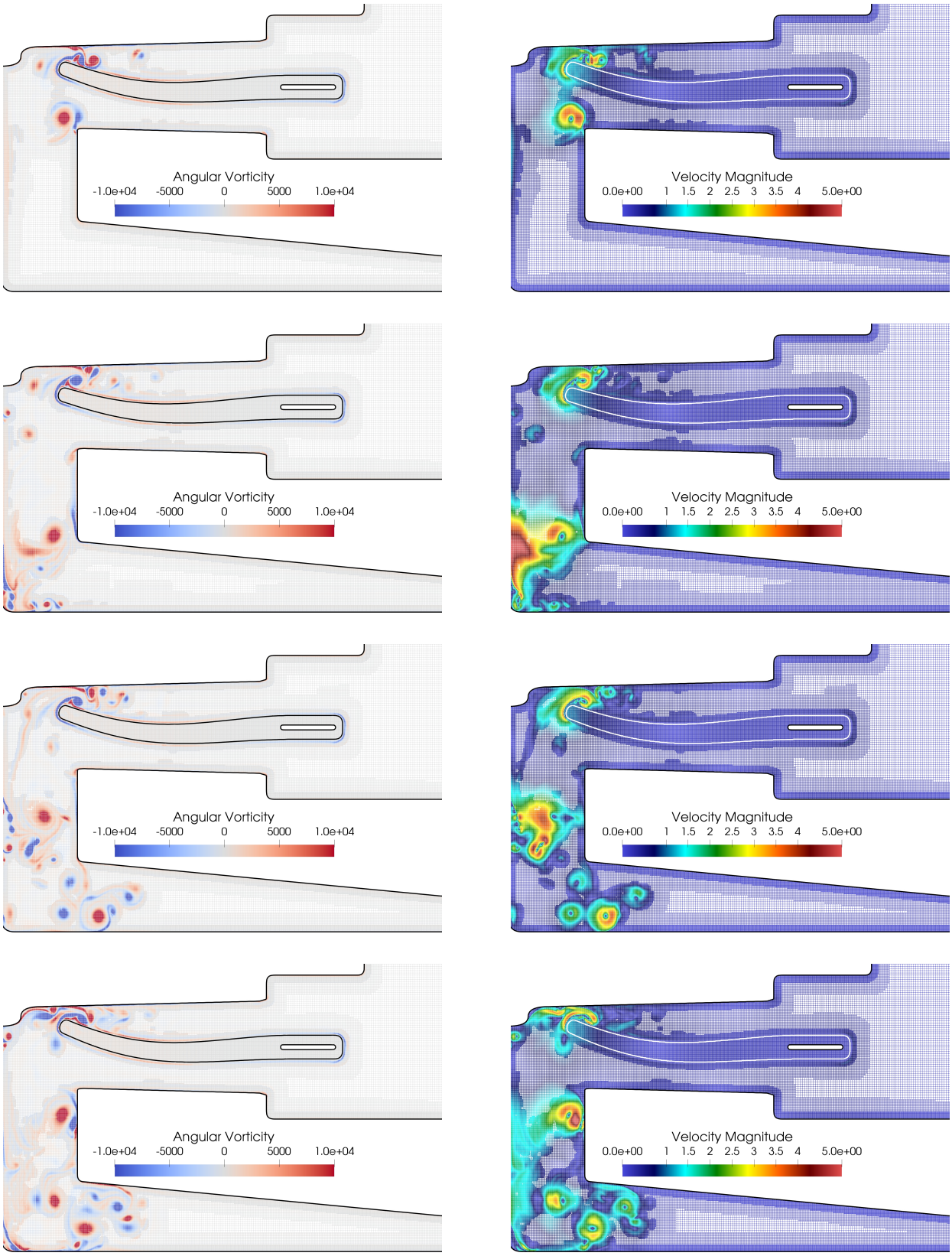
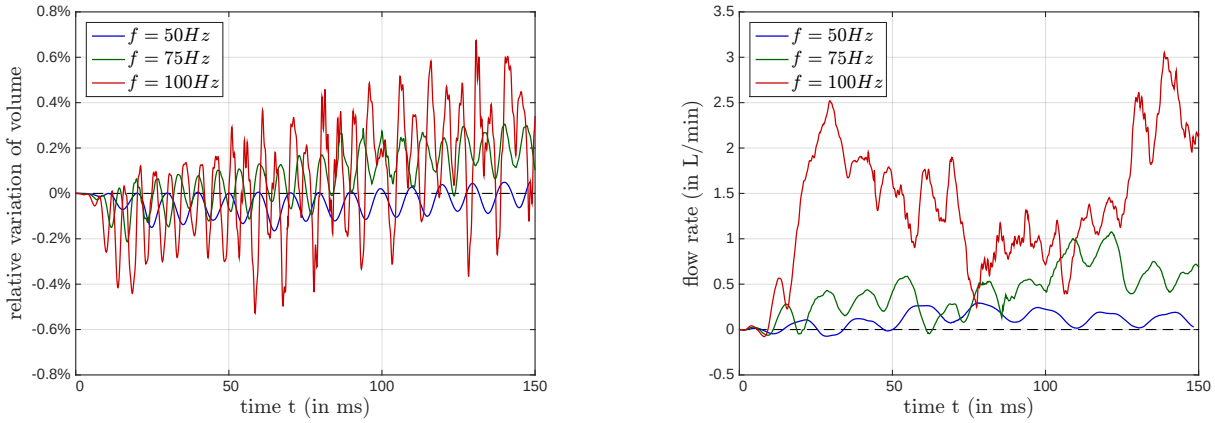



**Figure 19:** *Oscillating membrane immersed in a cardiac pump geometry: angular component of the vorticity (on the left) and velocity magnitude (on the right). From top to bottom: solution at time* $t = 2T$, $3T$, $4T$ *and* $5T$. *The AMR mesh is displayed in the background.*




**(a)** *Relative variation of volume.* **(b)** *Blood flow rate (in L/s).*

**Figure 20:** *Oscillating membrane immersed in a cardiac pump geometry : relative variation of the membrane volume and generated flow rate over time for different oscillation frequencies.*

flow thanks to the deformation of an oscillating rubber-like membrane. The simulation of other industrial applications, such as the simulation of blood flows in an aorta in the context of an aortic dissection or aneurysm, can be an outlook to this work.

# Appendix A Asymptotic analysis of the 1D discretization

Under the assumption of periodic boundary conditions, the solution $(\widetilde{u}, \widetilde{Y})$ is decomposed in a Fourier series. To simplify the notation, this series is restricted to a single Fourier mode. The discrete values $\widetilde{u}_j^n$ and $\widetilde{Y}_j^n$ are then expressed as:

$$\begin{aligned} \widetilde{u}_j^n &= \mathcal{U}(\xi)^n e^{2i\pi\xi j\Delta x}, \\ \widetilde{Y}_j^n &= \mathcal{Y}(\xi)^n e^{2i\pi\xi j\Delta x}, \end{aligned}$$

or in a more appropriate way by defining $\theta := 2\pi\xi\Delta x \in \mathbb{R}$:

$$\begin{aligned} \widetilde{u}_j^n &= \mathcal{U}(\theta)^n e^{ij\theta}, \\ \widetilde{Y}_j^n &= \mathcal{Y}(\theta)^n e^{ij\theta}, \end{aligned} \tag{A.1}$$

We recall here the different discretizations employed in Section 3.4. The 1D model (3.16) is integrated in time using a first-order linear multi-step method:

$$\frac{Y_j^{n+1} - Y_j^n}{\Delta t} = \mathcal{G}_j(u^{n+1}, Y^n), \quad \text{with } u^{n+1} = 2u^n - u^{n-1}, \tag{A.2a}$$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \mathcal{F}_j(u^n, Y^{n+1}). \tag{A.2b}$$

The linearized discrete formulations of the *FV2* and *FD2* schemes can be written for equation (A.2a) respectively as:

$$\frac{\widetilde{Y}_j^{n+1} - \widetilde{Y}_j^n}{\Delta t} = \begin{cases} -\dfrac{1}{2}\widetilde{u}_j - \dfrac{1}{4}\left(\widetilde{u}_{j+1} + \widetilde{u}_{j-1}\right) & \text{(FV2 scheme)}, \\ -\widetilde{u}_j & \text{(FD2 scheme)}, \end{cases} \tag{A.3}$$

in the limit of small deformations $\varepsilon \to 0$. The linearization of Equation (A.2b) gives:

$$\frac{\widetilde{u}_j^{n+1} - \widetilde{u}_j^n}{\Delta t} = -2G\frac{\Delta t}{\Delta x^2}(\widetilde{Y}_{j+1} - 2\widetilde{Y}_j + \widetilde{Y}_{j-1}). \tag{A.4}$$

We now apply the Fourier decomposition (A.1) to the linearized discretization of the model. Consequently, Equations (A.3) and (A.4) become respectively:

$$\mathcal{Y}^{n+1} = \begin{cases} \mathcal{Y}^n - 2\Delta t\ \cos^2(\theta/2)\,\mathcal{U}^n + \Delta t\ \cos^2(\theta/2)\,\mathcal{U}^{n-1} & \text{(FV2 scheme)}, \\ \mathcal{Y}^n - 2\Delta t\,\mathcal{U}^n + \Delta t\,\mathcal{U}^{n-1} & \text{(FD2 scheme)}, \end{cases} \tag{A.5}$$

and:

$$\mathcal{U}^{n+1} = \begin{cases} 4\dfrac{C^2}{\Delta t}\ \sin^2(\theta/2)\,\mathcal{Y}^n + \left(1 - 2C^2\sin^2(\theta)\right)\mathcal{U}^n + C^2\sin^2(\theta)\,\mathcal{U}^{n-1} & \text{(FV2 scheme)}, \\ 4\dfrac{C^2}{\Delta t}\sin^2(\theta/2)\,\mathcal{Y}^n + \left(1 - 8C^2\sin^2(\theta/2)\right)\mathcal{U}^n + 4C^2\sin^2(\theta/2)\,\mathcal{U}^{n-1} & \text{(FD2 scheme)}, \end{cases} \tag{A.6}$$

where $C = a\Delta t/\Delta x$ is a real positive number, and $a = \sqrt{2G}$ is the speed of elastic waves. In a more compact way, these two Equations (A.5) and (A.6) equations can be expressed using the following matrix form:

$$\begin{pmatrix} \mathcal{Y}^{n+1} \\ \mathcal{U}^{n+1} \\ \mathcal{U}^n \end{pmatrix} = \mathcal{A}\begin{pmatrix} \mathcal{Y}^n \\ \mathcal{U}^n \\ \mathcal{U}^{n-1} \end{pmatrix}. \tag{A.7}$$

where $\mathcal{A} \equiv \mathcal{A}(\theta)$ represents the amplification matrix. With respect to Equations (A.5) and (A.6), the amplification matrices obtained using the *FV2* and *FD2* schemes are respectively:

$$\mathcal{A}_{FV2} = \begin{pmatrix} 1 & -2\Delta t\cos^2(\theta/2) & \Delta t\cos^2(\theta/2) \\ 4\frac{C^2}{\Delta t}\ \sin^2(\theta/2) & 1 - 2C^2\sin^2(\theta) & C^2\sin^2(\theta) \\ 0 & 1 & 0 \end{pmatrix}, \tag{A.8}$$

and

$$\mathcal{A}_{FD2} = \begin{pmatrix} 1 & -2\Delta t & \Delta t \\ 4\frac{C^2}{\Delta t}\sin^2(\theta/2) & 1 - 8C^2\sin^2(\theta/2) & 4C^2\sin^2(\theta/2) \\ 0 & 1 & 0 \end{pmatrix}. \tag{A.9}$$

In order to conclude on the asymptotic stability of the numerical methods, the amplification factor of the matrices must be determined. In other words, one numerical method is asymptotically stable if the amplication factor $\rho(\mathcal{A})$ verifies:

$$\max_p\{|\lambda_p|\} \leq 1, \tag{A.10}$$

where the $\lambda_p$ refer to the eigenvalues of $\mathcal{A}$. The Figure 9 shows the amplification factors obtained from both matrices $\mathcal{A}_{FV2}$ and $\mathcal{A}_{FD2}$.

# Acknowledgments

Experiments presented in this paper were carried out using both the PlaFRIM experimental testbed, supported by INRIA, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional Aquitaine (see https://www.plafrim.fr/), and the CURTA platform, provided by the Mésocentre de Calcul Intensif Aquitain (MCIA) (see https://redmine.mcia.fr/).

This work has been partially supported by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska–Curie Actions, grant agreement 872442 (ARIA).

# References


[1] Shahnawaz Ahmed, Stanley Bak, Joyce McLaughlin, and Daniel Renzi. A third order accurate fast marching method for the eikonal equation in two dimensions. *SIAM Journal on Scientific Computing*, 33(5):2402–2420, 2011.

[2] Philippe Angot, Charles-Henri Bruneau, and Pierre Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81 (4):497–520, 1999.

[3] Tariq D Aslam. A partial differential equation approach to multidimensional extrapolation. *Journal of Computational Physics*, 193(1):349–355, 2004.

[4] Michel Bergmann and Angelo Iollo. Modeling and simulation of fish-like swimming. *Journal of Computational Physics*, 230(2):329–348, 2011.

[5] Michel Bergmann, Jessica Hovnanian, and Angelo Iollo. An accurate cartesian method for incompressible flows with moving boundaries. *Communications in Computational Physics*, 15(5):1266–1290, 2014.

[6] Michel Bergmann, Angelo Iollo, and Rajat Mittal. Effect of caudal fin flexibility on the propulsive efficiency of a fish-like swimmer. *Bioinspiration & biomimetics*, 9(4):046001, 2014.

[7] Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.

[8] Georges-Henri Cottet, Emmanuel Maitre, and Thomas Milcent. Eulerian formulation and level set models for incompressible fluid-structure interaction. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(3):471–492, 2008.

[9] Yves Coudière, Jean-Paul Vila, and Philippe Villedieu. Convergence rate of a finite volume scheme for a two dimensional convection-diffusion problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 33(3):493–516, 1999.

[10] Julien Deborde. *Modélisation et simulation de l'interaction fluide-structure élastique: application à l'atténuation des vagues*. PhD thesis, Bordeaux, 2017.

[11] Julien Deborde, Thomas Milcent, Pierre Lubin, and Stéphane Glockner. Numerical simulations of the interaction of solitary waves and elastic structures with a fully eulerian method. *Water Waves*, 2:433–466, 2020.

[12] Sarah Delcourte, Komla Domelevo, and Pascal Omnes. Discrete duality finite volume method for second order elliptic problems. *Hermes Science publishing*, pages 447–458, 2005.

[13] Antoine du Chéné, Chohong Min, and Frédéric Gibou. Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes. *Journal of Scientific Computing*, 35(2):114–131, 2008.

[14] Thomas Dunne and Rolf Rannacher. Adaptive finite element approximation of fluid-structure interaction based on an eulerian variational formulation. In *Fluid-structure interaction*, pages 110–145. Springer, 2006.

[15] Joel H Ferziger and Milovan Peric. *Computational methods for fluid dynamics*. Springer Science & Business Media, 2012.

[16] Yong-Bi Fu and Ray W Ogden. Nonlinear elasticity: Theory and applications. 2001.

[17] Katuhiko Goda. A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows. *Journal of computational physics*, 30(1):76–95, 1979.

[18] Arthur Guittet, Maxime Theillard, and Frédéric Gibou. A stable projection method for the incompressible navier–stokes equations on arbitrary geometries and adaptive quad/octrees. *Journal of Computational Physics*, 292:215–238, 2015.

[19] S Ii, K Sugiyama, S Takeuchi, S Takagi, and Yoichiro Matsumoto. An implicit full eulerian method for the fluid–structure interaction problem. *International Journal for Numerical Methods in Fluids*, 65(1-3):150–165, 2011.

[20] Benjamin Kadoch, Dmitry Kolomenskiy, Philippe Angot, and Kai Schneider. A volume penalization method for incompressible flows and scalar advection–diffusion with moving obstacles. *Journal of Computational Physics*, 231(12):4365–4383, 2012.

[21] Francky Luddens, Michel Bergmann, and Lisl Weynans. Enablers for high-order level set methods in fluid mechanics. *International Journal for Numerical Methods in Fluids*, 79 (12):654–675, 2015.

[22] Gilles Marckmann and Erwan Verron. Comparison of hyperelastic models for rubber-like materials. *Rubber chemistry and technology*, 79(5):835–858, 2006.

[23] Rajat Mittal, Haibo Dong, Meliha Bozkurttas, FM Najjar, Abel Vargas, and Alfred Von Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of computational physics*, 227(10):4825–4852, 2008.

[24] Melvin Mooney. A theory of large elastic deformation. *Journal of applied physics*, 11(9): 582–592, 1940.

[25] Guy M Morton. A computer oriented geodetic data base and a new technique in file sequencing. 1966.

[26] Chanwut Nitikitpaiboon and Klaus-Jürgen Bathe. An arbitrary lagrangian-eulerian velocity potential formulation for fluid-structure interaction. *Computers & structures*, 47(4-5): 871–891, 1993.

[27] Ray W Ogden. Nonlinear elasticity, anisotropy, material stability and residual stresses in soft tissue. In *Biomechanics of soft tissue in cardiovascular systems*, pages 65–108. Springer, 2003.

[28] Suhas Patankar. *Numerical heat transfer and fluid flow*. CRC press, 1980.

[29] Alice Raeli, Michel Bergmann, and Angelo Iollo. A finite-difference method for the variable coefficient poisson equation on hierarchical cartesian meshes. *Journal of computational physics*, 355:59–77, 2018.

[30] Rolf Rannacher and Thomas Richter. An adaptive finite element method for fluid-structure interaction problems based on a fully eulerian formulation. In *Fluid Structure Interaction II*, pages 159–191. Springer, 2011.

[31] Chae M Rhie and Wei-Liang Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA journal*, 21(11):1525–1532, 1983.

[32] Thomas Richter. A fully eulerian formulation for fluid–structure-interaction problems. *Journal of Computational Physics*, 233:227–240, 2013.

[33] Ronald S Rivlin. Large elastic deformations of isotropic materials iv. further developments of the general theory. *Philosophical transactions of the royal society of London. Series A, Mathematical and physical sciences*, 241(835):379–397, 1948.

[34] Giovanni Russo and Peter Smereka. A remark on computing distance functions. *Journal of Computational Physics*, 163(1):51–67, 2000.

[35] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[36] James A Sethian. Fast marching methods. *SIAM review*, 41(2):199–235, 1999.

[37] Atul Sharma. Level set method for computational multi-fluid dynamics: A review on developments, applications and analysis. *Sadhana*, 40(3):627–652, 2015.

[38] Kazuyasu Sugiyama, Satoshi Ii, Shintaro Takeuchi, Shu Takagi, and Yoichiro Matsumoto. A full eulerian finite difference approach for solving fluid–structure coupling problems. *Journal of Computational Physics*, 230(3):596–627, 2011.

[39] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1): 146–159, 1994.

[40] Roger Temam. Sur l'approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires (ii). *Archive for Rational Mechanics and Analysis*, 33(5): 377–385, 1969.

[41] Tayfun E Tezduyar, Mittal Behr, S Mittal, and J Liou. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: I. the concept and preliminary tests. *Computer methods in applied mechanics and engineering*, 94(3):339–351, 1992.

[42] Tayfun E Tezduyar, Mittal Behr, S Mittal, and J Liou. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: Ii. computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Computer methods in applied mechanics and engineering*, 94 (3):353–371, 1992.

[43] Yen-Hsi Richard Tsai, Li-Tien Cheng, Stanley Osher, and Hong-Kai Zhao. Fast sweeping algorithms for a class of hamilton–jacobi equations. *SIAM journal on numerical analysis*, 41(2):673–694, 2003.

[44] Stefan Turek and Jaroslav Hron. Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. In *Fluid-structure interaction*, pages 371–385. Springer, 2006.

[45] Boris Valkov, Chris H Rycroft, and Ken Kamrin. Eulerian method for multiphase interactions of soft solid bodies in fluids. *Journal of Applied Mechanics*, 82(4), 2015.

[46] Yong-Tao Zhang, Hong-Kai Zhao, and Jianliang Qian. High order fast sweeping methods for static hamilton–jacobi equations. *Journal of Scientific Computing*, 29(1):25–56, 2006.

[47] Hong Zhao, Jonathan B Freund, and Robert D Moser. A fixed-mesh method for incompressible flow–structure systems with finite solid deformations. *Journal of Computational Physics*, 227(6):3114–3140, 2008.