Journal of Computational Physics 471 (2022) 111647

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

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

Michel Bergmann [a,b], Antoine Fondanèche [a,b,*], Angelo Iollo [a,b]

[a] Memphis Team, INRIA, Talence, F-33400, France
[b] University of Bordeaux, IMB, UMR 5251, Talence, F-33400, France

ARTICLE INFO

Article history:
Received 17 December 2021
Received in revised form 16 September 2022
Accepted 20 September 2022
Available online 27 September 2022

Keywords:
Fluid-structure interaction
Monolithic Eulerian approach
Finite volume method
Hierarchical Cartesian meshes
Biomedical application

ABSTRACT

A quadtree-based fully Eulerian finite volume approach for the simulation of fluid-structure interaction problems is presented. Both fluid and structure phases, which are assumed to be incompressible and viscous, 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 medicine. 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 ciliated 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.

To solve this type of FSI problems, different numerical methods are developed depending on the severity of solid deformations. In the case of small or moderate interface deformations, body-fitted (or interface-tracking) methods are particularly efficient to reproduce accurately the behavior of elastic materials. Among others, the Arbitrary Lagrangian-Eulerian (ALE) [33,54] or Deforming Spatial-Domain/Space-Time (DSD/ST) [51,52] methods are commonly used in the literature. 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), these formalisms become challenging because of grid

[*] Corresponding author at: Memphis Team, INRIA, Talence, F-33400, France.
E-mail addresses: michel.bergmann@inria.fr (M. Bergmann), antoine.fondaneche@inria.fr (A. Fondanèche), angelo.iollo@inria.fr (A. Iollo).

https://doi.org/10.1016/j.jcp.2022.111647
0021-9991/© 2022 Elsevier Inc. All rights reserved.

deformation and distortions. If re-meshing techniques are classically employed to regularize the mesh, we can also mention the deformation/swapping approach [2] which avoids re-meshing. For the resolution of this type of problems, approaches that combine the Eulerian formalism in the fluid and the Lagrangian formalism in the solid are more appropriate. The Fictitious Domain, Immersed Boundary and Embedded Boundary methods (see, e.g., [5,9,18,23,36,37] and references therein) are based on non body-fitted meshes, in which the fluid–structure interface is deformed independently of the fluid mesh. However, these methods require additional care to take into account wall and transmission conditions at the fluid-structure interface.

In this work, we consider a fully Eulerian approach to FSI modeling. The fully Eulerian approaches, which are based on interface-capturing methods, are potentially less accurate than body fitted approaches, but they are more versatile and provide a more transparent handling of the mesh, if the structure is not too thin. Moreover, this type of methods enables a wide range of simulations to be performed, including all kinds of complex geometries, with limited offline geometrical modeling. The fluid and the structure are considered as a multi-phase incompressible viscous 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 [11,17,39,41,47,55]. 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 deviatoric elastic stress tensor. With a different perspective, in Sugiyama et al. [47] 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. [11]. It has been more recently employed by Deborde et al. [14] 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 neighbors, 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. [47], 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 [17] 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 [39] 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 [41] 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. [14] have proposed an improvement of this method by using a linear extrapolation [4] 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. 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.

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

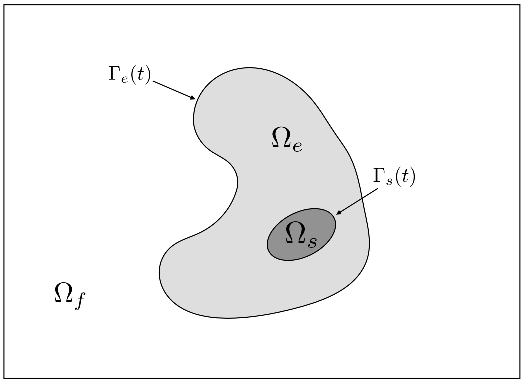


Fig. 1. Sketch of the fluid-structure interaction configuration.

2. Flow configuration and governing equations

The flow configuration under consideration is depicted in Fig. 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 Ω_f, the domain covered by the elastic material is Ω_e, and the non-deformable solid domains are Ω_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 Ω_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 Γ_e, and the interfaces between the solid and another phase (elastic or fluid) is Γ_s. Except the whole domain Ω, all other subdomains and interfaces can evolve in time.

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:

$$\begin{aligned} \rho_f\Big(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\Big) &= -\nabla p + \nabla\cdot\boldsymbol{\sigma}_f \qquad && \text{in } \Omega_f(t), \\ \nabla\cdot\mathbf{u} &= 0 && \text{in } \Omega_f(t), \end{aligned} \tag{1}$$

  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:

$$\begin{aligned} \rho_e\Big(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\Big) &= -\nabla p + \nabla\cdot\boldsymbol{\sigma}_e \qquad && \text{in } \Omega_e(t), \\ \nabla\cdot\mathbf{u} &= 0 && \text{in } \Omega_e(t), \end{aligned} \tag{2}$$

  with $\boldsymbol{\sigma}_e$ the deviatoric elastic stress tensor which will be defined later.
- Both velocity and normal components of the stress tensor $\widetilde{\boldsymbol{\sigma}} = -p\mathbf{I} + \boldsymbol{\sigma}$ are continuous across the interface $\Gamma_e(t)$, *i.e.* the jumps $[\cdot]_{\Gamma_e}$ across the interface are equal to zero:

$$\begin{aligned} [\mathbf{u}]_{\Gamma_e} &= 0, \\ [\widetilde{\boldsymbol{\sigma}}]_{\Gamma_e}\cdot\mathbf{n}_e &= \widetilde{\boldsymbol{\sigma}}_f\cdot\mathbf{n}_e - \widetilde{\boldsymbol{\sigma}}_e\cdot\mathbf{n}_e = 0, \end{aligned} \tag{3}$$

  where $\mathbf{n}_e$ is the normal vector to the interface Γ_e, arbitrarily pointing towards the fluid.
- Finally, the effect of rigid bodies is taken through no-slip boundary conditions on $\Gamma_s(t)$:

$$\mathbf{u} = \mathbf{u}_s \quad \text{on } \Gamma_s(t). \tag{4}$$

  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 Γ_s will be imposed, and is thus defined as an external data.

One difficulty in the numerical resolution of system (1)-(4) is the imposition of boundary conditions (3) and (4). One way would be to use a computational mesh that fits the interfaces Γ_e and Γ_s. Among the body-fitted methods, we can mention the work of Bathe [33], Turek [54], or Tezduyar [51,52]. The main advantage of this approach is that conditions (3) and (4) can be imposed in an accurate way on the interface-fitted mesh nodes. The main drawback, however, is that lots of remeshing steps have 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. The main advantage is that no remeshing steps are needed, but a special treatment is necessary at the boundary since no grid points are located on the interfaces (see [19] for instance). In the context of Immersed Boundary Methods (IBM) [7,30], the position of the boundary is reconstructed to impose precisely the boundary conditions. This strategy requires the construction of discretizations adapted to the geometry close to the interface, which implies the redefinition of specific stencils. However, the construction of high-order one-sided discretizations on quadtree meshes is very challenging, above all if we take into account the parallelism constraint, namely the exclusive use of compact stencils. As a consequence, we adopt a diffuse-interface formulation, in which the interfaces are represented in an implicit way using level-set functions.

2.1. Fluid-elastic structure coupling

Equations (1) and (2) differ only with material quantities such as density ρ and deviatoric stress tensor $\boldsymbol{\sigma}$. Let χ_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 η can generally be written as $\eta = (1-\chi_e)\eta_f + \chi_e \eta_e$. Thus, we have the following mixed quantities:

$$\begin{aligned} \rho &= (1-\chi_e)\rho_f + \chi_e \rho_e \qquad && \text{in } \Omega, \\ \boldsymbol{\sigma} &= (1-\chi_e)\boldsymbol{\sigma}_f + \chi_e \boldsymbol{\sigma}_e \qquad && \text{in } \Omega. \end{aligned} \tag{5}$$

According to this monolithic formulation (5), Equations (1) and (2) can be recasted as a single set of equations as:

$$\begin{aligned} \rho\Big(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\Big) &= -\nabla p + \nabla\cdot\boldsymbol{\sigma} \qquad && \text{in } [0, T_{max}] \times \Omega_f \cup \Omega_e, \\ \nabla\cdot\mathbf{u} &= 0 \qquad && \text{in } [0, T_{max}] \times \Omega_f \cup \Omega_e. \end{aligned} \tag{6}$$

This system is then completed with conditions (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 χ_e. This approach gives rise to a continuous problem across the interface, thus conditions (3) are naturally satisfied by continuity. As a counterpart, the accuracy in the vicinity of the interface is reduced since the position of the interface is taken into account only implicitly. The regularization of the mask function is performed using a regular level-set function. Here we chose the signed distance function ϕ arbitrarily defined to be negative in Ω_e. The mask function is hence smoothed on a narrow band of width 2δ:

$$\chi_e(\phi) = \begin{cases} 1 & \text{if } \phi < -\delta \\ 0 & \text{if } \phi > +\delta \\ 1 - \frac{1}{2}\big(1 + \frac{\phi}{\delta} + \frac{1}{\pi}\sin(\pi\frac{\phi}{\delta})\big) & \text{else.} \end{cases} \tag{7}$$

The width of the narrow band is usually limited to very few grid points, and we will choose $\delta = \min(3h_{min}, \delta_{min})$, h_{min} being the minimum characteristic size of the grid cells located near the interface Γ_e. Beyond a certain degree of grid refinement, i.e. when $h_{min} \to 0$, the regularization parameter is assumed to be constant ($\delta = \delta_{min}$), the equilibrium conditions at the interface are hence ensured by the continuity of mass and momentum flows at order δ_{min}.

The level-set function ϕ is transported using the following equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = 0 \quad \text{in } [0, T_{max}] \times \Omega. \tag{8}$$

Since the velocity is solenoidal in the whole domain $\Omega_f \cup \Omega_e$, we can use the conservative form of Equation (8):

$$\frac{\partial \phi}{\partial t} + \nabla\cdot(\mathbf{u}\phi) = 0 \quad \text{in } [0, T_{max}] \times \Omega. \tag{9}$$

The regularization (7) is based on Euclidian distance information. Since we have no guarantees that the level-set function ϕ is still a sign distance function after applying (9), reinitialization iterations have to be used (see [48]). Extension to several elastic bodies is straightforward introducing multiple level-set functions.

2.2. Volume penalization for rigid bodies

The previous sections have described a numerical modeling of system (1)-(3). These equations have been recasted in a simpler set of equations, based on the monolithic formulation (6). The last condition to close system (1)-(3) is thus the interaction of fluid and elastic materials with rigid bodies (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. [3], or more precisely its second order extension proposed in [7]. This method has been validated in several previous studies [6,8].

The main idea of this approach is to consider the whole system as a porous medium, with a variable permeability ε. In that regard, a rigid structure has a very low permeability, $\varepsilon \ll 1$. The system (1)-(3) can be solved adding extra penalization terms to the momentum equation in (6). We obtain the following equations in the whole domain Ω, written the conservation form:

$$\begin{aligned} \rho\Big(\frac{\partial \mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{u}\otimes\mathbf{u})\Big) &= -\nabla p + \nabla\cdot\boldsymbol{\sigma} + \frac{\chi_s}{\varepsilon}(\mathbf{u}_s - \mathbf{u}) \qquad && \text{in } [0, T_{max}]\times\Omega, \\ \nabla\cdot\mathbf{u} &= 0 && \text{in } [0, T_{max}]\times\Omega, \end{aligned} \tag{10}$$

where the mask function χ_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 (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 (10) converges towards the solution of the original decoupled system (see [3]) as $\sqrt{\varepsilon}$ tends to zero [27]. 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. This parameter is defined proportional to the volume of the smallest cell to be consistent with the other finite volume discretizations (in $1/|\Omega_i|$), and small enough that the term $\frac{\chi_s}{\varepsilon}(\mathbf{u}_s - \mathbf{u})$ becomes predominant over all other terms.

2.3. Definition of the deviatoric elastic stress tensor

The last point is the definition of the deviatoric elastic stress tensor $\boldsymbol{\sigma}_e$. Let Ω_0 be the initial or reference configuration of Ω_e at time $t=0$ and Ω_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 Ω_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:

$$\frac{\partial \mathbf{X}}{\partial t}(t,\boldsymbol{\xi}) = \mathbf{u}(t,\mathbf{X}(t,\boldsymbol{\xi})), \qquad \mathbf{X}(t=0,\boldsymbol{\xi}) = \boldsymbol{\xi}. \tag{11}$$

In an Eulerian framework, the continuous medium is described through the deformed configuration Ω_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 Fig. 2) since we have the relations:

$$\mathbf{X}(t,\mathbf{Y}(t,\mathbf{x})) = \mathbf{x} \quad \text{and} \quad \mathbf{Y}(t,\mathbf{X}(t,\boldsymbol{\xi})) = \boldsymbol{\xi}. \tag{12}$$

Differentiating the second equation of (12) with respect to time, we get a transport equation:

$$\begin{cases} \dfrac{\partial \mathbf{Y}}{\partial t}(t,\mathbf{x}) + \mathbf{u}(t,\mathbf{x})\cdot\nabla_{\mathbf{x}}\mathbf{Y}(t,\mathbf{x}) = 0, \\ \mathbf{Y}(t=0,\mathbf{x}) = \mathbf{x}. \end{cases} \tag{13}$$

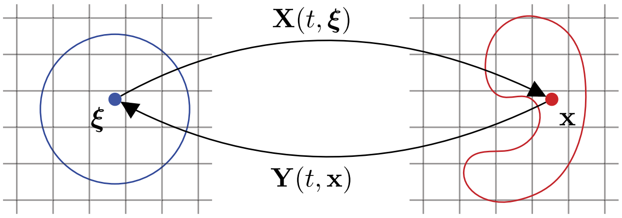


Fig. 2. Forward and backward characteristics are used to describe the deformation of a continuous medium.

This Eulerian relation is equivalent to the Lagrangian formulation (11). Moreover, differentiating one of the relations of (12) with respect to $\boldsymbol{\xi}$ or $\mathbf{x}$, the deformation tensor $\mathbf{F}$ can be expressed in the sense of both formalisms as

$$\mathbf{F} := [\nabla_{\boldsymbol{\xi}}\mathbf{X}] = [\nabla_{\mathbf{x}}\mathbf{Y}]^{-1}. \tag{14}$$

Under the assumption of incompressibility, we have the relation:

$$J := \det(\mathbf{F}) = 1. \tag{15}$$

In the Eulerian framework, the backward characteristics $\mathbf{Y}$ are considered to describe the deformation of an elastic material represented by Ω_e. The evolution in time of $\mathbf{Y}$ is followed by solving the transport problem (13). The deformation tensor $\mathbf{F}$ can be computed as a function of $\mathbf{Y}$ using relation (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 [22,34].

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:

$$\mathcal{E} = \int_{\Omega_0} W(\mathbf{F}(t,\boldsymbol{\xi}))\, d\boldsymbol{\xi}. \tag{16}$$

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 $\mathbf{B} = \mathbf{F}\mathbf{F}^T$):

$$\mathcal{E} = \int_{\Omega_0} \tilde{W}(\imath_{\mathbf{C}})\, d\boldsymbol{\xi} = \int_{\Omega_t} \tilde{W}(\imath_{\mathbf{B}})\; J^{-1}\, d\mathbf{x}, \tag{17}$$

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 (15). In this context, it is hence possible to prove that the deviatoric elastic stress tensor $\boldsymbol{\sigma}_e$ can be expressed as:

$$\boldsymbol{\sigma}_e = -p^*\mathbf{I} + 2\frac{\partial W}{\partial I_1}\mathbf{B} - \frac{\partial W}{\partial I_2}\mathbf{B}^{-1}, \tag{18}$$

where p^* refers to an arbitrary hydrostatic pressure. This Eulerian formulation shows that the deviatoric 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 [31,42], for which the strain-energy function is defined (in three dimensions) as:

$$W = c_1(I_1 - 3) + c_2(I_2 - 3), \tag{19}$$

where $c_1, c_2 \geq 0$ are the empirical parameters of the material related to the shear modulus through the consistency relation $G = 2(c_1 + c_2)$. From Equations (18) and (19), the Mooney-Rivlin deviatoric stress tensor is consequently:

$$\boldsymbol{\sigma}_e = -(2c_1I_1 - 2c_2I_2)\mathbf{I} + 2c_1\mathbf{B} - 2c_2\mathbf{B}^{-1}. \tag{20}$$

This kind of model is particularly adapted to rubber-like materials which undergo moderate deformations (200%-250%), as reported by [29]. 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 recently been studied by [11], and then used by [14]. In the latter paper, the numerical method relies on the use of fully uniform Cartesian meshes whereas here we rely on new quadtree-based discretizations on compact stencils.

We solve the whole fluid-structure system with a monolithic model, in contrast to the partitioned approaches where the models and the schemes for the fluid and the structure are distinct. Here, the model and the scheme are the same on the whole domain Ω, without making any distinction between the different media.

The single-continuum model in the entire computational domain reads:

$$\begin{aligned} \rho\Big(\frac{\partial \mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{u}\otimes\mathbf{u})\Big) &= -\nabla p + \nabla\cdot\boldsymbol{\sigma}(\mathbf{u},\mathbf{Y},\phi) + \frac{\chi_s}{\varepsilon}(\mathbf{u}_s - \mathbf{u}) && \text{in } [0,T_{max}]\times\Omega, \\ \nabla\cdot\mathbf{u} &= 0 && \text{in } [0,T_{max}]\times\Omega, \end{aligned} \tag{21a}$$

$$\begin{aligned} &\frac{\partial \mathbf{Y}}{\partial t} + \nabla\cdot(\mathbf{u}\otimes\mathbf{Y}) = 0 && \text{in } [0,T_{max}]\times\Omega, \\ &\frac{\partial \phi}{\partial t} + \nabla\cdot(\mathbf{u}\phi) = 0 && \text{in } [0,T_{max}]\times\Omega, \end{aligned} \tag{21b}$$

where the deviatoric stress tensor is defined as:

$$\boldsymbol{\sigma}(\mathbf{u},\mathbf{Y},\phi) = \boldsymbol{\sigma}_f(\mathbf{u}) + \chi_e(\phi)\boldsymbol{\sigma}_e(\mathbf{Y}), \tag{22}$$

with $\boldsymbol{\sigma}_f = \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ and $\boldsymbol{\sigma}_e = -(2c_1 I_1 - 2c_2 I_2)\mathbf{I} + 2c_1\mathbf{B} - 2c_2\mathbf{B}^{-1}$. Using this formulation, the deviatoric stress tensor (22) and the pressure are regularized across the interface, which guarantees the compliance with the transmission boundary conditions (3).

To improve the regularity of the results and to simplify the model, both the fluid and elastic materials are chosen to be viscous. In this work, we impose that the viscosities are identical, i.e. $\mu = \mu_f = \mu_e$. This system is closed using appropriate initial and boundary conditions for $\mathbf{u}$, p $\mathbf{Y}$ and ϕ.

A mathematical analysis of this Eulerian model has been proposed by Cottet et al. [11]. 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 (21). This model has already been studied by Deborde et al. [14] 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 (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. [38] 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 φ at time t^n. 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 maximum CFL number has been chosen by trial and error to ensure the stability of the numerical method and set to 0.25.

Since the time step might change at each iteration, the linear systems to solve are hence different from one iteration to another but the matrix coming from the discretizations in space can be stored in memory once and for all. To simplify the description of the numerical schemes, the time step is assumed to be constant in the remainder of this section. The generalization to a variable time step is straightforward.

One of the main difficulty in solving system (21) is the numerical discretization of the elastic part of the deviatoric stress tensor in (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 (21) could be solved implicitly 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 (21a) and (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 (21b) as a first step. The level-set function ϕ is transported from time t^n to t^{n+1} using the classical second-order TVD Runge-Kutta scheme as:

$$\begin{aligned}
\phi^* &= \phi^n - \Delta t \nabla \cdot (\mathbf{u}\phi)^n, \\
\phi^{n+1} &= \frac{1}{2}\phi^n + \frac{1}{2}\phi^* - \frac{\Delta t}{2}\nabla \cdot (\mathbf{u}\phi)^*.
\end{aligned} \tag{23}$$

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

Then, system (21a) is solved using the fractional time step method introduced by Chorin [10] and Temam [50]. We first compute a velocity field $\mathbf{u}^*$ from a pressure guess q with a second-order Gear scheme applied to the momentum equation:

$$\begin{aligned}
\frac{3\mathbf{u}^* - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\Delta t} &+ \left(2\nabla \cdot (\mathbf{u}\otimes\mathbf{u})^n - \nabla \cdot (\mathbf{u}\otimes\mathbf{u})^{n-1}\right) \\
&= \frac{1}{\rho}\Big(- \nabla q + \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}^*, \mathbf{Y}^{n+1}, \phi^{n+1}) + \frac{\chi_s^{n+1}}{\epsilon}(\mathbf{u}_s^{n+1} - \mathbf{u}^*)\Big)
\end{aligned} \tag{24}$$

Since the backward characteristics and the level-set function have already been advected in time, the quantities $\mathbf{Y}^{n+1}$ and ϕ^{n+1} are known and the elastic part of the deviatoric stress tensor can be finally computed in an explicit way. Also, since $\mathbf{u}_s^{n+1}$ and thus χ_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

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\left(\nabla p^{n+1} - \nabla q\right), \tag{25}$$

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

$$\Delta p' = \nabla \cdot \mathbf{u}^*, \tag{26}$$

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. In a FSI problem, the pressure is in general discontinuous across the fluid-solid interface. However, in our model the interface is regularized and hence the pressure is continuous even if it has a steep gradient. Once the pressure increment p' is computed, a correction step is performed:

$$\begin{cases}
p^{n+1} = q + \dfrac{\rho}{\Delta t}\, p', & \text{(a)} \\
\mathbf{u}^{n+1} = \mathbf{u}^* - \nabla p'. & \text{(b)}
\end{cases} \tag{27}$$

In this study, we use the incremental version of the projection method proposed by Goda [24] for which $q := p^n$.

3.2. Spatial discretizations

In this work, the monolithic model (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 Fig. 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 [32] in 1966, which enables the manipulation of the

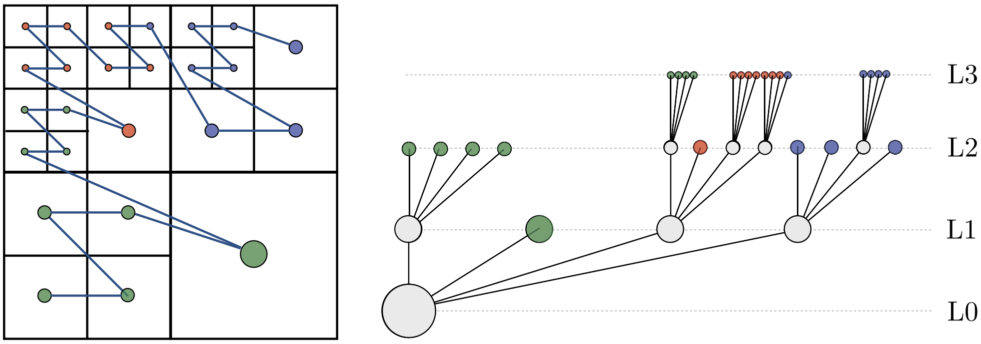


Fig. 3. Global Z-order indexing for a two-dimensional graded quadtree grid. The domain decomposition is depicted with colors. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

2D grid data from a 1D vector structure. We can thus access all the information 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 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 quantities, namely a scalar function φ and a vector field $\mathbf{v} \in \mathbb{R}^2$. The square domain Ω is decomposed into a quadtree partition of N_{cells} small square cells Ω_i of level L_i (being the *leaves* of the tree) such that $\Omega = \dot{\bigcup}_i \Omega_i$. 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 Ω_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 Ω. We denote by $\mathbf{x}_i$ the center of the cell Ω_i and $\varphi_i := \varphi(\mathbf{x}_i)$ the discrete value of a quantity φ 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 Ω_{out} and Ω_{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 Ω_{in} to Ω_{out}. The discrete values of φ in Ω_{in} and Ω_{out} are denoted by φ_{in} and φ_{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 (26), 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 [20], traditional collocated methods cannot simultaneously guarantee pressure regularity and mass conservation. One way to overcome this problem has been proposed by Patankar [35] and consists in adopting a fully staggered arrangement of variables. For this Marker-And-Cell (MAC) method, the prediction step (24) and the Poisson equation (26) 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 [40], and employed later by Mittal et al. [30], is chosen. An auxiliary face-center velocity, U^*_{fc}, is introduced:

$$\begin{cases} \tilde{\mathbf{u}} = \mathbf{u}^* + \dfrac{\Delta t}{\rho}(\nabla p^n)_{cc} & \text{(a)} \\ \widetilde{U_{fc}} = \mathcal{F}(\tilde{\mathbf{u}}) & \text{(b)} \\ U^*_{fc} = \widetilde{U_{fc}} - \dfrac{\Delta t}{\rho}(\nabla p^n)_{fc} & \text{(c)} \end{cases} \tag{28}$$

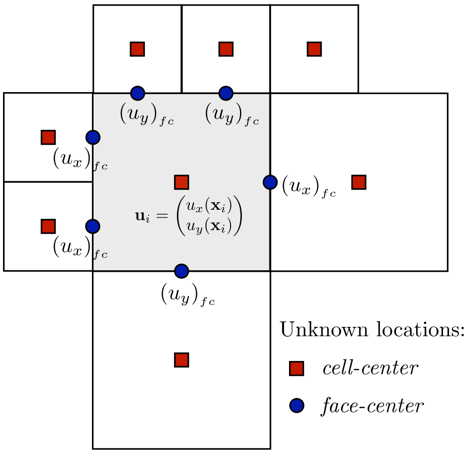


Fig. 4. Pattern of the velocity arrangement on quadtree grids.

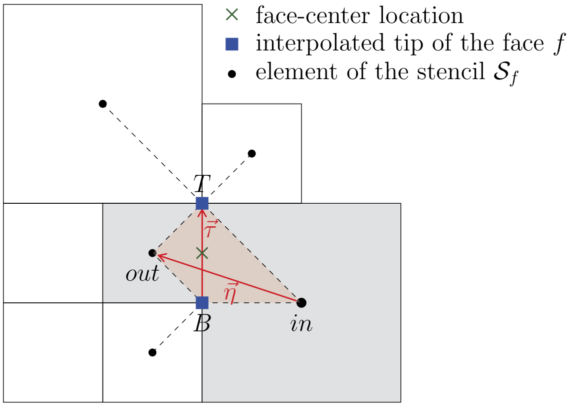

× face-center location
■ interpolated tip of the face f
• element of the stencil $\mathcal{S}_f$
T
$\vec{\tau}$
out
$\vec{\eta}$
B
in

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

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 Fig. 4.

The face-center velocity U^*_{fc} is now used to compute the right hand side of the Poisson equation (26):

$$\Delta p' = \frac{\rho}{\Delta t} \nabla \cdot U^*_{fc}. \tag{29}$$

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

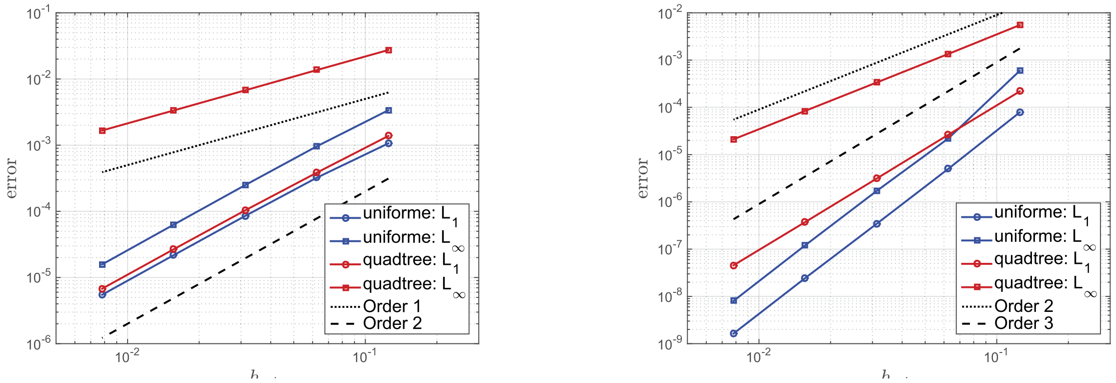
$$\begin{cases} \mathbf{u}^{n+1} = \mathbf{u}^* - (\nabla p')_{cc}, & \text{(a)} \\ U^{n+1}_{fc} = U^*_{fc} - (\nabla p')_{fc}. & \text{(b)} \end{cases} \tag{30}$$

If the face-center velocity U^{n+1}_{fc} satisfies the discrete mass conservation within the limit of the Poisson solver tolerance ϵ_Δ (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 φ_{fc} from cell-center values, a second order face-center interpolator $\mathcal{I}$ is introduced. Fig. 5 illustrates the stencil used for this interpolation. First, if the face configuration is uniform, which means



(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.*

Fig. 6. L_1 and L_∞ truncation errors $||\nabla \cdot \mathbf{v}_{num} - \nabla \cdot \mathbf{v}_{ex}||$ obtained for the divergence operator on uniform and quadtree meshes. The analytical solution is $\mathbf{v}_{ex}(x, y) = (-y, x)^T \; f/(0.385R)$ in $\Omega = [-2, 2]^2$ with $R = \sqrt{x^2 + y^2}$ and $f = \tanh(R)/\cosh^2(R)$. Dirichlet boundary conditions are prescribed on $\partial\Omega$. Tests are performed on uniform and quadtree grids (which rely on a refinement by boxes, see Fig. 7) for a same minimum level of refinement L_{min}.

that the two cells Ω_{in} and Ω_{out} have a same level of refinement, φ_{fc} is naturally defined as the average of φ_{out} and φ_{in}. Otherwise, φ_{fc} is computed as the average of φ_T and φ_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.

In the computation of the normal face-center velocity U_{fc} (see Equation (28b)), 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 Δt. 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 Fig. 6.

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

$$\widetilde{U}_{fc} = \mathcal{F}(\tilde{\mathbf{u}}) := \begin{cases} \mathcal{I}(\tilde{\mathbf{u}}) & \text{if the stencils of the two cells are both uniforms,} \\ U^n_{fc} + \mathcal{I}(\tilde{\mathbf{u}} - \mathbf{u}^n) & \text{else.} \end{cases} \tag{31}$$

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 [12,15]. This method, called diamond method, is based on the choice of artificial diamond-like polygons, as depicted in Fig. 5. The finite volume formulation of the Laplacian operator is discretized using a mid-point formula:

$$(\Delta\varphi)_i = \frac{1}{|\Omega_i|} \int\limits_{\partial\Omega_i} \nabla\varphi \cdot \mathbf{n}\, ds = \sum_{f \subset \partial\Omega_i} (\nabla\varphi)_{fc} \cdot \mathbf{n}_f \frac{|f|}{|\Omega_i|} + \mathcal{O}(h_i^2). \tag{32}$$

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

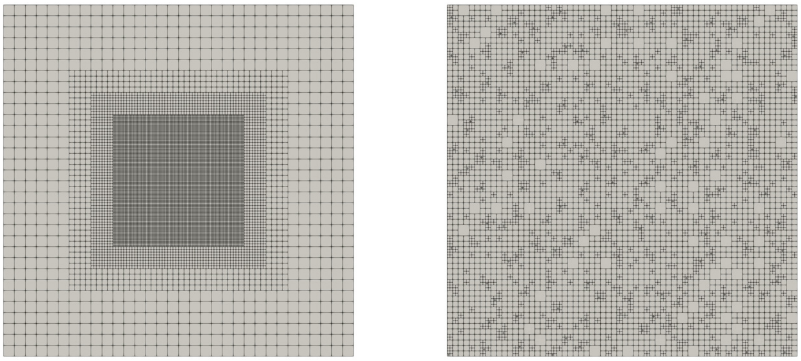
$$\nabla\varphi \cdot \boldsymbol{\eta} = \varphi_{out} - \varphi_{in}, \tag{33a}$$

$$\nabla\varphi \cdot \boldsymbol{\tau} = \varphi_T - \varphi_B, \tag{33b}$$

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 (33a) 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:

$$(\nabla\varphi)_{fc} \cdot \mathbf{n}_f = c_\tau(\varphi_T - \varphi_B) + c_\eta(\varphi_{out} - \varphi_{in}). \tag{34}$$

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



grid	uniform				by-boxes AMR				pseudo-random AMR			
L_{min}	L^1	q_1	L^∞	q_∞	L^1	q_1	L^∞	q_∞	L^1	q_1	L^∞	q_∞
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

Fig. 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.

The Poisson matrix resulting from this diamond discretization is not symmetric and positive definite. Guittet et al. [25] 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 Fig. 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 ϕ and the backward characteristics $\mathbf{Y}$ (see Equations (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 (24). 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:

$$\int\limits_{\Omega_i} \nabla \cdot \mathbf{F}(\varphi)\, d\mathbf{x} = \int\limits_{\partial\Omega_i} \mathbf{F}(\varphi) \cdot \mathbf{n}\, ds = \sum_{f \subset \partial\Omega_i} \mathbf{F}(\varphi_{fc}) \cdot \mathbf{n}_f\ |f|, \tag{35}$$

where the flux is defined by $\mathbf{F}(\varphi) := U\varphi$. Since the normal face-center velocity U_{fc} is directly computed from equation (30b), it is only required to reconstruct the function φ 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 with U_{fc}, involved in the discretization is only second order accurate. To that end, the linear reconstruction of φ in each cell Ω_i is performed using the cell-center gradient as:

$$\varphi\big|_{\Omega_i}(\mathbf{x}) = \varphi_i + (\mathbf{x} - \mathbf{x}_i)^T (\nabla\varphi)_i, \tag{36}$$

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 Ω_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:

$$\widetilde{\mathbf{F}}_{O2}(\varphi^+, \varphi^-) = \frac{1}{2} U_{fc}(\varphi^+ + \varphi^-) - \frac{1}{2}|U_{fc}|(\varphi^+ - \varphi^-), \tag{37}$$

where $\varphi^- = \varphi\big|_{\Omega_{in}}(\mathbf{x}_{fc})$ and $\varphi^+ = \varphi\big|_{\Omega_{out}}(\mathbf{x}_{fc})$ refer to the reconstruction of φ_{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. For the validation of the finite volume discretization, the reader can refer to Appendix A.

3.3. Re-initialization and extrapolation algorithms

In this study, the level-set function ϕ 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 (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 behavior 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 behavior. First of all, we can mention the relaxation method proposed by Russo and Smereka [43] and its extensions to high-order precision by [16,28] 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 [53,56] 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 [44,45] 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 deviatoric 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. [47], 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 [4] and employed recently by Deborde et al. [14]. The two transport equations are solved using a semi-Lagrangian scheme based on (bi-)linear upwind interpolations.

The reset of the level-set and backward characteristics function could in principle be done at each time step but it is necessary only when the regularity degrades. The extrapolation of the backward characteristics relies on the normal vector, computed from the level-set function. For regularity reasons, we thus chose to update the level-set function immediately before the extrapolation step.

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:

$$\partial_t u = \mathcal{F}(u, Y) := -u\partial_x u + \partial_x \sigma_e(Y), \tag{38a}$$

$$\partial_t Y = \mathcal{G}(u, Y) := -u\partial_x Y, \tag{38b}$$

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:

$$\sigma_e(Y) = G[(\partial_x Y)^{-2} - 1]. \tag{39}$$

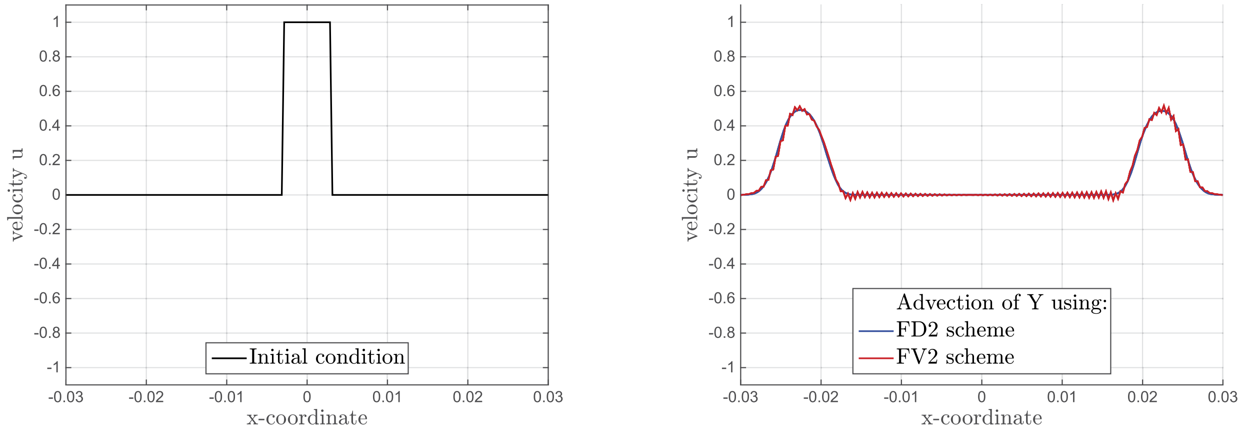
Here, the computational domain is purely elastic, *i.e.* $\Omega = \Omega_e$, the level-set function ϕ is thus not taken into account.

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 Δx. As depicted in Fig. 8a, a discontinuous Heaviside function is prescribed as an initial condition:

$$u(t = 0, x) = \begin{cases} 1 & \text{if } |x| < 0.003, \\ 0 & \text{else,} \end{cases} \tag{40}$$

to enable the apparition of numerical instabilities. The shear modulus is set to $G = 1$ MPa to simulate a stiff material. The 1D model (38) is integration in time using a first-order linear multi-step method:



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

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

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

$$\begin{aligned} &\frac{Y_i^{n+1}-Y_i^n}{\Delta t} = \mathcal{G}_i(u^{n+1}, Y^n), \quad \text{with } u^{n+1} = 2u^n - u^{n-1}, \\ &\frac{u_i^{n+1}-u_i^n}{\Delta t} = \mathcal{F}_i(u^n, Y^{n+1}). \end{aligned} \tag{41}$$

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 $u > 0$):

$$\left(\tilde{A}_{FV2}(Y)\right)_i = \frac{u_{i+\frac{1}{2}}}{4\Delta x}(Y_{i+1} - Y_{i-1}) - \frac{u_{i-\frac{1}{2}}}{4\Delta x}(-3Y_i + 4Y_{i-1} - Y_{i-2}). \tag{42}$$

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

$$\left(\tilde{A}_{FD2}(Y)\right)_i = \frac{u_i}{4\Delta x}(Y_{i+1} + 3Y_i - 5Y_{i-1} + Y_{i-2}). \tag{43}$$

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 Fig. 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. This asymptotic analysis is intended to provide an explanation of the results in Fig. 8. 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:

$$u(t,x) = u_0(t,x) + \varepsilon\widetilde{u}(t,x), \tag{44a}$$

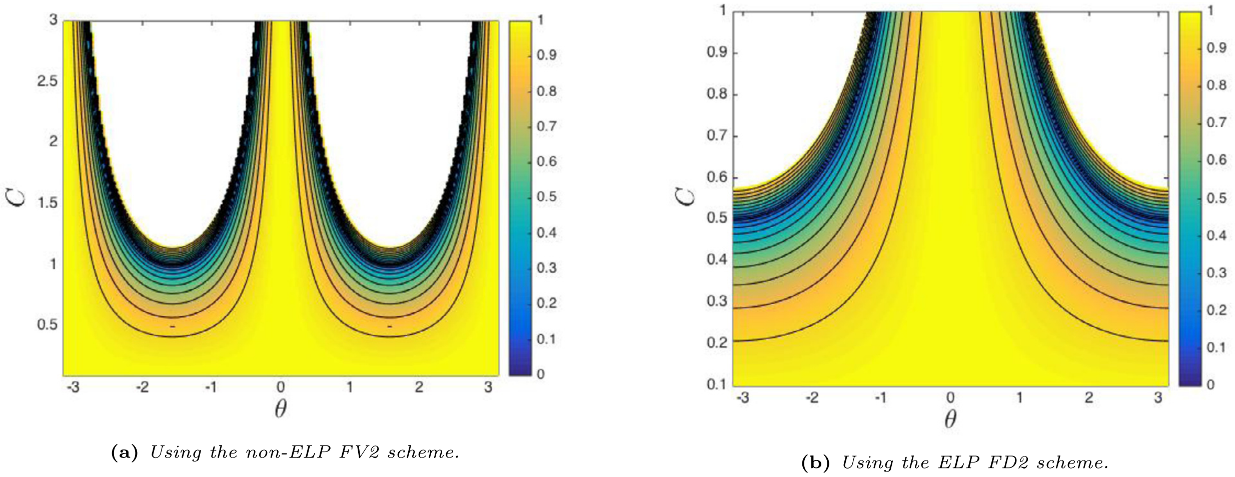
$$Y(t,x) = Y_0(t,x) + \varepsilon\widetilde{Y}(t,x). \tag{44b}$$

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 (44) into the 1D model (38), we have $u_0 \equiv 0$ and we obtain the following linearized model:

$$\partial_t\widetilde{u} = -2G\partial_{xx}^2\widetilde{Y}, \tag{45a}$$

$$\partial_t\widetilde{Y} = -\widetilde{u}. \tag{45b}$$

Differentiating Equations (45) with respect to time, these equations can be recasted into wave equations for which the propagation speed is $a = \sqrt{2G}$.



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

(b) *Using the ELP FD2 scheme.*

Fig. 9. Asymptotic analysis of the model (38): amplification factors as a function of the frequency of a Fourier mode θ 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.

We now compare the asymptotic behavior of the two advection schemes *FV2* and *FD2*. From the discrete formulations (42) and (43), the linearization leads to:

$$\textit{(FV2):}\quad (\partial_t \widetilde{Y})_i = -\widetilde{u}_i - \frac{\Delta x^2}{8}(\partial^2_{xx}\widetilde{u})_i + \mathcal{O}(\varepsilon), \tag{46a}$$

$$\textit{(FD2):}\quad (\partial_t \widetilde{Y})_i = -\widetilde{u}_i + \mathcal{O}(\varepsilon). \tag{46b}$$

On one hand, according to Equation (45b), 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 (46a) with respect to time, we then obtain:

$$(\partial^2_{tt}\widetilde{Y})_i = a^2(\partial^2_{xx}\widetilde{Y})_i + \frac{a^2\Delta x^2}{8}(\partial^4_{xxxx}\widetilde{Y})_i + \mathcal{O}(\varepsilon). \tag{47}$$

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 Δx. 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 B. The amplification factors obtained from both discretizations of the advective operator $A(Y)$ are presented in Fig. 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 displayed in Fig. 8b. The next section is thus dedicated to the construction of an ELP finite-volume scheme in order to get closer to the asymptotic behavior given naturally by the finite difference framework.

3.4.3. An ELP FV2 scheme

It is useful to enforce the ELP property in order to enhance the damping of high frequencies, and therefore prevent the propagation of these oscillations. We do that comparing the truncation errors of the *FD2* and *FV2* discretization schemes, it is possible to highlight the dispersive term involved in Equation (47).

Under the assumption $u > 0$, the two discretization schemes for the advective operator are defined by Equations (42) and (43). The truncation errors e_{FD2} and e_{FV2}, obtained from *FD2* and *FV2* schemes, can be respectively written as:

$$\left(e_{FD2}\right)_i = \left[-\frac{1}{12}u_i(\partial^3_{xxx}Y)_i\right]\Delta x^2 + \mathcal{O}(\Delta x^3), \tag{48a}$$

$$\left(e_{FV2}\right)_i = \left[-\frac{1}{12}u_i(\partial^3_{xxx}Y)_i + \frac{1}{8}(\partial^2_{xx}u)_i(\partial_x Y)_i\right]\Delta x^2 + \mathcal{O}(\Delta x^3). \tag{48b}$$

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

$$\left(S_{FV2}\right)_i = \left(e_{FV2}\right)_i - \left(e_{FD2}\right)_i + \mathcal{O}(\Delta x^3) = \frac{\Delta x^2}{8}(\partial^2_{xx}u)_i(\partial_x Y)_i. \tag{49}$$

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

$$(\partial_t Y)_i + (\widetilde{A}_{FV2}(Y))_i = (S_{FV2})_i. \tag{50}$$

A consistent discretization of the source term (49) for the advection of backward characteristics helps 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 Ω_{ij} is h. Under the assumption $u > 0$, the second order *FD2* and *FV2* schemes in two dimensions can be respectively written as:

$$\begin{aligned}\big(\mathbf{u}\cdot\nabla Y\big)_{i,j} \approx \big(\tilde{A}^{2D}_{FD2}(Y)\big)_{i,j} = &\frac{(\mathbf{u}_x)_{i,j}}{4h}(Y_{i+1,j}+3Y_{i,j}-5Y_{i-1,j}+Y_{i-2,j})\\ &+\frac{(\mathbf{u}_y)_{i,j}}{4h}(Y_{i,j+1}+3Y_{i,j}-5Y_{i,j-1}+Y_{i,j-2}),\end{aligned} \tag{51}$$

and:

$$\begin{aligned}\big(\nabla\cdot(\mathbf{u}Y)\big)_{i,j} &\approx \big(\tilde{A}^{2D}_{FV2}(Y)\big)_{i,j}\\ &= \frac{1}{h}\Big[(\mathbf{u}_x)_{i+\frac{1}{2},j}\big(Y_{i,j}+\frac{1}{4}(Y_{i+1,j}-Y_{i-1,j})\big) - (\mathbf{u}_x)_{i-\frac{1}{2},j}\big(Y_{i-1,j}+\frac{1}{4}(Y_{i,j}-Y_{i-2,j})\big)\\ &+(\mathbf{u}_y)_{i,j+\frac{1}{2}}\big(Y_{i,j}+\frac{1}{4}(Y_{i,j+1}-Y_{i,j-1})\big) - (\mathbf{u}_y)_{i,j-\frac{1}{2}}\big(Y_{i,j-1}+\frac{1}{4}(Y_{i,j}-Y_{i,j-2})\big)\Big],\end{aligned} \tag{52}$$

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 (51) and (52), are:

$$\big(e^{2D}_{FD2}\big)_{i,j} = -\frac{h^2}{12}\Big((\mathbf{u}_x)_{i,j}(\partial^3_{xxx}Y)_{i,j} + (\mathbf{u}_y)_{i,j}(\partial^3_{yyy}Y)_{i,j}\Big) + \mathcal{O}(h^3), \tag{53}$$

and:

$$\begin{aligned}\big(e^{2D}_{FV2}\big)_{i,j} = \big(e^{2D}_{FD2}\big)_{i,j} &+ \underbrace{\frac{h^2}{8}\Big((\partial^2_{xx}\mathbf{u}_x)_{i,j}(\partial_x Y)_{i,j} + (\partial^2_{yy}\mathbf{u}_y)_{i,j}(\partial_y Y)_{i,j}\Big)}_{(S_1)_{i,j}}\\ &+ \underbrace{\frac{h^2}{24}Y_{i,j}\Big((\partial^3_{xxx}\mathbf{u}_x)_{i,j} + (\partial^3_{yyy}\mathbf{u}_y)_{i,j}\Big)}_{(S_2)_{i,j}} + \mathcal{O}(h^3).\end{aligned} \tag{54}$$

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 (48) and (49)). It can be proved that the last term S_2 of Equation (54) 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

$$\frac{\partial \mathbf{Y}}{\partial t} + \nabla\cdot(\mathbf{u}\otimes\mathbf{Y}) = \mathbf{S}^{2D}_{FV2}, \tag{55}$$

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

$$\mathbf{S}^{2D}_{FV2} = \frac{h^2}{8}(\partial^2_{xx}\mathbf{u}_x, \partial^2_{yy}\mathbf{u}_y)^T[\nabla\mathbf{Y}] =: \frac{h^2}{8}(\nabla\cdot[\nabla^{diag}\mathbf{u}])[\nabla\mathbf{Y}]. \tag{56}$$

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.

3.5. Implementation of the method

To summarize this section, the numerical implementation of the method is detailed in the Algorithm 1. It provides a list and classification of the different steps.

The C++/MPI calculation code is based on the Bitpit/PABLO library for the generation and handling of AMR meshes. All functions and routines have been implemented using most of the built-in and optimized functions of this library in order to benefit from the quadtree vector data structure. The resolution of all linear systems is achieved using the BiCGStab solver provided by PETSc[2] library. To speed up the convergence of the iterative BiCGStab method, a restricted additive Schwarz method (PCASM) is used as a preconditioner.

Algorithm 1: Detailed implementation of the numerical method.

```
Data: Set up of the simulation test case
→ definition of the involved geometries;
→ definition of the adapted mesh;
→ initialization of variables;
while $t < T_{max}$, do
    → definition of the time step $\Delta t$ with respect to the CFL condition (see Section 3.1);
    → advection of backward characteristics and level-set $\phi$ functions Y and $\phi$ (Eq. (23)): TVD-RK2 scheme for the time discretization, FV2 scheme
     in space (see Section 3.2.3). For Y, the inhomogeneous advection equation (55) with source term (56) is considered;
    → eventual update of Y and $\phi$, according to a given frequency (see Section 3.3): re-initialization of $\phi$;
    and linear extrapolation of Y inside the fluid;
    → prediction step (Eq. (24)): Gear scheme in time, finite volume schemes in space (Section 3.2.3);
    → computation of face-center normal velocity $U_{fc}$ (see Eqs. (28) and (31));
    → projection step: resolution of the Poisson equation (29);
    → correction step (Eq. (30));
    → update of variables;
```

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 bio-medical 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.

In all simulations, the time step varies over time. To respect the CFL condition described in Section 3.1, the time step is defined as:

$$\Delta t := 0.25 \, \frac{h_{min}}{||\mathbf{u}||_{L^\infty} + c_{wave}} \tag{57}$$

where h_{min} is the minimum characteristic length of the mesh, $||\mathbf{u}||_{L^\infty}$ is the maximum convection speed and $c_{wave} = \sqrt{3G/\rho}$ is the speed of linear elastic waves.

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

We first study the deformation of an elastic structure immersed in a lid-driven cavity flow. This test has been studied in the literature using both Lagrangian and Eulerian formalisms [13,47,57]. 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 0.4 m. The set-up is shown in Fig. 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.

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 ϕ and the backward characteristics **Y** are updated every 0.01 s, 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 Fig. 11 for the soft material. The interface deformation, for both soft and stiff materials, is presented in Fig. 12. The interface evolution is in good agreements with reference results obtained by Zhao et al. [57].

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

[2] https://www.mcs.anl.gov/petsc/.

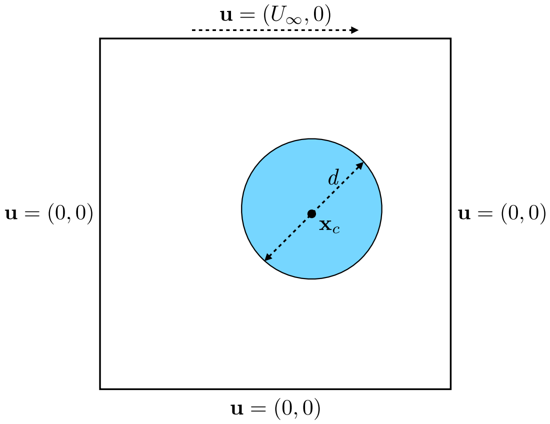


Fig. 10. Deformable solid in a lid-driven cavity flow: fluid-structure set-up.

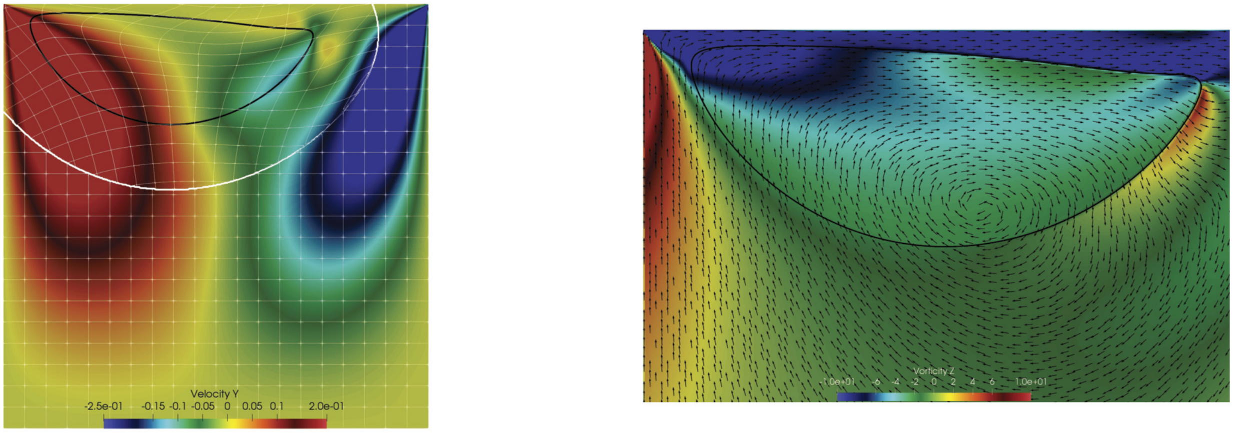


Fig. 11. Deformable soft solid ($G = 0.1$ Pa) in a lid-driven cavity flow: snapshots at time $t = 4.69$ s. Left: y-component of the velocity; right: z-component of the vorticity with velocity field. The black and white lines represent the interface Γ_e and the contours of the backward characteristics **Y** respectively. The discontinuity of the characteristics is due to the extrapolation process which is only performed up to a certain distance from the interface.

$$\mathbf{x}_c(t) \approx \frac{\sum_{i=1}^{N_{cells}} \widetilde{\Omega_i^e}(t)\,\mathbf{x}_i}{\sum_{i=1}^{N_{cells}} \widetilde{\Omega_i^e}(t)}, \tag{58}$$

where $\widetilde{\Omega_i^e}$ is an approximation of the volume fraction of solid in cell Ω_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. [47] and Deborde [13].

Finally, Fig. 14 presents the temporal evolution of the material volume

$$|\Omega_e(t)| \approx \sum_{i=1}^{N_{cells}} \widetilde{\Omega_i^e}(t). \tag{59}$$

These tests are performed to quantify the mass-conservation error usually observed [46] when dealing with level-set methods. For both rigidities, the relative loss of volume remains below 0.02%. As expected, using interface regularization, this error decreases with first order accuracy.

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.

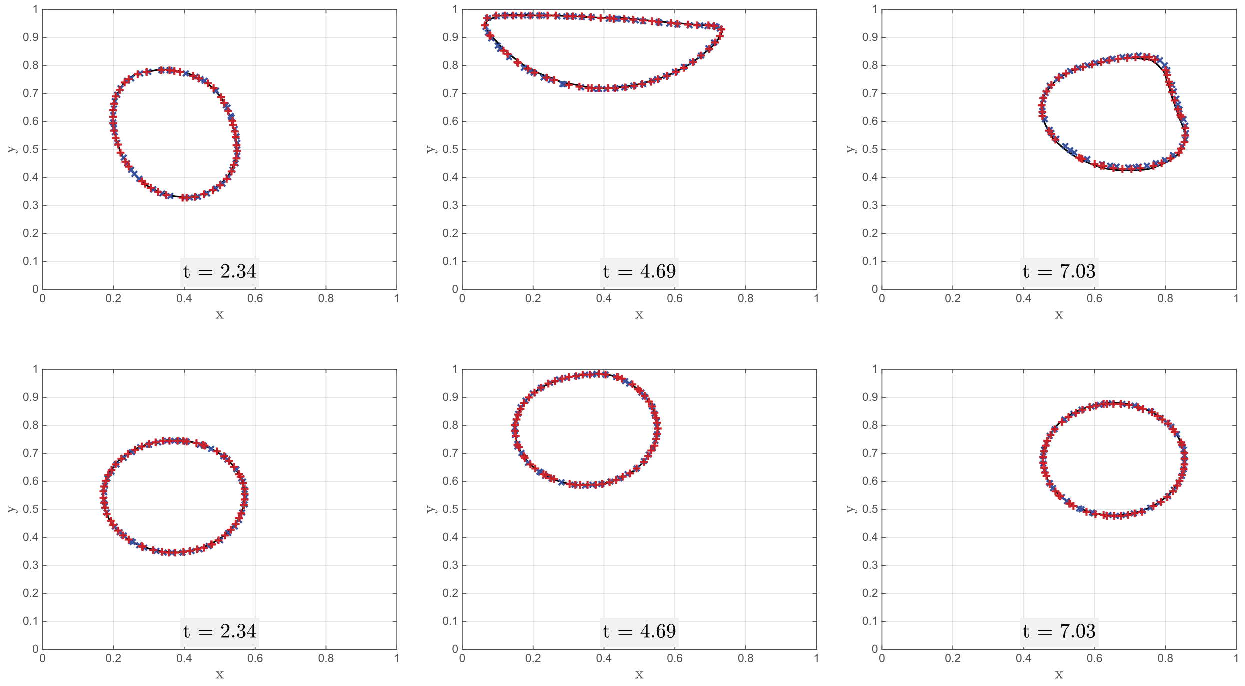


Fig. 12. Deformation of soft (top range) and stiff (bottom range) solids immersed in a lid-driven cavity flow: interface location Γ_e for various time instants, comparison with the work of Zhao [57] (black line). The blue-plus and red-cross markers refer to the results obtained on uniform L_9 and L_{10} grids respectively.

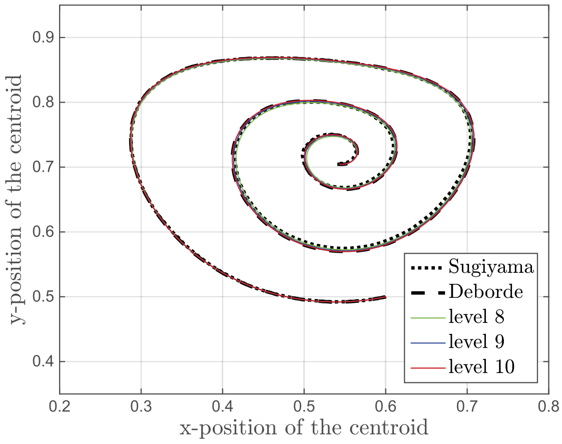


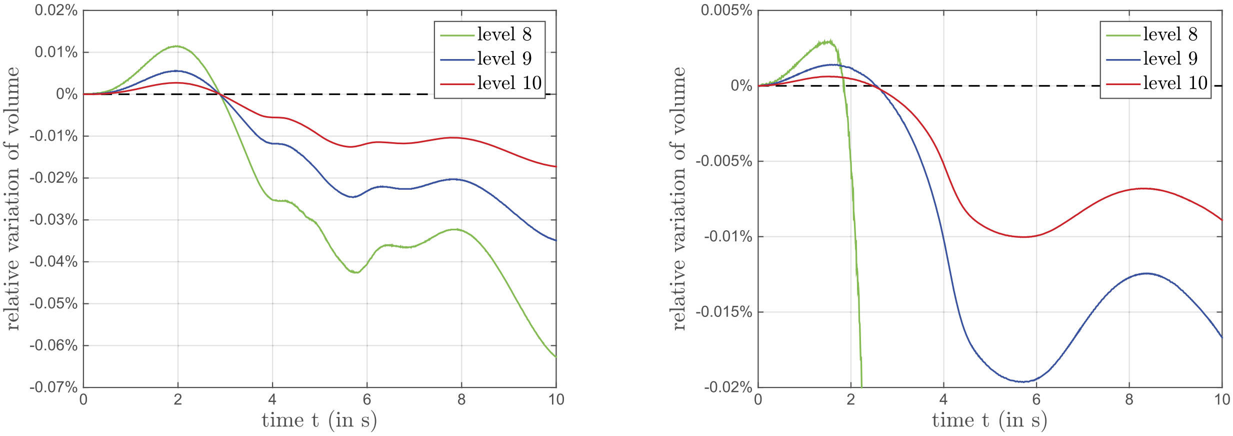
Fig. 13. Deformable soft solid ($G = 0.1$ Pa) in a lid-driven cavity flow: approximated centroid position over time until $t = 20$ s. The simulations are run on L_8 to L_{10} uniform grids. The centroid position is computed following Eq. (58).

An elastic cylinder is immersed in a cavity $\Omega = [0, 1]^2$ filled with viscous fluid as depicted in Fig. 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 $G = 1$ Pa.

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

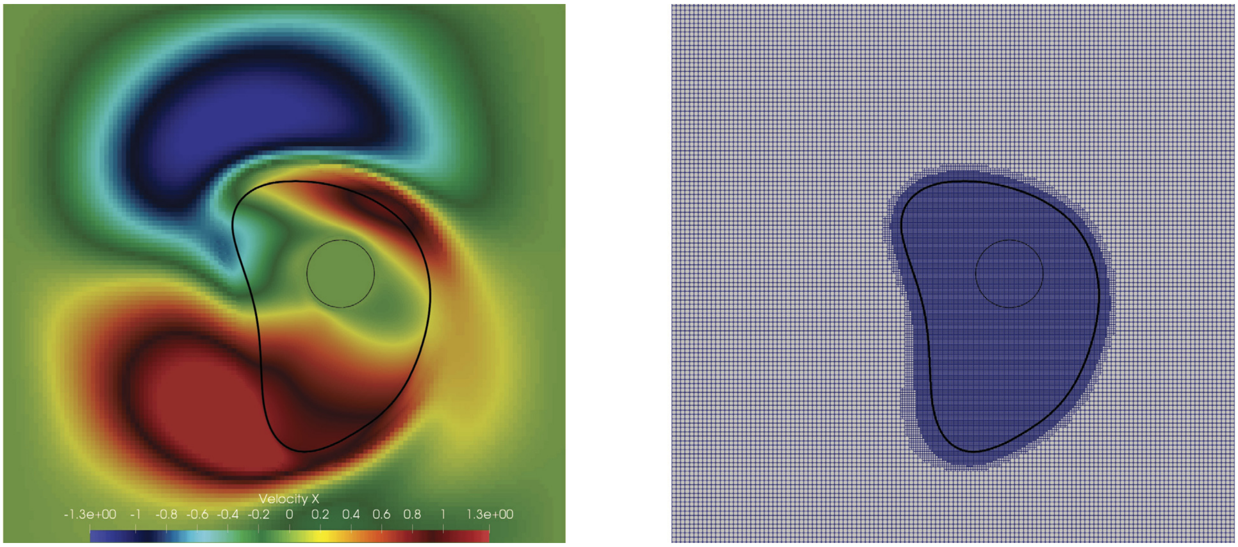
$$\mathbf{u}_s(t) = \begin{cases} \pi r_s t\big(-\sin(\pi t^2/2), \cos(\pi t^2/2)\big)^T & \text{if } t < 2s, \\ 2\pi r_s\big(-\sin(2\pi t), \cos(2\pi t)\big)^T & \text{it } t \geq 2s. \end{cases}$$

A transient velocity of the holder is imposed until $t = 2$ s to avoid heavy compression effects.



	level	8	9	10
soft material ($G = 0.1$)	maximum MCE rate	0.06268% /	0.03489 % 0.845	0.01727% 1.015
stiff material ($G = 10$)	maximum MCE rate	0.04188% /	0.01964% 1.092	0.01003% 0.969

Fig. 14. Two deformable solids in a lid-driven cavity flow: estimation of the relative mass-conservation error over time. At the top: time evolution of the relative variation of volume for both soft (left, $G = 0.1$) and stiff (right, $G = 10$) materials. At the bottom: convergence study on the maximum mass-conservation error (MCE) for both materials. The simulations are run on L_8 to L_{10} uniform grids. The volume of the structure is computed following Eq. (59).



(a) *x-component of the velocity*

(b) *dynamic AMR $L_7 - L_9$ grid*

Fig. 15. Solid deformation due to an immersed actuator at time $t = 5$. The thin and thick black lines represent the interfaces Γ_s and Γ_e respectively.

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

For fair comparison between uniform and quadtree AMR grids, a same mesh resolution is imposed close to the interface Γ_e. To this end, a geometrical criterion based on the level-set function ϕ 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 0.05 s to precisely follow the structure over time. To define the new grid data from the previous mesh, the solution is linearly reconstructed (following Eq. (36) in

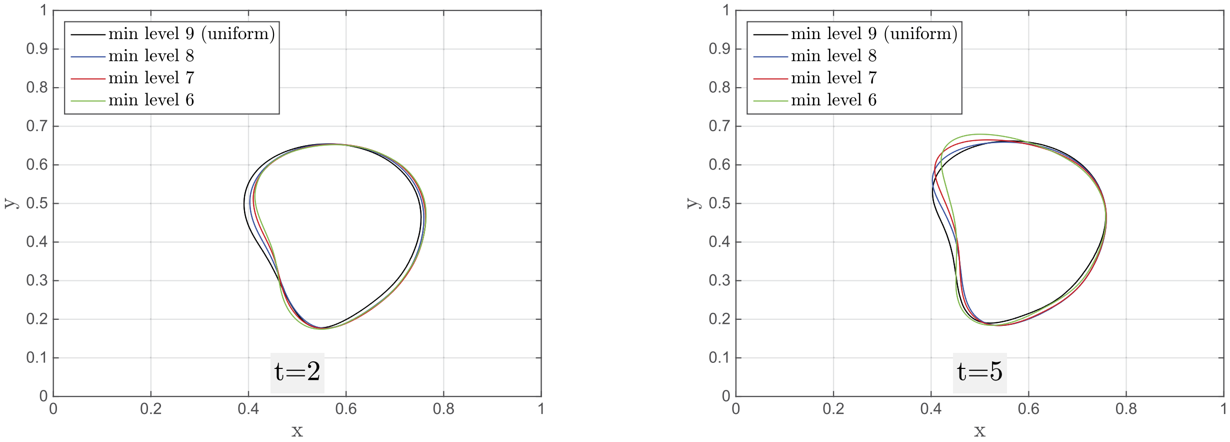


Fig. 16. Solid deformation due to an immersed actuator: position of the interface $\Gamma_e(t)$ at time $t = 2$ (left) and $t = 5$ (right). The comparison is performed between the L_9 uniform grid and dynamic AMR grids for a same maximum level of refinement L_{max}.

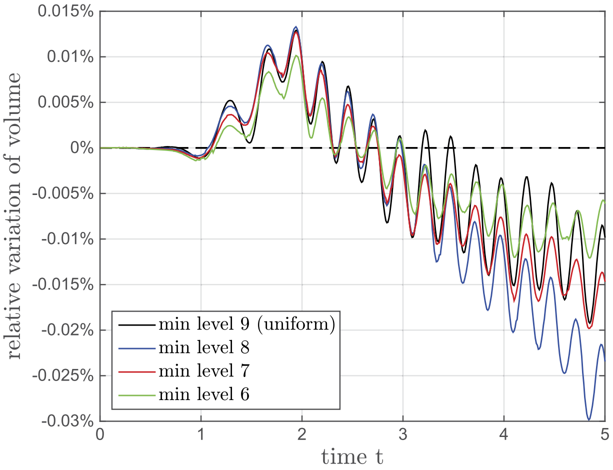


Fig. 17. Solid deformation due to an immersed actuator: estimation of the relative mass-conservation error over time. The comparison is performed between the L_9 uniform grid and dynamic AMR grids for a same maximum level of refinement L_{max}.

Section 3.2.3) according to the parent/children inheritance. Finally, the level-set function and the backward characteristics are updated 100 times per period $T = 1$ s.

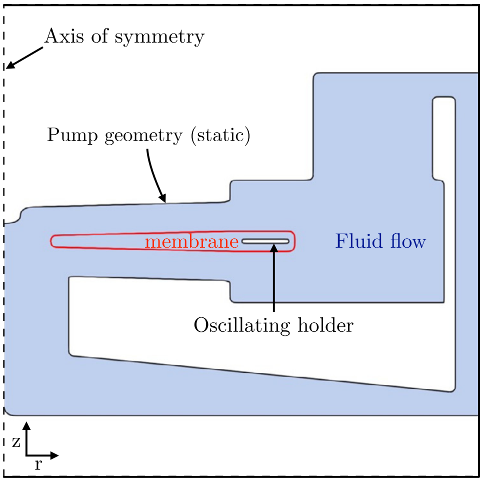
In what follows, we investigate how the accuracy of the numerical solution is impacted by coarsening the grid in the fluid outside the narrow band. We compare the interface position for different numbers of coarsening ($L_{max} - L_{min}$), ranging from 1 to 3. The positions of Γ_e at time $t = 2$ s and $t = 5$ s are presented in Fig. 16. Despite the large difference with regard to the number of grid points, we obtain similar deformations. Moreover, as shown in Fig. 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.

4.3. Oscillating membrane immersed in a cardiac pump geometry

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

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.

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%



(a) *spatial set-up.*

Parameters		Value
Membrane	G	1 MPa
	ρ_e	$\rho_f = 1.17\times10^3$ kg/m^3
	μ_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.*

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

$c_{wave}/c_{max,holder} \approx 168$, 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.

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 Fig. 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 ϕ is used to impose a maximum refinement (level L_{max}) inside the membrane and in the vicinity of interfaces Γ_s and Γ_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 η_{min} and η_{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 Fig. 19 after 2, 3, 4 and 5 periods of oscillation. Fig. 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. A more extensive range of accuracy tests for the validation of this numerical method can be found in [21]. By considering only

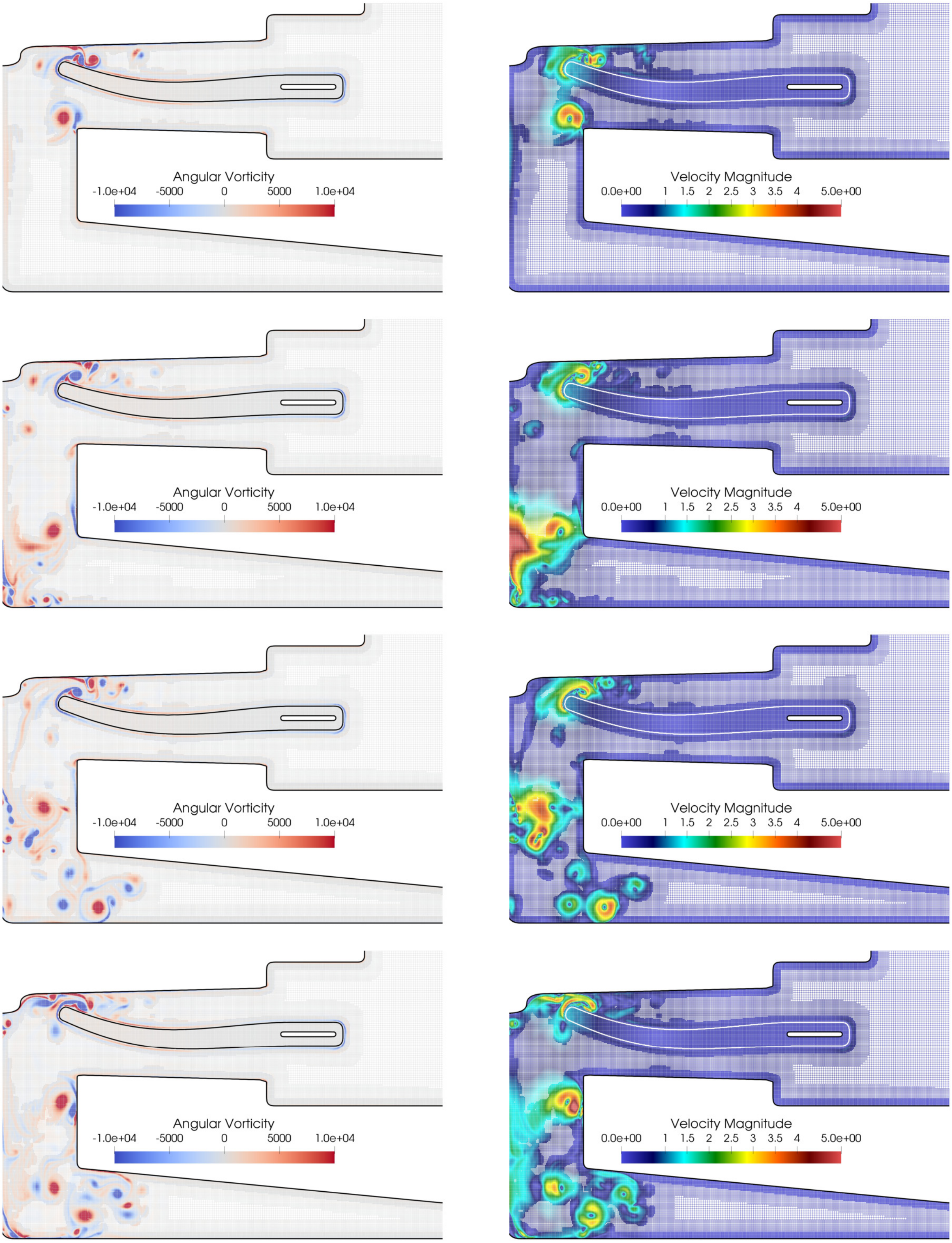
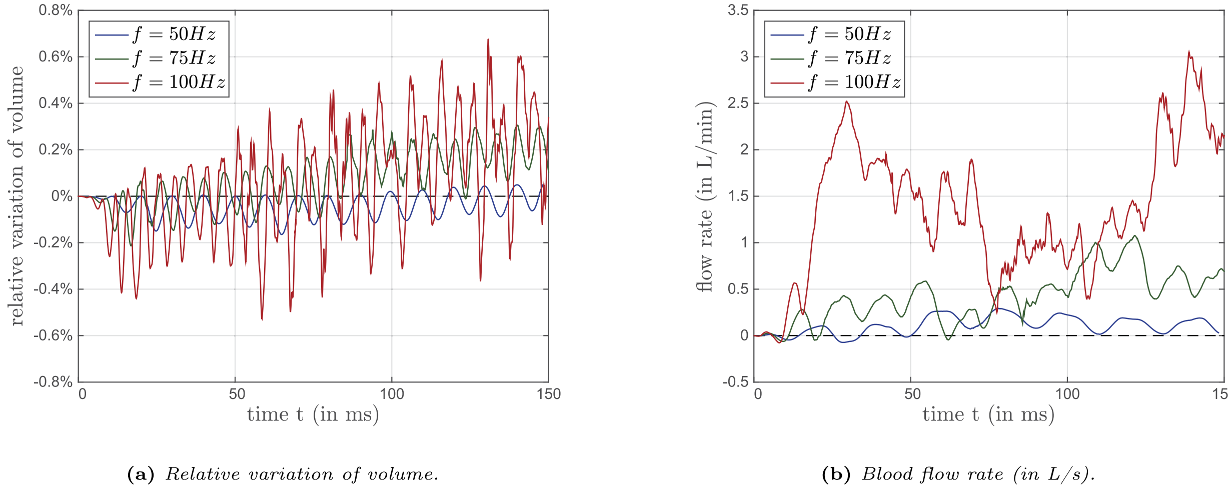


Fig. 19. Oscillating membrane immersed in a cardiac pump geometry: angular component of the vorticity (on the left) and velocity magnitude (on the right). From top to bottom: solution at time $t = 2T$, $3T$, $4T$ and $5T$. The AMR mesh is displayed in the background.



(a) *Relative variation of volume.*

(b) *Blood flow rate (in L/s).*

Fig. 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.

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. [26].

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 2D axi-symmetric LVAD geometry is finally proposed as a bio-medical application. In particular, this numerical experiment shows the ability to reproduce a pumping flow thanks to the deformation of an oscillating rubber-like membrane. The generalization to 3D would not present major difficulties concerning the discretizations in space, as done for different models in [38,49]. However, the development of an implicit formulation, as previously mentioned, will be the next development in order to make the complete FSI simulation practicable from a computational time point of view.

CRediT authorship contribution statement

Michel Bergmann: Conceptualization, Methodology, Supervision, Writing – review & editing. **Antoine Fondanèche:** Conceptualization, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **Angelo Iollo:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

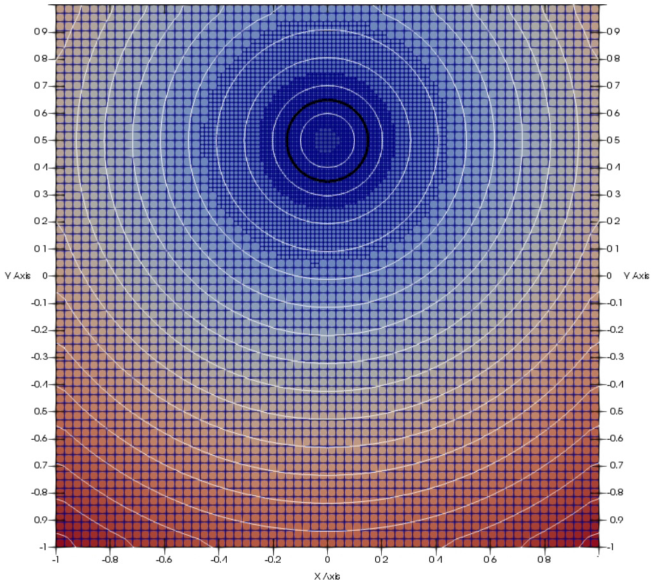


Fig. 21. Advection of a circle: contours of the level-set function (the interface is drawn in black) and the quadtree mesh $L_6 - L_8$ at time $t = T_{max}$.

Acknowledgements

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 Skłodowska-Curie Actions, grant agreement 872442 (ARIA).

Appendix A. Resolution of some transport problems

This section is dedicated to the validation of the finite volume discretization of the advection operator. We propose two different test cases including level-set functions. Some convergence studies are presented and a result comparison obtained on fully uniform and adapted quadtree meshes is provided in order to highlight the gain in computational time.

A.1. Advection of a circle

A circle of center $(0, 1/2)$ and radius 0.15 is transported in a domain $[-1, 1]^2$ via the rotating velocity field $\mathbf{u} = 2\pi(y, -x)^T$. During the simulation, the circle makes 3 complete rotations by choosing $T_{max} = 3$. As shown in Fig. 21, the simulations are performed on quadtree meshes. The dynamic mesh adaptation is performed according to a time frequency $f = 0.01$ using the value of the signed distance function.

The simulations are performed on both uniform and quadtree meshes, for a same maximum mesh refinement L_{max}. Table 2 presents the results obtained, namely the error in the sense of the L_∞ norm, the relative loss of volume and the computational time. For each of the quantities reported in the table, the ratio corresponds to the ratio between the quantity obtained using the uniform mesh and the one obtained using the quadtree mesh. For a computation which is about 7 times faster, the simulation carried out on the AMR mesh provides results that are comparable to those obtained on the corresponding uniform mesh.

A.2. The serpentine

This level-set test case is an efficient way to quantify the performance of an interface-capturing method. During the simulation, the interface stretches until it becomes a long filament, subjected to a large shear flow. In the domain $\Omega = [0, 1]^2$, the interface is first defined as a circle centered at $\mathbf{x}_0 = (1/2, 3/4)$ and of radius $R = 0.15$. The induced level-set function is deformed using a solenoidal velocity field

$$\mathbf{u}(t, \mathbf{x}) = \sin\left(\frac{2\pi t}{T_{max}}\right)\Big(-\sin^2(\pi x)\sin(2\pi y), \sin^2(\pi y)\sin(2\pi x)\Big)^T,$$

Table 2
Advection of a circle: comparison in terms of accuracy, relative variation of mass (RVM) and computational time (in s) obtained on uniform and quadtree meshes (with $L_{max} - L_{min} = 2$ level jumps) for a same maximum level L_{max}. The simulations were performed with 24 CPUs.

	L_{max} h_{min}	8 7.813E-03	9 3.906E-03	10 1.953E-03	11 9.766E-04
	L_∞ error	1.592E-04	1.959E-05	2.452E-06	3.072E-07
uniform	RVM	1.195E-03	1.458E-04	1.835E-05	2.299E-06
	time	631.32	4936.48	41499.43	370285.71
	L_∞ error	1.799E-04	2.167E-05	2.585E-06	3.280E-07
quadtree	RVM	1.302E-03	1.517E-04	1.879E-05	2.354E-06
	time	126.39	814.88	6142.61	51245.60
Comparison	L_∞ error	+ **12.51%**	+ **9.61%**	+ **5.15%**	+ **6.34%**
quadtree VS	RVM	+ **8.20%**	+ **3.87%**	+ **2.36%**	+ **2.30%**
uniform	time	- **80.0%**	- **80.2%**	- **85.2%**	- **86.2%**

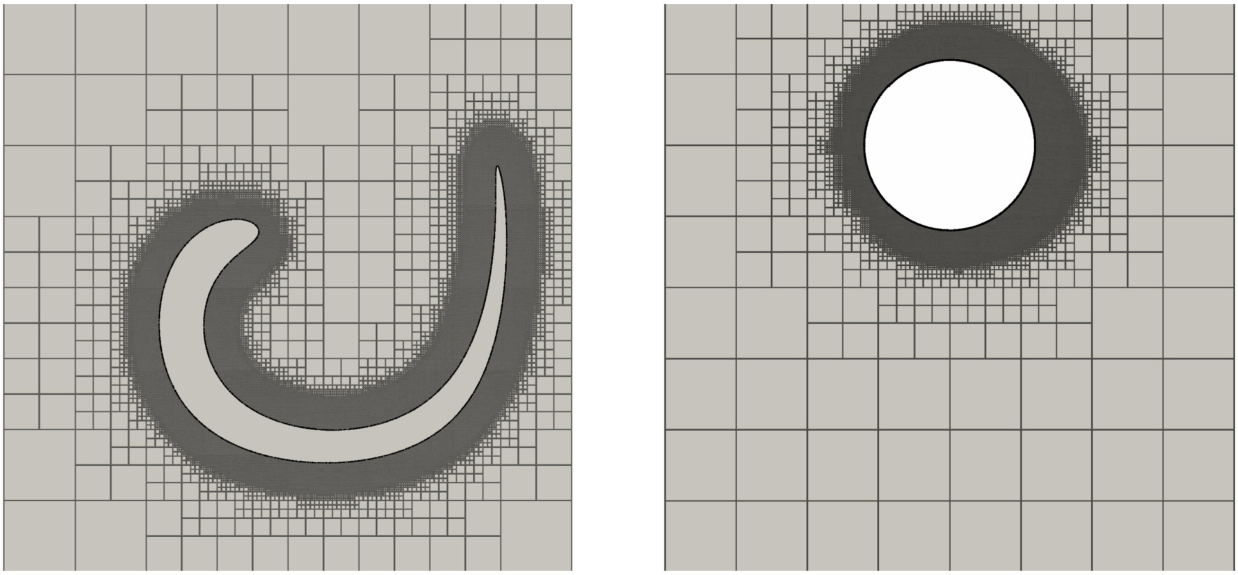


Fig. 22. The serpentine: position of the interface at times $t = T_{max}/2$ (left) and $t = T_{max}$ (right). The simulation was run on a dynamic quadtree mesh with $L_{max} = 10$.

Table 3
The serpentine: L_1 and L_∞ errors and relative variation of volume (RVV) at $T_{max} = 3$.

L_{max} h_{min}	8 7.813E-03	9 3.906E-03	10 1.953E-03	11 9.766E-04
L_1 error	3.434E-04	4.916E-05	1.123E-05	2.094E-06
q_1	/	2.804	2.130	2.423
L_∞ error	5.731E-03	7.905E-04	2.310E-04	5.658E-05
q_∞	/	2.858	1.775	2.029
RVV	0.0365%	0.0242%	0.0098%	0.0021%
q_{RVV}	/	0.591	1.309	2.253

up to time $T_{max} = 3$. Fig. 22 depicts the solution at time instants $t = T_{max}/2$ and $t = T_{max}$. The mesh inside the serpentine does exist but it is hidden for visualization purposes.

The simulations are performed on quadtree meshes with a dynamic mesh adaptation. The re-initialization procedure is triggered according to a frequency $f = 0.05$. Table 3 shows that the scheme is second order accurate in the L_1 and L_∞ sense, as well as on the relative variation of volume.

Appendix B. Asymptotic analysis of the 1D discretization

This appendix details the Fourier analysis stated in Section 3.4.2. This analysis is intended to demonstrate how the space discretization scheme to transport backward characteristics can affect the stability of the numerical method for the resolution of the 1D model (38). The results obtained, namely the amplification coefficients, have been displayed in Fig. 9.

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 $\theta := 2\pi\xi\Delta x \in \mathbb{R}$:

$$\begin{aligned} \widetilde{u}_j^n &= \mathcal{U}(\theta)^n e^{ij\theta}, \\ \widetilde{Y}_j^n &= \mathcal{Y}(\theta)^n e^{ij\theta}. \end{aligned} \tag{60}$$

We recall here the different discretizations employed in Section 3.4. The 1D model (38) is integrated in time using a first-order linear multi-step method:

$$\frac{Y_j^{n+1} - Y_j^n}{\Delta t} = \mathcal{G}_j(u^{n+1}, Y^n), \quad \text{with } u^{n+1} = 2u^n - u^{n-1}, \tag{61a}$$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \mathcal{F}_j(u^n, Y^{n+1}). \tag{61b}$$

The linearized discrete formulations of the *FV2* and *FD2* schemes can be written for equation (61a) respectively as:

$$\frac{\widetilde{Y}_j^{n+1} - \widetilde{Y}_j^n}{\Delta t} = \begin{cases} -\dfrac{1}{2}\widetilde{u}_j - \dfrac{1}{4}\left(\widetilde{u}_{j+1} + \widetilde{u}_{j-1}\right) & \text{(FV2 scheme)}, \\ -\widetilde{u}_j & \text{(FD2 scheme)}, \end{cases} \tag{62}$$

in the limit of small deformations $\varepsilon \to 0$. The linearization of Equation (61b) gives:

$$\frac{\widetilde{u}_j^{n+1} - \widetilde{u}_j^n}{\Delta t} = -2G\frac{\Delta t}{\Delta x^2}(\widetilde{Y}_{j+1} - 2\widetilde{Y}_j + \widetilde{Y}_{j-1}). \tag{63}$$

We now apply the Fourier decomposition (60) to the linearized discretization of the model. Consequently, Equations (62) and (63) become respectively:

$$\mathcal{Y}^{n+1} = \begin{cases} \mathcal{Y}^n - 2\Delta t\ \cos^2(\theta/2)\,\mathcal{U}^n + \Delta t\ \cos^2(\theta/2)\,\mathcal{U}^{n-1} & \text{(FV2 scheme)}, \\ \mathcal{Y}^n - 2\Delta t\,\mathcal{U}^n + \Delta t\,\mathcal{U}^{n-1} & \text{(FD2 scheme)}, \end{cases} \tag{64}$$

and:

$$\mathcal{U}^{n+1} = \begin{cases} 4\dfrac{C^2}{\Delta t}\ \sin^2(\theta/2)\,\mathcal{Y}^n + \left(1 - 2C^2\sin^2(\theta)\right)\mathcal{U}^n + C^2\sin^2(\theta)\,\mathcal{U}^{n-1} & \text{(FV2 scheme)}, \\ 4\dfrac{C^2}{\Delta t}\sin^2(\theta/2)\,\mathcal{Y}^n + \left(1 - 8C^2\sin^2(\theta/2)\right)\mathcal{U}^n + 4C^2\sin^2(\theta/2)\,\mathcal{U}^{n-1} & \text{(FD2 scheme)}, \end{cases} \tag{65}$$

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 (64) and (65) equations can be expressed using the following matrix form:

$$\begin{pmatrix} \mathcal{Y}^{n+1} \\ \mathcal{U}^{n+1} \\ \mathcal{U}^n \end{pmatrix} = \mathcal{A} \begin{pmatrix} \mathcal{Y}^n \\ \mathcal{U}^n \\ \mathcal{U}^{n-1} \end{pmatrix}, \tag{66}$$

where $\mathcal{A} \equiv \mathcal{A}(\theta)$ represents the amplification matrix. With respect to Equations (64) and (65), the amplification matrices obtained using the *FV2* and *FD2* schemes are respectively:

$$\mathcal{A}_{FV2} = \begin{pmatrix} 1 & -2\Delta t\cos^2(\theta/2) & \Delta t\cos^2(\theta/2) \\ 4\frac{C^2}{\Delta t}\ \sin^2(\theta/2) & 1 - 2C^2\sin^2(\theta) & C^2\sin^2(\theta) \\ 0 & 1 & 0 \end{pmatrix}, \tag{67}$$

and

$$\mathcal{A}_{FD2} = \begin{pmatrix} 1 & -2\Delta t & \Delta t \\ 4\frac{C^2}{\Delta t}\sin^2(\theta/2) & 1-8C^2\sin^2(\theta/2) & 4C^2\sin^2(\theta/2) \\ 0 & 1 & 0 \end{pmatrix}. \tag{68}$$

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 amplification factor $\rho(\mathcal{A})$ verifies:

$$\max_p\{|\lambda_p|\} \leq 1, \tag{69}$$

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

References

[1] Shahnawaz Ahmed, Stanley Bak, Joyce McLaughlin, Daniel Renzi, A third order accurate fast marching method for the Eikonal equation in two dimensions, SIAM J. Sci. Comput. 33 (5) (2011) 2402–2420.
[2] Frédéric Alauzet, A changing-topology moving mesh technique for large displacements, Eng. Comput. 30 (2) (2014) 175–200.
[3] Philippe Angot, Charles-Henri Bruneau, Pierre Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numer. Math. 81 (4) (1999) 497–520.
[4] Tariq D. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (1) (2004) 349–355.
[5] Frank P.T. Baaijens, A fictitious domain/mortar element method for fluid–structure interaction, Int. J. Numer. Methods Fluids 35 (7) (2001) 743–761.
[6] Michel Bergmann, Angelo Iollo, Modeling and simulation of fish-like swimming, J. Comput. Phys. 230 (2) (2011) 329–348.
[7] Michel Bergmann, Jessica Hovnanian, Angelo Iollo, An accurate Cartesian method for incompressible flows with moving boundaries, Commun. Comput. Phys. 15 (5) (2014) 1266–1290.
[8] Michel Bergmann, Angelo Iollo, Rajat Mittal, Effect of caudal fin flexibility on the propulsive efficiency of a fish-like swimmer, Bioinspir. Biomim. 9 (4) (2014) 046001.
[9] Ludovic Boilevin-Kayl, Miguel A. Fernández, Jean-Frédéric Gerbeau, Numerical methods for immersed fsi with thin-walled structures, Comput. Fluids 179 (2019) 744–763.
[10] Alexandre Joel Chorin, Numerical solution of the Navier-Stokes equations, Math. Comput. 22 (104) (1968) 745–762.
[11] Georges-Henri Cottet, Emmanuel Maitre, Thomas Milcent, Eulerian formulation and level set models for incompressible fluid-structure interaction, ESAIM: Math. Model. Numer. Anal. 42 (3) (2008) 471–492.
[12] Yves Coudière, Jean-Paul Vila, Philippe Villedieu, Convergence rate of a finite volume scheme for a two dimensional convection-diffusion problem, ESAIM: Math. Model. Numer. Anal. 33 (3) (1999) 493–516.
[13] Julien Deborde, Modélisation et simulation de l'interaction fluide-structure élastique: application à l'atténuation des vagues, PhD thesis, Bordeaux, 2017.
[14] Julien Deborde, Thomas Milcent, Pierre Lubin, Stéphane Glockner, Numerical simulations of the interaction of solitary waves and elastic structures with a fully Eulerian method, Water Waves 2 (2020) 433–466.
[15] Sarah Delcourte, Komla Domelevo, Pascal Omnes, Discrete Duality Finite Volume Method for Second Order Elliptic Problems, Hermes Science Publishing, 2005, pp. 447–458.
[16] Antoine du Chéné, Chohong Min, Frédéric Gibou, Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes, J. Sci. Comput. 35 (2) (2008) 114–131.
[17] Thomas Dunne, Rolf Rannacher, Adaptive finite element approximation of fluid-structure interaction based on an Eulerian variational formulation, in: Fluid-Structure Interaction, Springer, 2006, pp. 110–145.
[18] Charbel Farhat, Vinod K. Lakshminarayan, An ale formulation of embedded boundary methods for tracking boundary layers in turbulent fluid–structure interaction problems, J. Comput. Phys. 263 (2014) 53–70.
[19] Charbel Farhat, Michael Lesoinne, Patrick Le Tallec, Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity, Comput. Methods Appl. Mech. Eng. 157 (1–2) (1998) 95–114.
[20] Joel H. Ferziger, Milovan Peric, Computational Methods for Fluid Dynamics, Springer Science & Business Media, 2012.
[21] Antoine Fondaneche, Interaction fluide-structure dans un dispositif vasculaire actif, PhD thesis, Université de Bordeaux, 2021.
[22] Yong-Bi Fu, Ray W. Ogden, Nonlinear Elasticity: Theory and Applications, 2001.
[23] Roland Glowinski, Tsorng-Whay Pan, Todd I. Hesla, Daniel D. Joseph, Jácques Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, J. Comput. Phys. 169 (2) (2001) 363–426.
[24] Katuhiko Goda, A multistep technique with implicit difference schemes for calculating two-or three-dimensional cavity flows, J. Comput. Phys. 30 (1) (1979) 76–95.
[25] Arthur Guittet, Maxime Theillard, Frédéric Gibou, A stable projection method for the incompressible Navier–Stokes equations on arbitrary geometries and adaptive quad/octrees, J. Comput. Phys. 292 (2015) 215–238.
[26] S. Ii, K. Sugiyama, S. Takeuchi, S. Takagi, Yoichiro Matsumoto, An implicit full Eulerian method for the fluid–structure interaction problem, Int. J. Numer. Methods Fluids 65 (1–3) (2011) 150–165.
[27] Benjamin Kadoch, Dmitry Kolomenskiy, Philippe Angot, Kai Schneider, A volume penalization method for incompressible flows and scalar advection–diffusion with moving obstacles, J. Comput. Phys. 231 (12) (2012) 4365–4383.
[28] Francky Luddens, Michel Bergmann, Lisl Weynans, Enablers for high-order level set methods in fluid mechanics, Int. J. Numer. Methods Fluids 79 (12) (2015) 654–675.
[29] Gilles Marckmann, Erwan Verron, Comparison of hyperelastic models for rubber-like materials, Rubber Chem. Technol. 79 (5) (2006) 835–858.
[30] Rajat Mittal, Haibo Dong, Meliha Bozkurttas, F.M. Najjar, Abel Vargas, Alfred Von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys. 227 (10) (2008) 4825–4852.
[31] Melvin Mooney, A theory of large elastic deformation, J. Appl. Phys. 11 (9) (1940) 582–592.
[32] Guy M. Morton, A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing, 1966.
[33] Chanwut Nitikitpaiboon, Klaus-Jürgen Bathe, An arbitrary Lagrangian-Eulerian velocity potential formulation for fluid-structure interaction, Comput. Struct. 47 (4–5) (1993) 871–891.
[34] Ray W. Ogden, Nonlinear elasticity, anisotropy, material stability and residual stresses in soft tissue, in: Biomechanics of Soft Tissue in Cardiovascular Systems, Springer, 2003, pp. 65–108.

[35] Suhas Patankar, Numerical Heat Transfer and Fluid Flow, CRC Press, 1980.
[36] Charles S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220–252.
[37] Charles S. Peskin, The fluid dynamics of heart valves: experimental, theoretical, and computational methods, Annu. Rev. Fluid Mech. 14 (1) (1982) 235–259.
[38] Alice Raeli, Michel Bergmann, Angelo Iollo, A finite-difference method for the variable coefficient Poisson equation on hierarchical Cartesian meshes, J. Comput. Phys. 355 (2018) 59–77.
[39] Rolf Rannacher, Thomas Richter, An adaptive finite element method for fluid-structure interaction problems based on a fully Eulerian formulation, in: Fluid Structure Interaction II, Springer, 2011, pp. 159–191.
[40] Chae M. Rhie, Wei-Liang Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA J. 21 (11) (1983) 1525–1532.
[41] Thomas Richter, A fully Eulerian formulation for fluid–structure-interaction problems, J. Comput. Phys. 233 (2013) 227–240.
[42] Ronald S. Rivlin, Large elastic deformations of isotropic materials IV. Further developments of the general theory, Philos. Trans. R. Soc. Lond. Ser. A, Math. Phys. Sci. 241 (835) (1948) 379–397.
[43] Giovanni Russo, Peter Smereka, A remark on computing distance functions, J. Comput. Phys. 163 (1) (2000) 51–67.
[44] James A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Natl. Acad. Sci. 93 (4) (1996) 1591–1595.
[45] James A. Sethian, Fast marching methods, SIAM Rev. 41 (2) (1999) 199–235.
[46] Atul Sharma, Level set method for computational multi-fluid dynamics: a review on developments, applications and analysis, Sadhana 40 (3) (2015) 627–652.
[47] Kazuyasu Sugiyama, Satoshi Ii, Shintaro Takeuchi, Shu Takagi, Yoichiro Matsumoto, A full Eulerian finite difference approach for solving fluid–structure coupling problems, J. Comput. Phys. 230 (3) (2011) 596–627.
[48] Mark Sussman, Peter Smereka, Stanley Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1) (1994) 146–159.
[49] Claire Taymans, Solving Incompressible Navier-Stokes Equations on Octree grids: towards Application to Wind Turbine Blade Modelling, Bordeaux, PhD thesis, 2018.
[50] Roger Temam, Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (ii), Arch. Ration. Mech. Anal. 33 (5) (1969) 377–385.
[51] Tayfun E. Tezduyar, Mittal Behr, S. Mittal, 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, Comput. Methods Appl. Mech. Eng. 94 (3) (1992) 339–351.
[52] Tayfun E. Tezduyar, Mittal Behr, S. Mittal, 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, Comput. Methods Appl. Mech. Eng. 94 (3) (1992) 353–371.
[53] Yen-Hsi Richard Tsai, Li-Tien Cheng, Stanley Osher, Hong-Kai Zhao, Fast sweeping algorithms for a class of Hamilton–Jacobi equations, SIAM J. Numer. Anal. 41 (2) (2003) 673–694.
[54] Stefan Turek, Jaroslav Hron, Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow, in: Fluid-Structure Interaction, Springer, 2006, pp. 371–385.
[55] Boris Valkov, Chris H. Rycroft, Ken Kamrin, Eulerian method for multiphase interactions of soft solid bodies in fluids, J. Appl. Mech. 82 (4) (2015).
[56] Yong-Tao Zhang, Hong-Kai Zhao, Jianliang Qian, High order fast sweeping methods for static Hamilton–Jacobi equations, J. Sci. Comput. 29 (1) (2006) 25–56.
[57] Hong Zhao, Jonathan B. Freund, Robert D. Moser, A fixed-mesh method for incompressible flow–structure systems with finite solid deformations, J. Comput. Phys. 227 (6) (2008) 3114–3140.