

MOX-Report No. 35/2025

**Anisotropic recovery-based error estimators and mesh adaptation for
real-life engineering innovation**

Perotto, S.; Ferro, N.; Speroni, G.; Temellini, E.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it

<https://mox.polimi.it>

Anisotropic recovery-based error estimators and mesh adaptation for real-life engineering innovation

Simona Perotto¹, Nicola Ferro², Giacomo Speroni¹, and Erika Temellini¹

¹MOX - Department of Mathematics,
Politecnico di Milano,

Piazza L. da Vinci 32, 20133 Milano, Italy

²Department of Molecular Sciences and Nanosystems,
Ca' Foscari University of Venice,

Campus Scientifico, Via Torino 155, 30172 Venezia Mestre, Italy

Abstract

This chapter presents an overview of anisotropic mesh adaptation techniques driven by recovery-based a posteriori error estimators. The first part outlines the theoretical foundation for anisotropic error estimation and the construction of metric-based adapted meshes in a steady context. The methodology is then extended to time-dependent problems by coupling mesh adaptation with adaptive time stepping, in a unified space-time framework. The approach is tested on three representative engineering applications, namely structural topology optimization, microstructured material design, and unsteady fluid dynamics, demonstrating the effectiveness in capturing directional features in space and heterogeneities in time. The proposed strategy offers practical advantages in terms of computational efficiency, broad applicability, and ease of integration into existing numerical solvers.

Keywords Anisotropic meshes; Adapted meshes; Recovery-based error estimators; Structural optimization; Cellular materials; Fluid dynamics; Finite elements.

This chapter is dedicated to Maurizio Falcone

1 Introduction

In real-world engineering simulations, achieving high accuracy without incurring prohibitive computational costs remains a fundamental challenge. Mesh adaptation addresses this by dynamically refining or coarsening the computational grid in response to the specific features of the solution. Unlike uniform meshes, adaptive discretizations concentrate computational effort where it is most needed, offering a more efficient balance between accuracy and cost [1, 2, 3, 4, 5].

Mesh adaptation is essential across a wide range of fields, including aerodynamics,

structural mechanics, biomedical engineering, and multiphysics problems (see, for instance, [6, 7] for an up-to-date overview on mesh adaptation applications). Adaptation strategies enable accurate and efficient resolution of complex features like shock waves, boundary layers, crack propagation, and multiscale interactions, where static, uniformly fine meshes are typically inadequate or computationally wasteful.

The integration of mesh adaptation into both open-source and commercial platforms highlights its central role in bridging advanced numerical methods and engineering applications. Libraries such as deal.II, FEniCS, and OpenFOAM offer adaptive capabilities within flexible finite element or finite volume frameworks [8, 9, 10]. High-performance tools like p4est support scalable mesh management for parallel computing, while commercial software such as ANSYS, COMSOL, and Abaqus provide adaptive meshing strategies widely used in industry [11, 12, 13, 14].

Mesh adaptation strategies are typically guided by error estimators, which provide a quantitative information about where and how the mesh has to be modified. Error estimators can be distinguished between a priori and a posteriori. A priori estimators are derived from theoretical analysis and offer general insights into solution behavior, being generally impractical for effectively driving adaptive procedures in real simulations [5]. In contrast, a posteriori estimators are explicitly computable since depending on the discrete solution, thus leading to a impactful mesh adaptation process. In the a posteriori setting, a further distinction is made between heuristic indicators (i.e., ad hoc criteria without rigorous justification) and mathematically sound estimators, which provide reliable and sharp error bounds [1, 2, 3, 4].

Among the several a posteriori error estimators available in the literature, recovery-based estimators have emerged as a popular and effective tool [15, 16, 17]. These techniques reconstruct a higher-order approximation, in general, of the solution gradient, and compare it with the discrete solution to surrogate the discretization error. Their main advantages lie in generality, low computational overhead, and ease of implementation. Notably, recovery-based estimators are largely independent of the governing partial differential equations, boundary conditions, or discretization schemes, making them highly versatile for a wide range of problems (for examples in different disciplines, see, for instance, [18, 19, 20, 21, 22]). While recovery-based estimators are applicable in various numerical frameworks, this chapter focuses on the implementation within the finite element method, which is widely used in engineering due to the high flexibility in handling complex geometries and boundary conditions, as well as the strong theoretical foundations.

Another crucial aspect of mesh adaptation is the distinction between isotropic and anisotropic grids. Isotropic adaptation adjusts the mesh uniformly in all directions, which is simple but often inefficient for problems with directional features. Anisotropic adaptation, by contrast, allows for element resizing, reshaping, and reorientation to align the grid with the solution behavior. This leads to a more efficient allocation of degrees of freedom, particularly in problems exhibiting intrinsic directionally dominant physics. Although more complex to implement, anisotropic adaptation offers significant benefits in terms of both computational cost and accuracy (see, e.g., [23, 24, 25, 26, 27, 28, 29,

30, 31, 32, 33]).

This chapter addresses anisotropic mesh adaptation guided by recovery-based a posteriori error estimators, aiming to leverage the respective strengths of both approaches. Sections 2 and 3 provide a detailed explanation of the methodology, initially in a time-independent context that offers a simplified yet representative setting for development and validation. The proposed approach is then validated in Sects. 4 and 5 through two practical use cases related to structural optimization and cellular material design, respectively.

As a further improvement, with a view to a faithful representation of real-world engineering phenomena, it is necessary to move to a time-dependent setting, where the adaptation of the time discretization is advisable. Indeed, time adaptivity becomes particularly valuable in simulations featuring localized temporal events, such as fast transients, moving fronts, combustion, cardiac electrophysiology, fracture propagation, or phase-field dynamics, where uniform time stepping would either miss critical dynamics or lead to excessive computational effort. In the second part of this chapter (Sect. 6), we explore the extension of the proposed recovery-based framework to time-dependent problems, incorporating adaptive time stepping and space-time mesh adaptation, thus responding efficiently to the demands of real-life applications. Section 7 illustrates the methodology through a fluid dynamics case study as an example of the challenges commonly faced in real-world engineering simulations.

2 An anisotropic recovery-based a posteriori error analysis

This section outlines the mathematical framework supporting the error control adopted in this chapter to carry out an anisotropic mesh adaptation process. We first introduce the reference geometric setting, then present the selected error estimator, initially in its original isotropic formulation and subsequently extended to the anisotropic context of interest. Here, we focus on the 3-dimensional (3D) context.

2.1 The anisotropic setting

We adopt as reference framework the one proposed in [34], which generalizes the seminal contributions in [35, 36] to the 3D case. We denote by $\Omega \subset \mathbb{R}^3$ the selected computational domain and we assume to deal with a polyhedral volume in order to discard the geometric error associated with the rectification of curved boundaries. We partition Ω into a family $\mathcal{T}_h = \{K\}$ of conformal tetrahedral elements, such that $\bigcup_{K \in \mathcal{T}_h} K = \Omega$ (see, e.g., [37]). According to [35, 36], we extract the anisotropic features of elements K from the standard affine transformation, T_K , between the reference tetrahedron \hat{K} and K , such that

$$T_K : \hat{K} \rightarrow K, \quad \mathbf{x} = T_K(\hat{\mathbf{x}}) = J_K \hat{\mathbf{x}} + \mathbf{t}_K,$$

with $J_K \in \mathbb{R}^{3 \times 3}$ the Jacobian matrix of T_K , $\mathbf{t}_K \in \mathbb{R}^3$ a translation vector, $\mathbf{x} = (x, y, z)$ and $\hat{\mathbf{x}} = (\hat{x}, \hat{y}, \hat{z})$ the generic point in K and \hat{K} , respectively. The choice of the reference tetrahedron is not unique. In particular, we identify \hat{K} with the equilateral tetrahedron

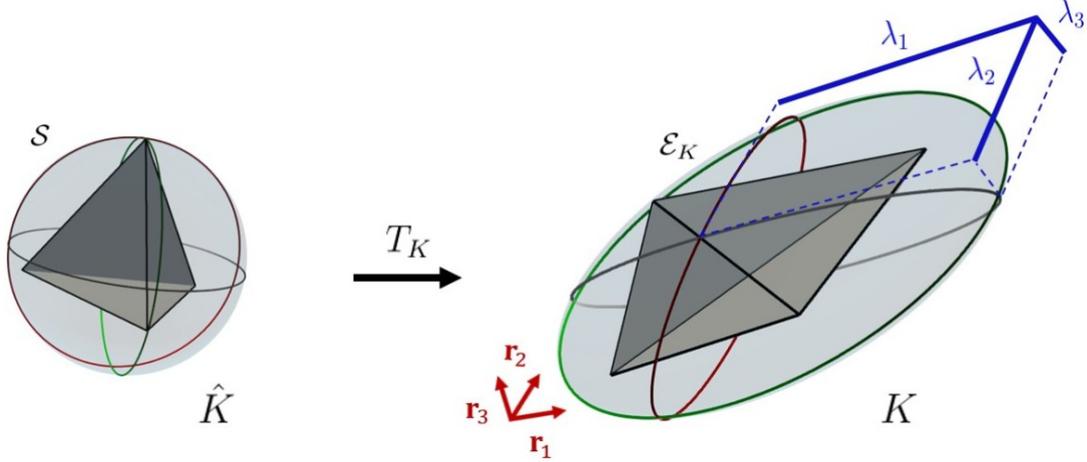


Figure 1: Affine transformation between the reference and the generic tetrahedron and main anisotropic quantities.

centered at the origin and inscribed into the unit sphere \mathcal{S} . We observe that map T_K changes \mathcal{S} into an ellipsoid, \mathcal{E}_K , circumscribed to K (see Fig. 1). The geometry of \mathcal{E}_K can be used to determine the anisotropic properties of element K . To this aim, we apply a polar decomposition combined with a standard spectral factorization to matrix J_K , so that

$$J_K = B_K Z_K = (R_K^T \Lambda_K R_K) Z_K, \quad (1)$$

where B_K and Z_K are a symmetric positive definite and an orthogonal matrix, respectively, while matrices $R_K = [\mathbf{r}_{1,K}, \mathbf{r}_{2,K}, \mathbf{r}_{3,K}]^T$ and $\Lambda_K = \text{diag}(\lambda_{1,K}, \lambda_{2,K}, \lambda_{3,K})$ collect the eigenvectors and the eigenvalues of B_K . From a geometric viewpoint, matrix B_K is responsible for the deformation of K with respect to \hat{K} , while Z_K prescribes a rotation to K . Concerning the eigenvectors and the eigenvalues of B_K , quantities $\{\lambda_{i,K}\}$ and $\{\mathbf{r}_{i,K}\}$ provide the measure and the direction of the semi-axes of the ellipsoid \mathcal{E}_K . In addition, matrix Z_K identifies element K among all the possible tetrahedra inscribed into \mathcal{E}_K . Without loss of generality, we assume the ordering $\lambda_{1,K} \geq \lambda_{2,K} \geq \lambda_{3,K} > 0$. Moreover, we quantify the element deformation through the stretching factors

$$s_{i,K} = \left(\prod_{j=1}^3 \lambda_{j,K} \right)^{-\frac{2}{3}} \lambda_{i,K}^2 \quad i = 1, 2, 3, \quad (2)$$

so that $\prod_{i=1}^3 s_{i,K} = 1$ (notice that the standard isotropic setting is a particular case of the anisotropic framework for $\lambda_{1,K} = \lambda_{2,K} = \lambda_{3,K}$, so that $s_{i,K} = 1$ for $i = 1, 2, 3$).

2.2 Recovery-based a posteriori error estimators

In this section, we present the foundational ideas for a recovery-based error analysis, distinguishing between the pioneering isotropic setting and the most recent extension to

an anisotropic context. To this aim, we consider a general physical problem governed by a partial differential equation (PDE), whose solution is denoted by $z = z(\mathbf{x}) \in Z$, with Z a proper function space [37] and $\mathbf{x} \in \bar{\Omega}$. The discrete counterpart of the PDE model can be obtained by resorting to different approximation schemes. Throughout this chapter, we refer to the finite element method [37], indicating by $z_h = z_h(\mathbf{x}) \in Z_h \subset Z$ the finite element discretization of the weak solution to the PDE problem¹, where

$$Z_h = X_h^p = \{v_h \in C^0(\bar{\Omega}) : v_h|_K \in \mathbb{P}_p(K), \forall K \in \mathcal{T}_h\} \quad (3)$$

is the space of the continuous finite element functions of degree p , with $\dim(Z_h) < +\infty$. With a view to the formalization of mesh adaptation, function z_h will play the role of the procedure driver.

2.2.1 The isotropic case

Recovery-based error estimators were proposed for the first time in the seminal works by O.C. Zienkiewicz and J.Z. Zhu [15, 16, 17]. The authors establish a heuristic recipe to estimate the $H^1(\Omega)$ -seminorm

$$|e_z|_{H^1(\Omega)}^2 = \int_{\Omega} |\Theta z - \Theta z_h|^2 d\Omega \quad (4)$$

of the discretization error $e_z = z - z_h$, associated with the finite element approximation z_h of z , with Θ a generic first-order differential operator, here coinciding with the gradient. The interest in controlling the $H^1(\Omega)$ -seminorm of e_z is justified by practical reasons since the gradient of the reference quantity often plays a key role in many applications (e.g, stress in elasticity, current in semiconductors, electric fields in electrostatics).

The idea pursued to estimate the seminorm in (4) simply consists in replacing $\Theta z = \nabla z$ – in principle not explicitly computable – with a quantity depending on the discretization z_h , known as the recovered gradient $\mathcal{R}_{\nabla}(z_h)$. This approach leads to define a so-called recovery-based error estimator $\eta = \eta(z_h)$, with

$$\eta^2 = \sum_{K \in \mathcal{T}_h} \eta_K^2 = \sum_{K \in \mathcal{T}_h} \int_K |\mathbf{E}_K(z_h)|^2 dK, \quad (5)$$

with $\mathbf{E}_K(z_h) = \mathcal{R}_{\nabla}(z_h) - \nabla z_h$ the recovered error and η_K the (local) error estimator associated with tetrahedron K .

Despite the simple structure of estimator η in (5), recovery-based error estimators are widely employed in diverse engineering applications due the ease of implementation (compared, for instance, to residual- and/or adjoint-based approaches [1]); computational cheapness (the estimator generally depending only on z_h); reliability (providing robust error estimates for general-purpose applications); and versatility (being easily applicable to a wide range of numerical schemes alternative to finite elements).

¹In accordance with the classical literature, we adopt the same symbol, z , to denote the solution to both the strong and the weak form of the PDE problem.

Concerning the choice of the recovered gradient $\mathcal{R}_\nabla(z_h)$, several recipes have been experimented in the last decades, with a particular attention to the specific application and to the balance between accuracy and computational efficiency. It is common to distinguish between local and global approaches.

The local techniques compute the upgraded discrete gradient, $\mathcal{R}_\nabla(z_h)$, by using information from a small neighborhood of elements. These methods are designed to be computationally efficient and handle irregular meshes effectively, and are commonly used in finite volume and finite element methods. Among the most commonly used local approaches, we mention:

- i) the Green-Gauss methods (which approximate the gradient as a flux across element boundaries using Green's theorem, providing a simple and efficient, but sensitive to mesh quality, procedure – see, e.g., [38]);
- ii) the least-squares methods (which recover gradients by minimizing residuals in a least-squares sense, being robust on unstructured meshes but computationally demanding – see, e.g., [39]);
- iii) the polynomial fitting (which resorts to a local polynomial over a suitable patch of elements to reconstruct the gradients, thus offering flexible accuracy although affected by oscillations on irregular meshes – see, e.g., [40]);
- iv) the recovery-based methods (which post-process the discrete gradient with high-order fitting to obtain superconvergent approximations, proving to be highly accurate although, in some cases, computationally demanding – see, e.g., [16]).

The global approaches use information from the entire domain to reconstruct $\mathcal{R}_\nabla(z_h)$. They involve more computational overhead compared to local recipes, but ensure higher accuracy. These methods are particularly suited to guarantee a smooth and accurate gradient across the entire Ω , or when employing high-order discretizations (e.g., spectral or *hp*-finite element schemes) [41]. Some of the most widely adopted global approaches include:

- v) the spline-based reconstructions (which achieve smooth gradients via global polynomial or spline interpolation, resulting in accurate but costly procedures on unstructured meshes – see, e.g., [42]);
- vi) the projection methods (which construct $\mathcal{R}_\nabla(z_h)$ by projecting ∇z_h onto a higher-order basis, ensuring compatibility with the adopted discretization, although at a high computational cost for complex geometries – see, e.g., [43]);
- vii) the harmonic smoothing (which solves a global harmonic problem to produce smooth gradient fields, effectively reducing noise – see, e.g., [44]);
- viii) the radial basis function interpolation (which provides smooth gradients from scattered data, although being sensitive to point distribution – see, e.g., [45]).

Throughout this chapter, we choose $\mathcal{R}_\nabla(z_h)$ as the area-weighted average of the discrete gradient across the patch, $\Delta_K = \{T \in \mathcal{T}_h : T \cap K \neq \emptyset\}$, of the elements sharing at least one vertex with K , namely

$$\mathcal{R}_\nabla(z_h)(\mathbf{x}) = \frac{1}{|\Delta_K|} \sum_{T \in \Delta_K} |T| \nabla z_h|_T \quad \text{for } \mathbf{x} \in K, \quad (6)$$

with $|\varpi|$ the measure of the generic set $\varpi \subset \mathbb{R}^3$. Although basic in its formulation, the recovery-based error estimator (5) identified by the choice in (6) demonstrates a high degree of reliability [31, 46, 47, 48, 49, 50]. Moreover, this simplicity has been crucial for extending recovery-based estimators to an anisotropic setting, which is intrinsically challenging.

2.2.2 The anisotropic case

In 2010, an anisotropic counterpart of the recovery-based error estimator in (5) was proposed in [51]. To this aim, the authors take inspiration from the local quasi-interpolation anisotropic error estimate

$$\|z - I_K^1(z)\|_{L^2(K)}^2 \leq C_1 \left(\sum_{i=1}^3 \lambda_{i,K}^2 (\mathbf{r}_{i,K}^T G_{\Delta_K}(\nabla z) \mathbf{r}_{i,K}) \right) \quad (7)$$

proved in [35], where $C_1 > 0$ is a constant depending on the regularity of the patch Δ_K , $I_K^1 : H^1(\Omega) \rightarrow \mathbb{R}$ is a quasi-interpolant operator [37], and G_{Δ_K} denotes the symmetric semi-definite positive matrix with entries

$$[G_{\Delta_K}(\mathbf{w})]_{ij} = \int_{\Delta_K} w_i w_j d\Delta_K \quad \text{for } i, j = 1, 2, 3, \quad (8)$$

with $\mathbf{w} = [w_1, w_2, w_3]^T \in [L^2(\Omega)]^3$ a generic vector-valued function. The idea behind the anisotropic estimator is based on a direct comparison between (7) and the corresponding isotropic error estimate

$$\|z - I_K^1(z)\|_{L^2(K)}^2 \leq C_2 \left(h_K^2 \int_{\Delta_K} |\nabla z|^2 d\Delta_K \right), \quad (9)$$

with $C_2 > 0$ constant. This suggests that the term on the right-hand side of (7) provides, up to a constant, the anisotropic counterpart of the quantity in brackets in (9). This leads to define the anisotropic recovery-based error estimator $\eta_A = \eta_A(z_h)$, with

$$\eta_A^2 = \sum_{K \in \mathcal{T}_h} \eta_{A,K}^2 = \sum_{K \in \mathcal{T}_h} \left(\mathcal{S}_K \sum_{i=1}^3 \lambda_{i,K}^2 (\mathbf{r}_{i,K}^T G_{\Delta_K}(\mathbf{E}_K(z_h)) \mathbf{r}_{i,K}) \right), \quad (10)$$

with $\mathcal{S}_K = \prod_{i=1}^3 \lambda_{i,K}^{-2/3}$ (for more details, we refer to [51]). We observe that the anisotropic and the recovery-based features of estimator η_A are justified by quantities $\{\lambda_{i,K}, \mathbf{r}_{i,K}\}_{i=1}^3$ and $\mathbf{E}_K(z_h)$, respectively while the scaling factor \mathcal{S}_K guarantees the consistency of the anisotropic estimator with the isotropic counterpart η in (5).

As detailed in the next section, estimator η_A will be instrumental to settle the mesh adaptation procedure used in the real-life applications in Sects. 4-5 and 7.

3 From the estimator to the adapted mesh

In this section we provide a practical procedure to commute estimator η_A computed on mesh \mathcal{T}_h into a new adapted mesh, \mathcal{T}_h^* , matching prescribed optimality criteria. To this aim, we resort to the concept of metric [5], namely a symmetric, positive definite tensor field $\mathcal{M} : \Omega \rightarrow \mathbb{R}^{3 \times 3}$ defining the distribution of the mesh elements in Ω . Although \mathcal{M} is defined for each point $\mathbf{x} \in \Omega$, it is standard practice to approximate the metric through a piecewise constant information. In particular, there exists a bijection, $\mathcal{M} \leftrightarrow \mathcal{T}_h$, between metric and mesh, so that we can associate:

- ← a piecewise-constant metric $\mathcal{M}(\mathcal{T}_h) = \{\mathcal{M}_K\}_{K \in \mathcal{T}_h}$ with a given mesh \mathcal{T}_h , with $\mathcal{M}_K = R_K^T \Lambda_K^{-2} R_K$, matrices R_K and Λ_K being defined as in (1);
- a mesh \mathcal{T}_h^* with a given piecewise-constant metric $\mathcal{M}^* = \mathcal{M}(\mathcal{T}_h^*) = \{\mathcal{M}_K^*\}_{K \in \mathcal{T}_h^*}$, when an auxiliary background mesh \mathcal{T}_h^b is available. Metric \mathcal{M}_K^* is assigned through the quantities $\{\lambda_{i,K}^*, \mathbf{r}_{i,K}^*\}_{i=1}^3$, which have the same geometric meaning as in Fig. 1 and are organized in the matrices R_K^* and Λ_K^* , such that $\mathcal{M}_K^* = (R_K^*)^T (\Lambda_K^*)^{-2} R_K^*$.

According to a metric-based framework, it is well-established that an adapted mesh has the property that each edge has a unit length when measured through the associated metric [5].

To generate the adapted mesh \mathcal{T}_h^* , we resort to an iterative process that, at the generic iteration j , computes the estimator η_A on the current mesh $\mathcal{T}_h^{(j)}$ and derives the metric $\mathcal{M}^{(j+1)} = \mathcal{M}(\mathcal{T}_h^{(j+1)})$ to be used for generating the new mesh $\mathcal{T}_h^{(j+1)}$. The computation of η_A is carried out through the expression in (10), after setting $\mathcal{T}_h = \mathcal{T}_h^b = \mathcal{T}_h^{(j)}$. Instead, the derivation of $\mathcal{M}^{(j+1)}$ deserves more comments. As a first task, we select the criteria driving the metric prediction that we identify with the minimization of the number of the mesh elements (i.e., of the mesh cardinality, $\#\mathcal{T}_h^{(j+1)}$), the imposition of a desired accuracy, TOL, on the error (i.e., on the estimator), being

$$\eta_A = \text{TOL},$$

and the equidistribution of the error (i.e., of the estimator) throughout the mesh elements so that

$$\eta_{A,K}^2 = \frac{\text{TOL}^2}{\#\mathcal{T}_h^{(j)}}. \quad (11)$$

To match these requirements, we start by rewriting the local estimator by highlighting a volume information, namely

$$\eta_{A,K}^2 = \mathcal{V}_K \mathcal{J}(\{s_{i,K}, \mathbf{r}_{i,K}\}_{i=1}^3),$$

with

$$\mathcal{V}_K = |\Delta_K| = \left(\prod_{i=1}^3 \lambda_{i,K} \right) |\widehat{\Delta}_K|, \quad \mathcal{J}(\{s_{i,K}, \mathbf{r}_{i,K}\}_{i=1}^3) = \sum_{i=1}^3 s_{i,K} (\mathbf{r}_{i,K}^T \widehat{G}_{\Delta_K}(\mathbf{E}_K(z_h)) \mathbf{r}_{i,K}),$$

the patch volume and the size-independent local error estimator, respectively where $\widehat{\Delta}_K = T_K^{-1}(\Delta_K)$ is the pullback of the patch Δ_K through map T_K , while \widehat{G}_{Δ_K} denotes matrix G_{Δ_K} in (8) scaled with respect to the patch volume. Now, the error equidistribution imposes that

$$\eta_{A,K}^2 = \mathcal{V}_K \mathcal{J}(\{s_{i,K}, \mathbf{r}_{i,K}\}_{i=1}^3) = \text{constant}. \quad (12)$$

In addition, we observe that the minimization of the mesh cardinality is equivalent to the maximization of the element volume. Thus, to ensure relation (12) under the volume maximization requirement, we are led to solve the constrained minimization problem

$$\min_{\{s_{i,K}, \mathbf{r}_{i,K}\}_{i=1}^3} \mathcal{J}(\{s_{i,K}, \mathbf{r}_{i,K}\}_{i=1}^3) \quad \text{s.t.} \quad \begin{cases} \mathbf{r}_{i,K} \cdot \mathbf{r}_{j,K} = \delta_{ij} \\ \prod_{i=1}^3 s_{i,K} = 1, \end{cases} \quad (13)$$

for any $K \in \mathcal{T}_h^{(j)}$. Problem (13) has to be solved on each element. However, it admits an explicit solution given by

$$s_{i,K}^* = \left(\prod_{i=1}^3 g_i \right)^{1/3} g_{4-i}^{-1}, \quad \mathbf{r}_{i,K}^* = \mathbf{g}_{4-i} \quad i = 1, 2, 3, \quad (14)$$

where $\{g_i, \mathbf{g}_i\}_{i=1}^3$ denote the eigenpairs of matrix $\widehat{G}_{\Delta_K}(\mathbf{E}_K(z_h))$ ordered such that $g_1 \geq g_2 \geq g_3$. To extract the optimal lengths $\lambda_{i,K}^*$ from the stretching factors in (14), we explicitly impose the equidistribution relation in (11), to obtain

$$\lambda_{i,K}^* = \left(\frac{\text{TOL}^2}{3 \#\mathcal{T}_h^{(j)} |\widehat{\Delta}_K|} \right)^{1/3} \left(\prod_{i=1}^3 g_i \right)^{1/18} g_{4-i}^{-1/2} \quad i = 1, 2, 3. \quad (15)$$

Quantities $\lambda_{i,K}^*$ in (15) and $\mathbf{r}_{i,K}^*$ in (14) identify the metric $\mathcal{M}^{(j+1)}$ associated with the new adapted mesh, being

$$\mathcal{M}^{(j+1)} = \{\mathcal{M}_K^{(j+1)}\}_{K \in \mathcal{T}_h^{(j)}},$$

where $\mathcal{M}_K^{(j+1)} = (R_K^*)^T (\Lambda_K^*)^{-2} R_K^*$, with $R_K^* = [\mathbf{r}_{1,K}^*, \mathbf{r}_{2,K}^*, \mathbf{r}_{3,K}^*]^T$, $\Lambda_K^* = \text{diag}(\lambda_{1,K}^*, \lambda_{2,K}^*, \lambda_{3,K}^*)$. For the proof of relations (14)-(15), we refer the interested reader to [34] where the authors also tackle the degenerate case when matrix $\widehat{G}_{\Delta_K}(\mathbf{E}_K(z_h))$ is symmetric positive semidefinite.

The construction of mesh $\mathcal{T}_h^{(j+1)}$ is handled by a metric-based mesh generator. Many software tools associate the metric with the background mesh vertices. This requirement calls for a vertex-wise definition of $\mathcal{M}^{(j+1)}$ that we define as $\widetilde{\mathcal{M}}^{(j+1)} = \{\mathcal{M}_V^{(j+1)}\}_{V \in \mathcal{T}_h^{(j)}}$, with $\mathcal{T}_h^b = \mathcal{T}_h^{(j)}$, V the generic vertex of $\mathcal{T}_h^{(j)}$ and

$$\mathcal{M}_V^{(j+1)} = \frac{1}{|\Delta_V|} \sum_{K \in \Delta_V} |K| \mathcal{M}_K^{(j+1)}, \quad (16)$$

with $\Delta_V = \{T \in \mathcal{T}_h^{(j)} : T \ni V\}$. Metric $\widetilde{\mathcal{M}}^{(j+1)}$ is now suited to be used as an input for a metric-based mesh generator, which returns the adapted mesh $\mathcal{T}_h^{(j+1)}$ through the

implementation of local refinement and coarsening operations (e.g., edge/face splitting and swapping, edge collapsing, point insertion) [52]. This concludes the j -th step of the adaptive procedure.

To stop the iterative adaptation process, we impose a maximum number of iterations, N_{\max} , combined with a control within a tolerance Δ_S on the mesh cardinality stagnation, or, as an alternative, on the error estimator stagnation, defined by

$$\Delta_{\mathcal{T}_h} = \frac{|\#\mathcal{T}_h^{(j+1)} - \#\mathcal{T}_h^{(j)}|}{\#\mathcal{T}_h^{(j)}}, \quad \Delta_{\eta_A} = \frac{|\eta_A^{(j+1)} - \eta_A^{(j)}|}{\eta_A^{(j)}}, \quad (17)$$

respectively, where we have denoted by $\eta_A^{(s)}$ the error estimator in (10) evaluated on the mesh $\mathcal{T}_h^{(s)}$ for $s = j, j + 1$.

Remark 3.1 (The case of an isotropic metric). *The geometric information provided by an isotropic metric reduces to the size (i.e., the diameter h_K) of the mesh tetrahedra in contrast to an anisotropic metric which features the size (through lengths $\lambda_{i,K}$), the shape (through ratios $s_{i,K}$) and the orientation (through directions $\mathbf{r}_{i,K}$) of each element. This property leads to considerably simplify the definition of an isotropic metric, which can be written as $\mathbf{M}_K^* = (\mathbf{R}_K^*)^T (\Lambda_K^*)^{-2} \mathbf{R}_K^*$, with $\Lambda_K^* = \text{diag}(h_K, h_K, h_K)$.*

Concerning the optimal metric in (14)–(15), we can replicate the procedure above, leading to preserve the optimal directions $\mathbf{r}_{i,K}^$, while replacing the optimal lengths with*

$$\lambda_{1,K}^* = \lambda_{2,K}^* = \lambda_{3,K}^* = \left(\frac{\text{TOL}^2}{3 \#\mathcal{T}_h |\widehat{\Delta}_K|} \right)^{1/3} \left(\frac{3}{g_1 + g_2 + g_3} \right)^{1/3}.$$

Remark 3.2 (The 2D case). *The definition of the anisotropic recovery-based error estimator in (10), as well as the derivation of the optimal metric in (14)–(15) can be restricted to a 2-dimensional setting upon suitable dimension downscaling. In particular, in estimator η_A the index i takes the values 1 and 2 only, while the scaling factor becomes $\mathbf{S}_K = \prod_{i=1}^2 \lambda_{i,K}^{-1}$.*

The metric derived by the optimality procedure described above leads to identify $\mathbf{r}_{1,K}^ = \mathbf{g}_2$, $\mathbf{r}_{2,K}^* = \mathbf{g}_1$, and to select the lengths*

$$\lambda_{1,K}^* = \left(\frac{\text{TOL}^2}{2 \#\mathcal{T}_h |\widehat{\Delta}_K|} \right)^{1/2} g_2^{-1/2}, \quad \lambda_{2,K}^* = \left(\frac{\text{TOL}^2}{2 \#\mathcal{T}_h |\widehat{\Delta}_K|} \right)^{1/2} g_1^{-1/2}, \quad (18)$$

with $\{g_i, \mathbf{g}_i\}_{i=1}^2$ the eigenpairs of matrix $\widehat{\mathbf{G}}_{\Delta_K}(\mathbf{E}_K(z_h))$ defined exactly as in (8) for $i, j = 1, 2$. In a similar way, we can derive the isotropic counterpart of lengths (18), being

$$\lambda_{1,K}^* = \lambda_{2,K}^* = \left(\frac{\text{TOL}^2}{2 \#\mathcal{T}_h |\widehat{\Delta}_K|} \right)^{1/2} \left(\frac{g_1 + g_2}{2} \right)^{-1/2}.$$

Remark 3.3 (Computational considerations). *When running the adaptive procedure, it is advisable to introduce a control, h_{\min} and h_{\max} , both on the minimum and the maximum size allowed for the generic element. This caution prevents an over-refinement/coarsening*

in the areas where the discrete solution z_h exhibits, for instance, a singularity or is constant, respectively (see [34, Remark 4.2]).

Concerning the vertex-wise metric in (16), it might be useful to properly scale such a definition in order to preserve the unit-length property shared by the edges of the adapted mesh.

Below, we provide a pseudo-code that implements the anisotropic mesh adaptation process in detail. The routine `space_estimator` evaluates the quantity in (10), while `make_metrica` computes the optimal metric defined in (14)–(16). The `adapt_mesh` routine generates the anisotropically adapted mesh, interfacing directly with the selected metric-based mesh generator. The `project` routine transfers the numerical solution from the old mesh to the new one. Finally, the `update_Delta` routine computes Δ_{Delta} based on the value of the flag `Delta`. This input can be set to either ‘`cardinality`’ or ‘`estimator`’ to monitor the stagnation of the mesh cardinality or of the error estimator in (17), respectively.

Algorithm 1 : anisotropic mesh adaptation

```

function  $[\mathcal{T}_h^*, \eta_A^*] = \text{space\_adaptation}(z_h, \mathcal{T}_h^b, \text{TOL}, N_{\text{max}}, \Delta_S, \text{Delta}, h_{\text{min}}, h_{\text{max}})$ 
1:  $j = 0$ ;
2:  $\Delta_{\text{Delta}} = \Delta_S + 1$ ;
3:  $\eta_A = \text{space\_estimator}(z_h, \mathcal{T}_h^b)$ ;
4: while ( $\Delta_{\text{Delta}} \geq \Delta_S$  &  $j < N_{\text{max}}$ ) do
5:    $\widetilde{\mathcal{M}} = \text{make\_metrica}(\mathcal{T}_h^b, \eta_A, \text{TOL}, h_{\text{min}}, h_{\text{max}})$ ;
6:    $\mathcal{T}_h^* = \text{adapt\_mesh}(\mathcal{T}_h^b, \widetilde{\mathcal{M}})$ ;
7:    $z_h = \text{project}(z_h, \mathcal{T}_h^b, \mathcal{T}_h^*)$ ;
8:    $\eta_A^* = \text{space\_estimator}(z_h, \mathcal{T}_h^b)$ ;
9:    $\Delta_{\text{Delta}} = \text{update\_Delta}(\mathcal{T}_h^b, \mathcal{T}_h^*, \eta_A, \eta_A^*)$ ;
10:   $\mathcal{T}_h^b = \mathcal{T}_h^*$ ;
11:   $\eta_A = \eta_A^*$ ;
12:   $j = j+1$ ;
13: end while

```

4 Engineering use case 1: design of structures

The design of structural components across diverse engineering fields is increasingly supported by advanced numerical modeling techniques, which enable more accurate, efficient, and high-performance solutions. While traditional strategies such as size and

shape optimization refine existing geometries and parametric or heuristic methods (e.g., genetic algorithms) explore variations within fixed design spaces, topology optimization (TO) offers greater flexibility by determining the material distribution from scratch during the design process. This unique capability allows for the emergence of innovative, non-intuitive designs, unconstrained by predefined shapes.

The integration of TO with modern simulation tools and manufacturing technologies (e.g., additive manufacturing and 3D printing) has further amplified its impact, enabling the physical realization of complex structures directly from computational outputs. As a result, TO has become a powerful enabler of next-generation design across disciplines: from reducing material use in civil structures, to lightweighting components in aerospace and automotive applications, to crafting custom biomedical implants with enhanced biomechanical integration.

4.1 Technical background for use case 1

As a general standard, TO redistributes material and void within the design domain, $D \subset \mathbb{R}^3$, aiming to meet performance targets under given design or physical constraints. This allows structural boundaries to evolve freely and new features, such as holes or complex substructures, to emerge. To this aim, TO requires the definition of a design function, ρ , which represents the material/void alternation to be optimized across D . In this context, various mathematical methods are available, such as density-based approaches [53, 54, 55], level set methods [56, 57], topological derivatives [58], phase field methods [59], evolutionary approaches [60]. Following [61], we adopt here a density-based TO formulation which assumes density ρ to take values between 0 (void portions) and 1 (full material portions). In particular, the general TO problem can be written as

$$\min_{\rho \in \mathcal{S}} J(\mathbf{w}_1(\rho), \dots, \mathbf{w}_N(\rho); \rho) : \begin{cases} a_{i,\rho}(\mathbf{w}_i(\rho), \mathbf{v}) = f_{i,\rho}(\mathbf{v}) & \forall \mathbf{v} \in V, \quad i = 1, \dots, N \\ \underline{c}_j \leq c_j(\mathbf{w}_1(\rho), \dots, \mathbf{w}_N(\rho); \rho) \leq \bar{c}_j & j = 1, \dots, M \\ \underline{\rho} \leq \rho \leq 1, \end{cases} \quad (19)$$

where: $J(\cdot, \dots, \cdot; \rho)$ denotes the objective functional that drives the optimization process; $a_{i,\rho}(\cdot, \cdot)$ and $f_{i,\rho}(\cdot)$, for $i = 1, \dots, N$, are the bilinear and linear forms identifying the weak form of N state equations governing the physical behavior of the system, with $\mathbf{w}_i(\rho)$ the state variables and \mathbf{v} the corresponding test function, both varying in a suitable function space V ; functionals $c_j(\cdot, \dots, \cdot; \rho)$, for $j = 1, \dots, M$, model design and/or physical quantities of interest, being lower- and/or upper-bounded through values \underline{c}_j and \bar{c}_j , respectively; $\underline{\rho}$, with $0 < \underline{\rho} \ll 1$, is a lower bound on the density ρ used to ensure the well-posedness of (19); \mathcal{S} coincides with the admissible design space.

In this section, we consider the benchmark problem of structural compliance minimization (i.e., stiffness maximization) under a volume constraint in the physical setting of linear elasticity. The design domain D is a bounded open set whose boundary ∂D is decomposed into three disjoint parts, namely Γ_D (the Dirichlet boundary), Γ_N (the Neumann boundary), and Γ_F (the traction-free boundary). The constraining state equations

in (19) reduce to the weak form of the single (i.e., $N = 1$) linear elasticity equation

$$\begin{cases} -\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) = \mathbf{0} & \text{in } D \\ \mathbf{u} = \mathbf{0} & \text{on } \Gamma_D \\ \boldsymbol{\sigma}(\mathbf{u}) \mathbf{n} = \mathbf{t} & \text{on } \Gamma_N \\ \boldsymbol{\sigma}(\mathbf{u}) \mathbf{n} = \mathbf{0} & \text{on } \Gamma_F, \end{cases} \quad (20)$$

given by

$$a(\mathbf{u}, \mathbf{v}) = \int_D \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) \, dD = \int_{\Gamma_N} \mathbf{t} \cdot \mathbf{v} \, d\Gamma_N = f(\mathbf{v}) \quad \forall \mathbf{v} \in V, \quad (21)$$

where the state variable \mathbf{w}_1 coincides with the displacement field \mathbf{u} in $V = \{\mathbf{v} \in [H^1(D)]^3 : \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\}$; $\boldsymbol{\sigma}(\mathbf{u}) = 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) + \lambda \operatorname{tr}(\boldsymbol{\varepsilon}(\mathbf{u})) \mathbf{I}$ is the Cauchy stress tensor, with $\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ the linearized strain tensor, $\operatorname{tr}(\cdot)$ the trace operator, \mathbf{I} the identity tensor, μ and λ the Lamé coefficients; \mathbf{t} denotes the prescribed surface traction; \mathbf{n} is the unit outward normal vector to ∂D .

The design variable ρ , varying in the admissible design space $S = H^1(D; [\rho, 1])$, is included in (21) through the Solid Isotropic Material with Penalization (SIMP) approach [61]. SIMP replaces the Lamé coefficients λ and μ with the quantities $\lambda_\rho = \rho^p \lambda$ and $\mu_\rho = \rho^p \mu$ including a power-law penalization of the material density, so that

$$a_\rho(\mathbf{u}, \mathbf{v}) = a_{1,\rho}(\mathbf{u}, \mathbf{v}) = \int_D \boldsymbol{\sigma}_\rho(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) \, dD, = \int_{\Gamma_N} \mathbf{t} \cdot \mathbf{v} \, d\Gamma_N = f_{1,\rho}(\mathbf{v}) = f_\rho(\mathbf{v}), \quad (22)$$

with $\boldsymbol{\sigma}_\rho(\mathbf{u}) = 2\mu_\rho \boldsymbol{\varepsilon}(\mathbf{u}) + \lambda_\rho \operatorname{tr}(\boldsymbol{\varepsilon}(\mathbf{u})) \mathbf{I}$, and where we set p to 4 according to [61]. Concerning the objective functional in (19), it is given by

$$J(\mathbf{u}; \rho) = f_\rho(\mathbf{u}),$$

i.e., by the system compliance, measuring the work done by the external traction \mathbf{t} when the structure is subject to the displacement \mathbf{u} , while a single (i.e., $M = 1$) box-constraint is imposed to control the maximum allowed volume, being

$$0 \leq |D|^{-1} \int_D \rho \, dD \leq v_f,$$

namely $\underline{c}_1 = 0$, $c_1(\mathbf{u}; \rho) = c_1(\rho) = |D|^{-1} \int_D \rho \, dD$, and $\bar{c}_1 = v_f$, the maximum volume fraction.

4.2 Integration of anisotropic mesh adaptation into use case 1

The SIMP method is known to suffer from important issues when moving to the discrete counterpart. In our work, we adopt a finite element discretization, which, although

well-established, requires specific strategies to be effective in practice. Among the main challenges to be tackled, we cite the mesh dependency and the occurrence of checkerboard patterns. The former is associated with the dependence of the optimal layout on the adopted computational mesh, due to the non-uniqueness of the optimal solution. The latter is characterized by the presence of non-physical layouts, which alternate solid and void elements, and can be ascribed to the employment of low-order finite element spaces. Common remedies for both issues include density filtering (e.g., through Helmholtz- or Heaviside-type filters [62]), penalization schemes, and the use of higher-order elements for displacement, although the problem of non-uniqueness generally persists unless additional constraints are imposed.

Regardless the specific adopted mitigation, the choice of the computational grid remains a key point. Indeed, a too coarse mesh can yield vague, poorly defined designs that require significant post-processing with a view to a manufacturing phase. Vice versa, an extremely fine mesh increases the computational effort significantly while leading to excessively thin components to be manufactured. In such a context, SIMPATY algorithm has been proposed as an improvement to standard fixed-mesh approaches [31]. The idea behind is to enrich SIMP method with an anisotropic mesh adaptation, starting from the theoretical setting presented in Sects. 2–3 and after identifying function z with the material density ρ . In particular, estimator η_A through metric $\bar{\mathcal{M}}$ generates a mesh whose tetrahedra are sharply aligned with the material/void interface, where steep gradients of ρ occur. As a consequence, the SIMPATY algorithm enables free-form, cost-effective designs by automatically producing sharp and smooth density profiles. Final layouts can be extracted through simple thresholding, significantly reducing the post-processing effort required by standard methods. The good performance of SIMPATY algorithm has been validated in several application contexts, such as biomedical, civil and aerospace engineering [63, 48, 64, 65, 31].

As a final remark, we observe that the optimization process inherently involves an iterative procedure, analogously to mesh adaptation. Therefore, it is important to carefully coordinate the interplay between these two iterative algorithms. In principle, one can adopt either a tightly coupled or a mildly coupled strategy. In the former, the mesh is adapted at every optimization step, while in the latter, mesh adaptation is performed only after the optimization loop reaches a certain degree of convergence. These two approaches differ significantly in terms of both accuracy and computational cost (we refer the interested reader to [66] for a detailed discussion). In this work, we adopt the mildly coupled strategy, building on its previously validated effectiveness in [65, 31].

4.3 Simulation outcomes for use case 1

In this section, we test the design capability of SIMPATY algorithm on the 3D rod test case in [67] where the authors consider a cylindrical rod subject to pure torsion. The design domain D is a solid cylinder with radius $R = 0.5$, and height $L = 3$, while the material properties are assigned through the Young modulus $E_Y = 1$ and the Poisson

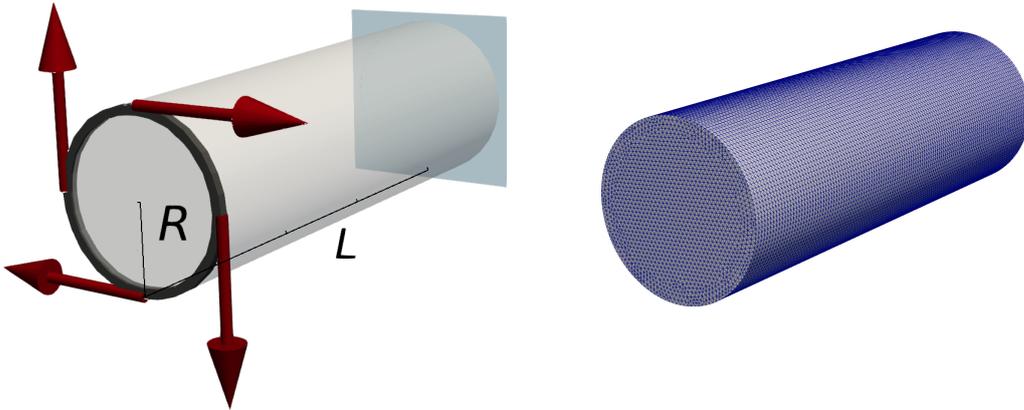


Figure 2: Use case 1: problem setting (left) and initial mesh (right)

ratio $\nu = 0.3$, so that

$$\lambda = \frac{\nu E_Y}{(1 + \nu)(1 - 2\nu)} = 0.577, \quad \mu = \frac{E_Y}{2(1 + \nu)} = 0.385. \quad (23)$$

The domain is clamped on one end and loaded on the opposite face through a rigid annulus with an inner radius of 0.45 and thickness of 0.05, characterized by a higher Young modulus equal to 10^3 . In particular, we apply the torsional load at four distinct points through a unitary tangential force (see Fig. 2 for a sketch). The remaining part of the rod boundary is traction-free. This setup induces a shear-dominated stress configuration.

Concerning the optimization setting in Sect. 4.1, we identify \mathbf{t} in (22) with the torsional load, we choose the volume fraction as $\nu_f = 0.136$, and we keep the annular region fixed. This optimization context turns out to be particularly challenging since it involves multiple geometric and numerical issues, namely the curved surface structure has to be preserved, while the rigid annulus and the force application points are expected to remain sharply resolved throughout the optimization in order to correctly transmit the applied loads. To take into account these requirements in the mesh adaptation procedure, in accordance with Remark 3.3, we opt for the assignment of a minimum/maximum element size, here assumed to range between 0.001 and 0.250. In addition, to guarantee an unbiased mechanical analysis of the optimized structure, we modify the metric in (16) by resorting to the hybrid approach proposed in [65], which adopts stretched elements along the material/void interface, and isotropic tetrahedra with prescribed uniform element size h_{iso} (here set to 0.050) in solid regions. As demonstrated, this hybrid approach provides a more accurate estimation of structural compliance compared to a fully anisotropic strategy, which tends to underestimate this mechanical quantity. Moreover, such an approach guarantees a reduction in terms of degrees of freedom relative to an isotropic adaptation with comparable element sizes [65].

SIMPATY algorithm is run starting from a uniform mesh $\mathcal{T}_h^{(0)}$ consisting of 1.082.880

elements (see Fig. 2, right), a initial uniform density field $\rho = v_f$, a mesh adaptation tolerance $\text{TOL} = 20$, while constraining the adaptation loop with a maximum number of iterations N_{max} equal to 10 and a control on the relative variation of the mesh cardinality $\Delta\mathcal{T}_h$ set to $\Delta_S = 0.1$. At this stage, the values adopted for the mesh adaptation parameters are selected to balance resolution, geometric fidelity, and computational cost. Given the central role of filtering schemes in TO [62], we consider two scenarios, one without any minimum size control, and the other where such a control is enforced using the well-established Helmholtz filter².

The TO in absence of the minimum size control yields, after 3 mesh updates, the hollow cylindrical tube in Fig. 3, consistently with the shell-like configurations provided in the literature, characterized by a compliance equal to 0.031 and a relative volume equal to 0.136. The final mesh features 7.957.378 elements. In particular, highly stretched (with a maximum aspect ratio on the order of 10^4) tetrahedra are concentrated along the thin material/void interface, which is mainly oriented in the axial direction (see the clipped and enlarged views in Fig. 3).

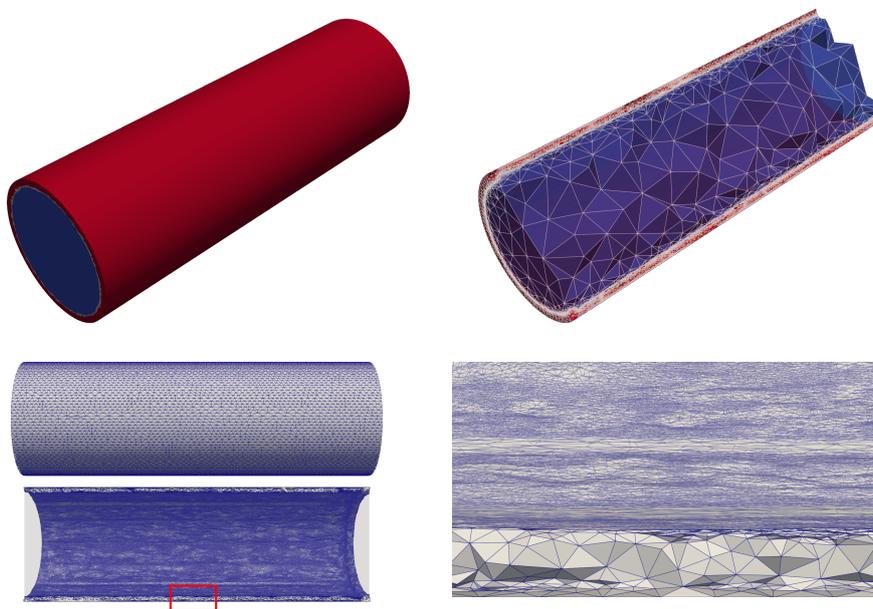


Figure 3: Use case 1 (without minimum size control): final material density distribution (top-left), clipped views of the internal and external rod parts (top-right, bottom-left) and of the material/void interface in correspondence of the red box (bottom-right).

²The simulations are carried out on a high-performance computing node equipped with two AMD EPYC 7413 24-core processors (96 threads total) and 512GB of RAM. The computational framework leverages the open-source software `mmg3d` for fully anisotropic 3D mesh adaptation, while the numerical solution of the governing equations is performed using FEniCSx, coupled with PETSc’s Flexible Generalized Minimal RESidual (FGMRES) Krylov solver and a Geometric-Algebraic MultiGrid (GAMG) preconditioner. The workflow is parallelized by task: the optimization stage runs on 48 processors, the metric construction on 12 processors, whereas mesh adaptation is performed serially.

When a minimum size control is enforced via a Helmholtz filter of radius 0.1, the optimized design exhibits a completely different topology, characterized by holes along the shell surface while maintaining a hollow cylindrical shape, featuring a compliance equal to 0.097 with an associated relative volume equal to 0.136 (see Fig. 4, top-left). The top-right and bottom-left panels in the figure provide a clipped view of internal and external portions of the final adapted mesh provided by SIMPATY algorithm after 4 mesh updates. The grid is constituted by 8.428.143 elements, with a maximum aspect ratio on the order of 10^3 , mainly concentrated at the material/void interface along the axial direction. On the contrary, isotropic elements are placed in the bulk regions of the structure, according to the hybrid requirement. We emphasize the capability of SIMPATY algorithm to produce shell-like designs, even though the optimization is performed over a volume.

To sum up, the use of an anisotropic mesh adaptation tool enables efficient management of large-scale, high-resolution 3D simulations, with mesh adaptation accounting for only a small portion of the total computational cost (between 5% and 15% for the cases without and with minimum size control, respectively).

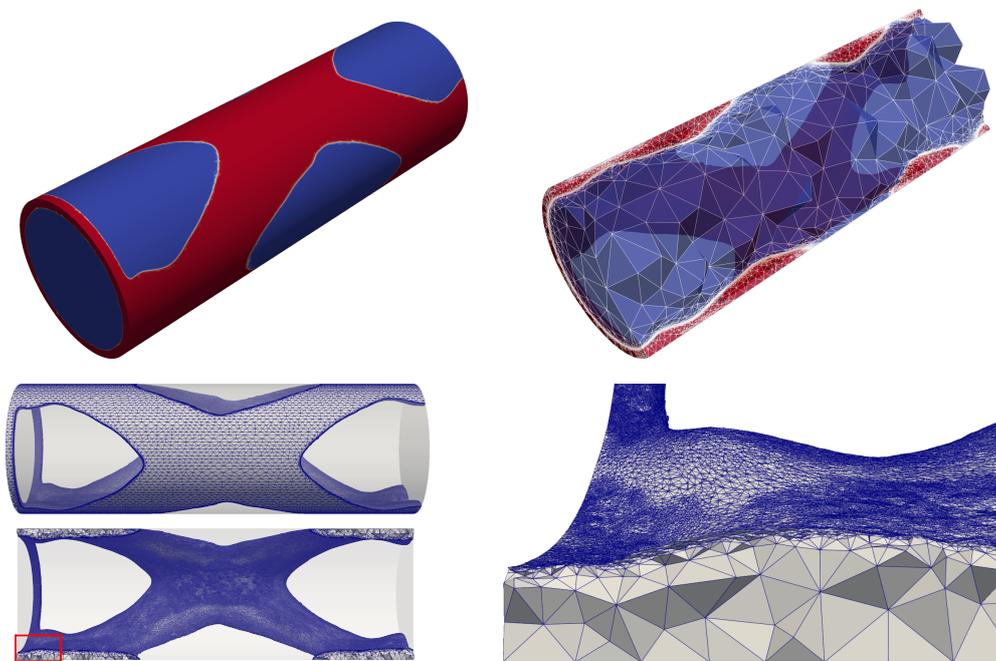


Figure 4: Use case 1 (with minimum size control): final material density distribution (top-left), clipped views of the internal and external rod parts (top-right, bottom-left) and of the material/void interface in correspondence of the red box (bottom-right).

5 Engineering use case 2: design of cellular materials

Cellular materials, or metamaterials, are engineered structures featuring periodic voids within a solid matrix. When these voids follow a regular, repeating pattern, the material is typically referred to as a lattice structure. The geometry of the unit cell governs the material's effective macroscopic behavior through homogenization theory, allowing designers to fine-tune the material properties by manipulating microscale topology.

The rise of additive manufacturing technologies, such as 3D printing, has made it feasible to fabricate such complex structures, encouraging a shift from empirical design toward optimization-based strategies. For this reason, cellular materials have become particularly relevant across various engineering fields, including aerospace, biomedical, and automotive sectors, where their lightweight, tailorable nature enables the creation of high-performance, functionally efficient components.

Metamaterials naturally complement structural optimization (e.g., through TO techniques), where they serve both as design elements at the macro-scale and as infill strategies at the micro-scale. As a result, these materials support multiscale and multimaterial strategies where not only the shape, but also the internal infill evolves across the domain – either by grading the size of a single lattice or by varying the topology itself to adapt to local performance.

5.1 Technical background for use case 2

In this section, we focus on asymptotic homogenization theory, which provides the necessary framework to extend the prototyping activity introduced in the previous use case to the microscale, with the goal of designing periodic cellular materials.

5.1.1 Direct homogenization

The homogenization method is a rigorous asymptotic technique aimed at deriving the effective macroscopic properties of a body, $\Omega \subset \mathbb{R}^3$, composed through the periodic repetition of a microscopic structure, $\omega \subset \mathbb{R}^3$ with $|\omega| \ll |\Omega|$. Homogenization is a reference technique in multiscale modeling, as it systematically upscales the influence of microscale features to the macroscale. This approach enables the replacement of complex heterogeneous media with simpler, equivalent homogeneous models, preserving essential physical behavior at a reduced computational cost.

In the following, we exemplify the homogenization procedure when applied to the linear elasticity equation as in the first use case. To this aim, it is common practice to express the stress, the strain and the stiffness tensors related by the linear constitutive law

$$\sigma(\mathbf{u}) = E\varepsilon(\mathbf{u}), \quad (24)$$

according to Voigt notation, being

$$\begin{aligned}\sigma(\mathbf{u}) &= [\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{23}, \sigma_{13}, \sigma_{12}]^T, \\ \varepsilon(\mathbf{u}) &= [\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, 2\varepsilon_{23}, 2\varepsilon_{13}, 2\varepsilon_{12}]^T, \\ E &= [E_{ijkl}]_{ij,kl \in \mathcal{I}} \in \mathbb{R}^{6 \times 6},\end{aligned}$$

with $\mathcal{I} = \{11, 22, 33, 23, 13, 12\}$. The stiffness tensor E , characterizing both isotropic and anisotropic materials, plays a central role in the homogenization theory [68, 69, 70, 71]. In particular, in (24), tensor E is replaced by the homogenized counterpart, E^H , whose components are computed, for $ij, kl \in \mathcal{I}$, as

$$E_{ijkl}^H = \frac{1}{|\omega|} \int_{\omega} [\sigma(\mathbf{u}^{0,ij}) - \sigma(\mathbf{u}^{*,ij})] : [\varepsilon(\mathbf{u}^{0,kl}) - \varepsilon(\mathbf{u}^{*,kl})] \, d\omega. \quad (25)$$

Vectors $\mathbf{u}^{0,ij}$ are the prescribed test fields, with $\mathbf{u}^{0,11} = [x, 0, 0]^T$, $\mathbf{u}^{0,22} = [0, y, 0]^T$, $\mathbf{u}^{0,33} = [0, 0, z]^T$, $\mathbf{u}^{0,23} = [0, z, 0]^T$, $\mathbf{u}^{0,13} = [z, 0, 0]^T$, $\mathbf{u}^{0,12} = [y, 0, 0]^T$, while $\mathbf{u}^{*,ij}$ are the microscopic corrector fields satisfying the cell problems

$$\int_{\omega} \sigma(\mathbf{u}^{*,ij}) : \varepsilon(\mathbf{v}) \, d\omega = \int_{\omega} \sigma(\mathbf{u}^{0,ij}) : \varepsilon(\mathbf{v}) \, d\omega \quad \forall \mathbf{v} \in V_{\#}, \quad (26)$$

with $V_{\#} = [H_{\#}^1(\omega)]^6$ the space of vector-valued periodic functions on ω .

5.1.2 Inverse homogenization topology optimization

In the inverse homogenization setting, the relation between the body Ω and the microstructure ω is reversed with respect to direct homogenization of Sect. 5.1.1. Instead of computing the macroscopic properties from a given microstructure, we start from the desired behavior of Ω and determine a microstructure ω that, when periodically repeated, reproduces the target effective behavior. We observe that the components of the homogenized stiffness tensor can be used to model the required macroscopic properties. Moreover, the design of the microstructure can be carried out in different ways. Here, we adopt a topology optimization strategy, in continuity with use case 1.

With a view to the design setting in Sect. 5.3, we are interested in engineering lightweight metamaterials exhibiting prescribed stiffness properties or auxetic performance. This goal leads us to particularize problem (19) to the unit cell ω , by choosing: the objective functional as

$$J(\rho) = |\omega|^{-1} \int_{\omega} \rho \, d\omega \quad (27)$$

with ρ varying in $S = H_{\#}^1(\omega; [\underline{\rho}, 1])$, the periodic counterpart of the space S in (19); the state equations as the SIMP variant of the cell problems in (26), being

$$a_{ij,\rho}(\mathbf{u}^{*,ij}, \mathbf{v}) = \int_{\omega} \rho^p \sigma(\mathbf{u}^{*,ij}) : \varepsilon(\mathbf{v}) \, d\omega \quad f_{ij,\rho}(\mathbf{v}) = \int_{\omega} \rho^p \sigma(\mathbf{u}^{0,ij}) : \varepsilon(\mathbf{v}) \, d\omega,$$

with $\mathbf{u}^{*,ij}$ and \mathbf{v} in $V = V_{\#}$, for $ij \in \mathcal{I}$ (i.e., $N = 6$), and $p = 4$; the quantities of interest involved in the box constraints as M selected components, $E_{ijkl,\rho}^H$, of the homogenized stiffness tensor, requiring that

$$\underline{E}_{ijkl}^H \leq E_{ijkl,\rho}^H \leq \overline{E}_{ijkl}^H \quad (28)$$

for specific choices of indices $ij, kl \in \mathcal{I}$, where $E_{ijkl,\rho}^H$ is the SIMP variant of E_{ijkl}^H in (25), being

$$E_{ijkl,\rho}^H = \frac{1}{|\omega|} \int_{\omega} \rho^p [\sigma(\mathbf{u}^{0,ij}) - \sigma(\mathbf{u}^{*,ij})] : [\varepsilon(\mathbf{u}^{0,kl}) - \varepsilon(\mathbf{u}^{*,kl})] \, d\omega.$$

5.2 Integration of anisotropic mesh adaptation into use case 2

Inverse homogenization topology optimization and anisotropic mesh adaptation are combined in the microSIMPATY algorithm [72], building upon the implementation strategies originally developed for the macroscale SIMPATY framework. In this setting, transitioning from the macro- to the microscale is relatively straightforward for 2D metamaterial design. In contrast, the extension to 3D introduces significant challenges, primarily due to the imposition of periodic boundary conditions, which must now be enforced not only along edges, as in 2D, but also across faces and corners of the computational domain. This added complexity is reflected in the scarcity of mesh generation tools capable of handling periodicity alongside anisotropic adaptation in three dimensions.

To address this gap, a dedicated 3D tool for generating adapted periodic meshes has been developed in [73]. Here, the authors adjust the adaptive strategy described in Sect. 3 to account for mesh periodicity, specifically by ensuring that vertices and elements match across opposite faces of the cell ω . The proposed approach consists of the following steps: starting from the unconstrained anisotropic mesh adaptation in Sect. 3, the anisotropically adapted mesh \mathcal{T}_h^* is partitioned into a boundary shell, $\mathcal{T}_h^{*,\text{shell}} = \{K \in \mathcal{T}_h^* : K \cap \partial\omega \neq \emptyset\}$, and an interior region $\mathcal{T}_h^{*,\text{int}} = \mathcal{T}_h^* \setminus \mathcal{T}_h^{*,\text{shell}}$. Periodicity is then enforced along the boundary $\partial\omega$ by matching the vertices and triangles on each pair of opposite faces through targeted local remeshing operations. This identifies the periodic boundary mesh $\mathcal{T}_{h,\#}^{*,\text{shell}}$. Finally, the global adapted periodic mesh is obtained by consistently merging $\mathcal{T}_h^{*,\text{int}}$ and $\mathcal{T}_{h,\#}^{*,\text{shell}}$. The effectiveness of microSIMPATY algorithm has been verified in several application contexts (see, e.g., [63, 48, 74, 47]).

5.3 Simulation outcomes for use case 2

The design setting addressed in this section exactly coincides with the one of Sect. 5.1.2, for two different choices of the constraints in (28). In particular, we identify the lower and upper bounds with quantities

$$\underline{E}_{ijkl}^H = E_{ijkl}^G(1 - \delta), \quad \overline{E}_{ijkl}^H = E_{ijkl}^G(1 + \delta), \quad (29)$$

imposing the matching between E_{ijkl}^H and the goal value E_{ijkl}^G , up to a tolerance δ , here set to 5%.

We choose the unitary cube $\omega = (0, 1)^3$ and we select the material properties through $E_Y = 350$ and $\nu = 0.281$. As initial density, we set $\rho = 1 - |(\sin(\pi x) \sin(\pi y) \sin(\pi z))|$, while we start the adaptive procedure picking an initial isotropic structured mesh, $\mathcal{T}_h^{(0)}$, consisting of 750.000 tetrahedra. The mesh adaptation parameters are selected to ensure a minimum element size of 0.005 and a maximum size of 1. A uniform element size of $h_{\text{iso}} = 0.15$ is used for the initial hybrid mesh generation. The adaptation procedure is further constrained by a tolerance of $\text{TOL} = 17.5$, a maximum number of iterations $N_{\text{max}} = 5$, and a relative mesh cardinality variation threshold set to $\Delta_S = 0.05^3$.

As a first assessment, we focus on designing an isotropic microstructure, which requires controlling the diagonal components E_{1111}^H , E_{2222}^H , and E_{3333}^H of the homogenized stiffness tensor. The corresponding target values in (29) are

$$E_{1111}^G = E_{2222}^G = E_{3333}^G = 64.899.$$

The microSIMPATY algorithm stops after 4 mesh updates. Consistently with the literature [75], the design procedure provides as an output a closed shell-like cell with a volume of 0.251. Figure 5 shows the associated anisotropically adapted mesh consisting of 318.223 elements. In particular, the two close-up views confirm the effectiveness of anisotropic mesh adaptation, which generates highly stretched tetrahedra (the maximum aspect ratio being equal to 583.992) aligned with the wall directions, thus allowing for a very sharp resolution of the thin shell. From a computational perspective, we observe that the time required for the mesh adaptation module is not prohibitive, accounting for approximately 20% of the total CPU time.

The second verification focuses on the design of a microstructure with auxetic behavior. To this end, the optimization procedure includes constraints on some of the off-diagonal components of the homogenized stiffness tensor, in addition to the diagonal ones considered in the previous case. In particular, the goal values in (29) are identified with

$$E_{1111}^G = E_{2222}^G = E_{3333}^G = 14.800, \quad E_{1122}^G = E_{1133}^G = E_{2233}^G = -1.202.$$

The optimized microstructure is designed by microSIMPATY algorithm in 4 mesh updates (see Fig. 6) and is characterized by a relative volume equal to 0.336. The introduction of a control on the off-diagonal entries of E^H is responsible for an extremely complex topology in ω . In particular, the shape of the cavity opening onto the top and bottom faces can be associated with the auxetic behavior enforced during the design process. The adapted mesh supporting the structure in Fig. 6 consists of 1.234.829 tetrahedra, predominantly concentrated along the material/void interface. The irregular geometry of this boundary justifies both the increased number of elements and the reduced maximum aspect ratio (equal to 338.599) compared to the previous design configuration. Nevertheless, this increase in mesh cardinality does not affect the relative computational cost of mesh adaptation, which remains approximately 20% of the total simulation time.

³The results in the following are carried out on the same high performance computing node and by using the same solver and mesh generator tools as in use case 1.

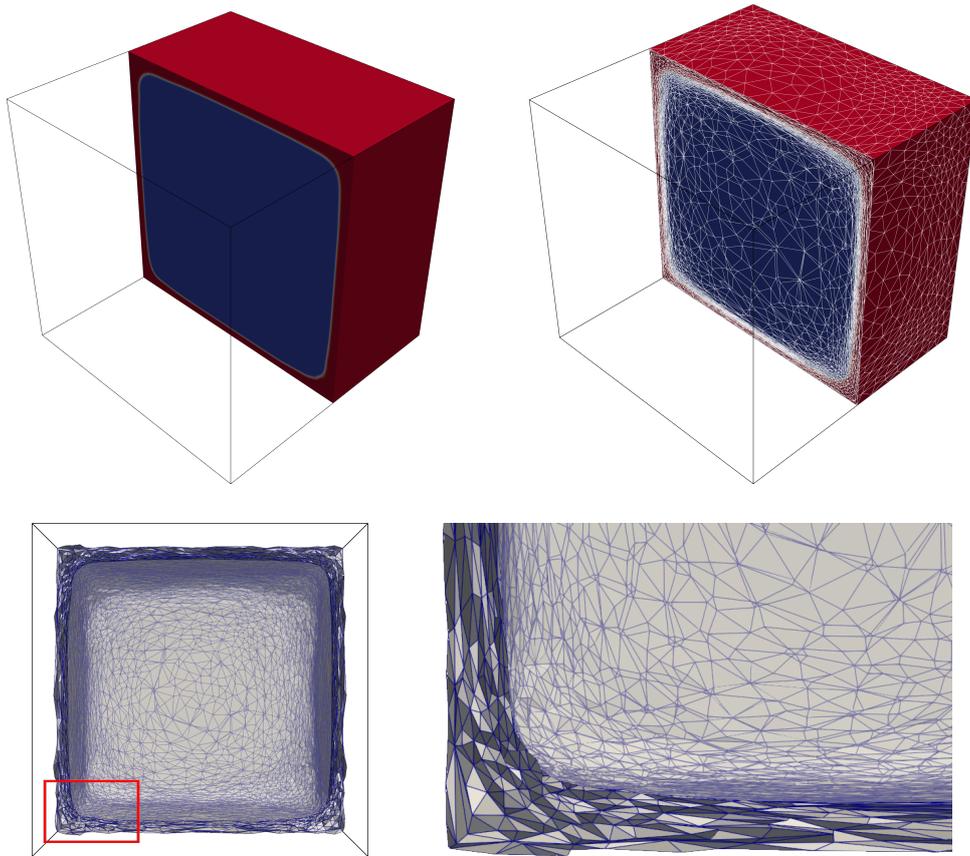


Figure 5: Use case 2 (isotropic design): clipped views of the density (top-left) overlapped to the anisotropic adapted mesh (top-right); close-up view of the anisotropic adapted mesh (bottom-left) together with a detail of the material/void interface in correspondence of the red box (bottom-right).

We exploit this second design to visualize the lattice structure resulting from the periodic repetition of the optimized cell. In Fig. 6, bottom-left, bottom-center panels, a $2 \times 2 \times 2$ -cell lattice is shown, where different colors represent a potential multimaterial configuration. The close-up view in the bottom-right panel confirms the effectiveness of the 3D algorithm developed to handle the periodicity of the adapted mesh when transitioning from one cell to its neighbors, guaranteeing full geometric consistency and compatibility across interfaces.

6 Generalization to a time-dependent setting

Many physical, biological, and engineering phenomena are governed by time-dependent PDEs, where the evolution of the system depends not only on spatial variables but also on time. In such contexts, the efficient and accurate resolution of transient fea-

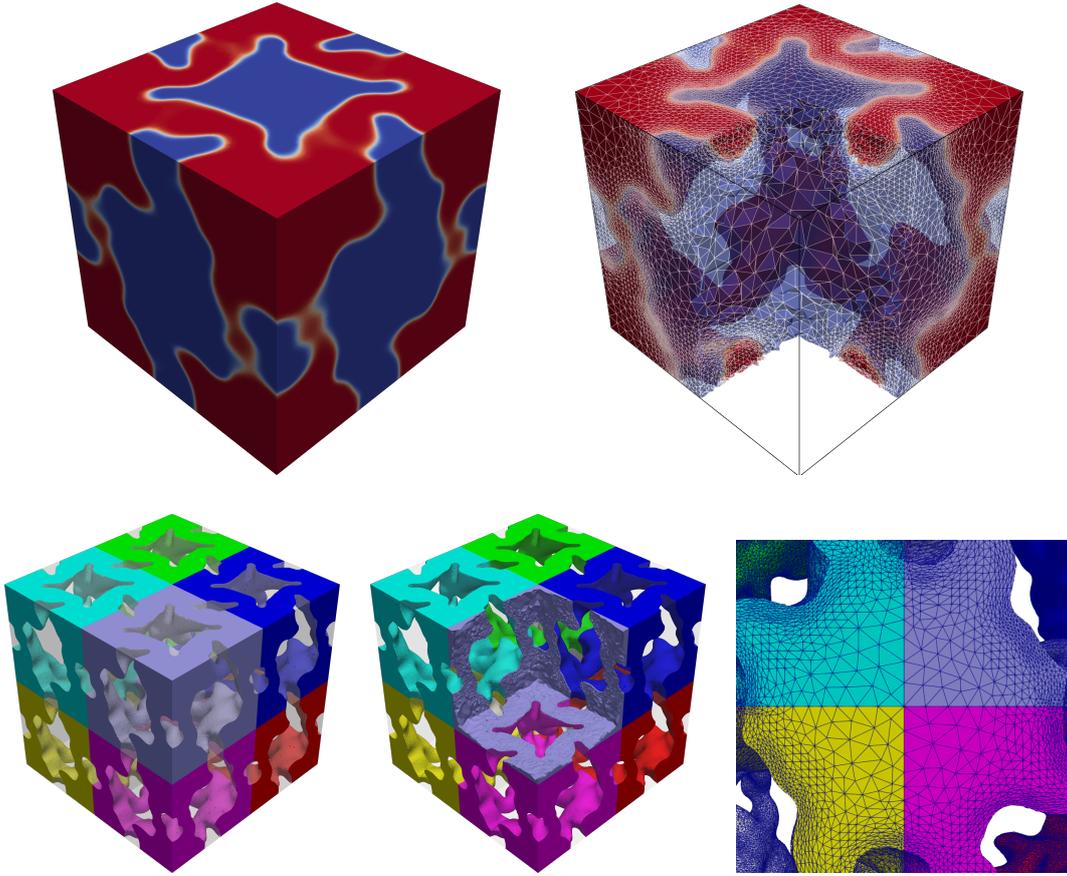


Figure 6: Use case 2 (auxetic design): density (top-left) overlapped to a clipped view of the anisotropic adapted mesh (top-right); $2 \times 2 \times 2$ -cell lattice (bottom-left), corresponding clipped view (bottom-center) and detail of the periodic interface (bottom-right).

tures is crucial. Classical discretization approaches often employ uniform time-stepping schemes. This strategy may result in either an under-resolution of rapid dynamics or an unnecessary computational burden during slow temporal evolution. To address these challenges, the development of numerical methods that adapt the computational mesh both in space and in time has gained increasing attention, see for instance [76, 77, 78]. However, while spatial adaptivity is well-established, also in many commercial tools, temporal adaptivity is less widespread, despite its potential in handling problems with strong time variability.

6.1 Time-step adaptive procedure: theoretical background

This section introduces the mathematical foundation we adopt to set a temporal adaptive strategy in the context of time-dependent PDEs. We preserve the notation z and z_h as in Sect. 2 to denote the solution to a general unsteady PDE model and the corresponding

finite element approximation. Thus, for any time t varying in a certain temporal window $I = [0, T]$, we have $z = z(\mathbf{x}, t) \in Z$ and $z_h = z_h(\mathbf{x}, t) \in Z_h$, with Z and Z_h the same (spatial) function spaces in Sect. 2.2 and with $\mathbf{x} \in \Omega$.

The temporal adaptive strategy we propose aims to construct a partition of the time domain I based solely on the prescribed final time T , as the number and size of time steps are not known in advance. In particular, starting from a given time t^n , we determine the next time step $\Delta t^{n+1} = t^{n+1} - t^n$, thereby defining the upcoming time interval $I_{n+1} = [t^n, t^{n+1}]$. As a consequence, the total number of the time steps will only be known at the end of the adaptive procedure. Accordingly, we must ensure that the final time step does not exceed the endpoint T .

As for the spatial case, the adaptive selection of the time step relies on a dedicated mathematical tool, namely, an a posteriori error estimator. Of course, since time is a one-dimensional quantity, the adaptive strategy is, by definition, carried out in an isotropic context. Following [26], we adopt a recovery-based a posteriori error estimator to dynamically adjust the time step length at each instant. With reference to the generic scheme in (4), we identify the differential operator Θ with the first order temporal partial derivative, $\partial/\partial t$, and the spatial domain Ω with the time interval I_n . Thus, at each vertex V of the spatial mesh \mathcal{T}_h^n associated with time t_n , we generalize the estimator in (5) to estimate the seminorm

$$|e_z(V, t)|_{H^1(I_n)}^2 = \int_{I_n} \left| \frac{\partial z}{\partial t}(V, t) - \frac{\partial z_h}{\partial t}(V, t) \right|^2 dt$$

through the quantity

$$[\eta_n(V)]^2 = \int_{I_n} \left| \mathcal{R}_{\partial t}(z_h^n)(V) - \delta_-(z_h^n)(V) \right|^2 dt, \quad (30)$$

with z_h^n the finite element approximation z_h at time t_n , where $\mathcal{R}_{\partial t}(z_h^n)$ is the recovered time derivative at time t^n , and $\delta_-(z_h^n) = (z_h^n - z_h^{n-1})/\Delta t^n$ is the backward finite difference approximation of $\partial z_h/\partial t$ at t^n . Specifically, following [26], we compute quantity $\mathcal{R}_{\partial t}(z_h^n)$ as the derivative of the quadratic polynomial interpolating the pairs of data (t^{n-2}, z_h^{n-2}) , (t^{n-1}, z_h^{n-1}) , and (t^n, z_h^n) . Finally, we remove the spatial dependence in (30) by averaging the vertex contributions across the elements of the grid \mathcal{T}_h^n . This leads to introduce the global error estimator over I_n , given by

$$\eta_{n,T}^2 = \frac{1}{4} \sum_{K \in \mathcal{T}_h^n} \sum_{V \in K} [\eta_n(V)]^2. \quad (31)$$

To predict the next time step Δt^{n+1} , inspired by the strategy adopted for the spatial adaptive procedure, we first scale and adimensionalize (31), thereby defining the new estimator

$$\tilde{\eta}_{n,T}^2 = \tilde{T} [\Delta t^n \rho_{n,T}]^2, \quad (32)$$

with

$$\rho_{n,T}^2 = \frac{1}{4[\Delta t^n]^2} \sum_{K \in \mathcal{T}_h^n} \sum_{V \in K} [\eta_n(V)]^2 \quad (33)$$

and \tilde{T} a problem-dependent characteristic temporal scale. Then, we prescribe a desired local accuracy, τ_ℓ , for the estimator $\tilde{\eta}_{n,T}$, since assigning a global tolerance is not feasible a priori, as the total number of time intervals is only determined at the end of the adaptive process. This leads to the condition

$$\tilde{T}[\Delta t^n \rho_{n,T}]^2 = \tau_\ell^2, \quad (34)$$

which enforces the equidistribution of the error in time over the entire simulation time window. Finally, relation (34) is inverted to predict the next time-step length as

$$\Delta t^{n+1} = \sqrt{\frac{1}{\tilde{T}}} \frac{\tau_\ell}{\rho_{n,T}^2}. \quad (35)$$

The implementation of time adaptation is simpler than its spatial counterpart, due to the one-dimensional nature of the problem. Algorithm 2 outlines a pseudo-code that sketches the required steps, which essentially reduce to the sequential combination of the routine `time_estimator`, used to compute the quantities defined in (31)–(33), and the routine `adapt_timestep`, which implements the update formula in (35). Notice that notation $\{z_h\}$ refers to the three consecutive values of z_h required to compute the recovered time derivative $\mathcal{R}_{\partial t}(z_h)$. Moreover, as for the mesh adaptation in space, we constrain the selection of the time step through a minimum (Δt_{\min}) and a maximum (Δt_{\max}) value for Δt .

Algorithm 2 : time-step adaptation

```
function  $[\Delta t^*, \eta_T^*] = \text{time\_adaptation}(\{z_h\}, \mathcal{T}_h, \Delta t, \tau_\ell, \tilde{T}, \Delta t_{\min}, \Delta t_{\max})$ 
1:  $\eta_T = \text{time\_estimator}(\{z_h\}, \mathcal{T}_h, \Delta t)$ ;
2:  $\Delta t^* = \text{adapt\_timestep}(\eta_T, \tau_\ell, \tilde{T}, \Delta t, \Delta t_{\max}, \Delta t_{\min})$ ;
3:  $\eta_T^* = \text{time\_estimator}(\{z_h\}, \mathcal{T}_h, \Delta t^*)$ ;
```

In the next section, we will explore a possible strategy to combine time and space adaptation, a task that is not necessarily straightforward or uniquely defined.

6.2 Space-time adaptive procedure

In this section, we propose a simple strategy to combine space and time mesh adaptation. At each time t^n , the spatial mesh and the time step are updated independently in a sequential manner, each starting from the corresponding error control criterion. Unlike other approaches in the literature (see, for instance, [79]), this method does not enforce a global tolerance, but rather ensures accuracy separately in space and time.

To streamline the pseudo-code, we focus solely on the transition from time t^n to t^{n+1} , for $n > 1$ ⁴. Specifically, we assume that we are at time t^n on the mesh \mathcal{T}_h^n , and we aim

⁴Note that the first two time steps are kept constant and equal to Δt^0 , since the time error estimator requires information from three consecutive time levels.

to predict both the next time level t^{n+1} (i.e., the time step Δt^{n+1}) and the corresponding spatial mesh \mathcal{T}_h^{n+1} . For simplicity, we omit the explicit time dependence and introduce the following shorthand notation: $\mathcal{T}_h^n \rightarrow \mathcal{T}_h$, $\mathcal{T}_h^{n+1} \rightarrow \mathcal{T}_h^*$, $\Delta t^n \rightarrow \Delta t$, $\Delta t^{n+1} \rightarrow \Delta t^*$, $z_h^n \rightarrow z_h$.

Algorithm 3 : space-time adaptation

```

function [ $\mathcal{T}_h^*$ ,  $\eta_A^*$ ,  $\Delta t^*$ ,  $\eta_T^*$ ] = spacetime_adaptation( $\{z_h\}$ ,  $\mathcal{T}_h$ ,  $\Delta t$ , TOL,  $N_{\max}$ ,  $\Delta_S$ ,
Delta,  $\tau_\ell$ ,  $\tilde{T}$ ,  $h_{\min}$ ,  $h_{\max}$ ,  $\Delta t_{\min}$ ,  $\Delta t_{\max}$ )
1: [ $\Delta t^*$ ,  $\eta_T^*$ ] = time_adaptation( $\{z_h\}$ ,  $\mathcal{T}_h$ ,  $\Delta t$ ,  $\tau_\ell$ ,  $\tilde{T}$ ,  $\Delta t_{\min}$ ,  $\Delta t_{\max}$ );
2:  $\eta_A = \text{space\_estimator}(z_h, \mathcal{T}_h^b)$ ;
3: if  $\eta_A < 0.99 \cdot \text{TOL} \mid \eta_A > 1.01 \cdot \text{TOL}$  then
4:   [ $\mathcal{T}_h^*$ ,  $\eta_A^*$ ] = space_adaptation( $z_h$ ,  $\mathcal{T}_h$ , TOL,  $N_{\max}$ ,  $\Delta_S$ , Delta,  $h_{\min}$ ,  $h_{\max}$ );
5: else
6:    $\mathcal{T}_h^* = \mathcal{T}_h$ ;
7:    $\eta_A^* = \eta_A$ ;
8: end if

```

It is worth noting that time adaptation phase cannot be neglected, as it serves as a predictive tool for determining the next instant. However, the associated computational cost remains negligible since dealing with a one-dimensional problem. Conversely, spatial adaptation, which is more computationally demanding, is triggered only when the solution deviates sufficiently from the prescribed tolerance, in order to optimize the overall computational cost.

7 Engineering use case 3: fluid dynamics modeling

Fluid dynamics phenomena are widely encountered in engineering and often require time-dependent simulations, which can be computationally demanding. Examples include applications in aerospace, renewable energy, biomedicine, and environmental modeling. To address the computational issues inherent in fluid dynamics modeling, various mathematical tools can be employed, including reduced order modeling (see, e.g., [80]), machine-learning algorithms (see, e.g., [81]), and adaptive techniques. Among these, mesh adaptation strategies in space and/or time have proven particularly effective for accurately and efficiently capturing multiscale and heterogeneous behaviours across both spatial and temporal dimensions (we refer, e.g., to [76, 82, 83]).

The objective of this section is to assess how the space-time mesh adaptation procedure introduced in the previous section can provide computational advantages when applied to the reference model in fluid dynamics simulations, namely to the unsteady Navier–Stokes equations.

7.1 Technical background for use case 3

Navier-Stokes equations provide a rigorous framework for modeling fluid flows across a broad spectrum of physical scenarios [84, 85]. For simplicity, we focus on a 2D setting. We aim to determine the velocity, $\mathbf{u} = [u_1, u_2]^T$, and the pressure field, p , which satisfy the initial-boundary value problem

$$\left\{ \begin{array}{ll} \frac{\partial \mathbf{u}}{\partial t}(\mathbf{x}, t) + (\mathbf{u}(\mathbf{x}, t) \cdot \nabla) \mathbf{u}(\mathbf{x}, t) - \nu \Delta \mathbf{u}(\mathbf{x}, t) + \frac{1}{\rho} \nabla p(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) & \text{in } \mathcal{D} \\ \nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0 & \text{in } \mathcal{D} \\ \mathbf{u}(\mathbf{x}, t) = \boldsymbol{\varphi}(\mathbf{x}) & \text{on } \partial\Omega_D \times I \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}}(\mathbf{x}, t) - \frac{1}{\rho} p(\mathbf{x}, t) \mathbb{I} \mathbf{n} = \boldsymbol{\psi}(\mathbf{x}) & \text{on } \partial\Omega_N \times I \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) & \text{in } \Omega, \end{array} \right. \quad (36)$$

with $\mathcal{D} = \Omega \times I$ the space-time domain, being $\Omega \subset \mathbb{R}^2$ the spatial region of interest, $I = (0, T]$ the considered time window, and (\mathbf{x}, t) the generic point in \mathcal{D} . Here, $\nu = \mu/\rho$ is the kinematic viscosity, defined as the ratio between the dynamic viscosity, μ , and the fluid density, ρ . Vector \mathbf{f} represents an external force per unit mass. Functions $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ prescribe Dirichlet and Neumann boundary conditions on the domain boundary portions $\partial\Omega_D$ and $\partial\Omega_N$, respectively with $\partial\Omega_D \cup \partial\Omega_N = \partial\Omega$ and $\partial\Omega_D \cap \partial\Omega_N = \emptyset$. Moreover, $\mathbb{I} \in \mathbb{R}^{2 \times 2}$ denotes the identity matrix, \mathbf{n} the unit outward normal vector to $\partial\Omega$, and \mathbf{u}_0 the initial velocity field. In this setting, the boundary data, $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$, are assumed to be time-independent. We also consider an incompressible flow, with constant and strictly positive density ρ . Finally, the dependence of all functions on space and time is omitted unless otherwise specified.

With a view to discretizing equations (36), we first introduce the associated weak formulation, so that, for any time t in I , we look for $\mathbf{u} = \mathbf{u}(t) \in V$ and $p = p(t) \in Q$ such that

$$\begin{aligned} & \int_{\Omega} \left\{ \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} + [(\mathbf{u} \cdot \nabla) \mathbf{u}] \cdot \mathbf{v} + \nu \nabla \mathbf{u} : \nabla \mathbf{v} - \frac{1}{\rho} p \nabla \cdot \mathbf{v} - q \nabla \cdot \mathbf{u} \right\} d\Omega \\ & = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega + \int_{\Gamma_N} \boldsymbol{\psi} \cdot \mathbf{v} d\Gamma_N \end{aligned} \quad (37)$$

for any $\mathbf{v} \in V$ and $q \in Q$, with $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0$, $\mathbf{u}(\mathbf{x}, t) = \boldsymbol{\varphi}(\mathbf{x})$ for $\mathbf{x} \in \partial\Omega_D$ and for any t , and where $V = [H^1(\Omega)]^2$ and $Q = L^2(\Omega)$.

To derive the approximate counterpart of (37), we first partition domain Ω by introducing a conforming triangulation, $\mathcal{T}_h = \{K\}$, with mesh size h , and the time window I by means of the partition $\{t^0 = 0, t^1, \dots, t^f = T\}$, so that $I_n = [t^n, t^{n+1}]$ denotes the generic interval of length $\Delta t^{n+1} = t^{n+1} - t^n$. To discretize the space we resort to the finite element method, while we employ a semi-implicit integration scheme to discretize the time [37]. Thus, for each $n = 0, \dots, f-1$, we search $\mathbf{u}_h^{n+1} = \mathbf{u}_h(t^{n+1}) = [u_{h,1}^{n+1}, u_{h,2}^{n+1}]^T \in V_h \subset V$ and

$p_h^{n+1} = p_h(t^{n+1}) \in \mathcal{Q}_h \subset \mathcal{Q}$, with $\dim(V_h) < +\infty$ and $\dim(\mathcal{Q}_h) < +\infty$, such that

$$\begin{aligned} & \int_{\Omega} \left\{ \frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n}{\Delta t^{n+1}} \cdot \mathbf{v}_h + [(\mathbf{u}_h^n \cdot \nabla) \mathbf{u}_h^{n+1}] \cdot \mathbf{v}_h + \nu \nabla \mathbf{u}_h^{n+1} : \nabla \mathbf{v}_h \right. \\ & \left. - \frac{1}{\rho} p_h^{n+1} \nabla \cdot \mathbf{v}_h - q_h \nabla \cdot \mathbf{u}_h^{n+1} \right\} d\Omega = \int_{\Omega} \mathbf{f}^{n+1} \cdot \mathbf{v}_h d\Omega + \int_{\Gamma_N} \psi_h \cdot \mathbf{v}_h d\Gamma_N \end{aligned} \quad (38)$$

for any $\mathbf{v}_h \in V_h$ and $q_h \in \mathcal{Q}_h$, with $\mathbf{u}_h(\mathbf{x}, 0) = \mathbf{u}_{h,0}$, $\mathbf{u}_h(\mathbf{x}, t) = \varphi_h(\mathbf{x})$ for $\mathbf{x} \in \partial\Omega_D$ and for any t , $\mathbf{f}^{n+1} = \mathbf{f}(t^{n+1})$, and where functions φ_h , ψ_h , and $\mathbf{u}_{h,0}$ are suitable approximations of φ , ψ , and \mathbf{u}_0 in V_h , respectively. Concerning the choice of V_h and \mathcal{Q}_h , we adopt the classical Taylor–Hood finite element pair, which satisfies the inf-sup (Ladyzhenskaya–Babuška–Brezzi) condition [86]. This choice employs continuous piecewise quadratic functions for the velocity and linear ones for the pressure, namely $V_h = [X_h^2]^2$, $\mathcal{Q}_h = X_h^1$, with X_h^r defined according to (3). Finally, to ensure the stability of the time integration scheme, the spatial mesh size h and the time step Δt are required to satisfy the classical CFL condition [87], which properly relates the temporal and spatial discretizations to maintain numerical stability.

7.2 Integration of space-time mesh adaptation into use case 3

The space-time adaptation strategy described in Algorithm 3 can be applied to the Navier–Stokes equations. In particular, the driving quantity used in the estimators η_A and $\eta_{n,T}$, defined in (10) and (31), is taken as the magnitude of the velocity field at time t^n , i.e., $z_h = |\mathbf{u}_h^n|$, although other choices are possible. For example, in [88], the authors explore various alternatives for the driving quantity, including vector-valued fields or combinations of scalar quantities, in the context of turbulent flow simulations.

Several options are available for selecting the characteristic temporal scale \tilde{T} in (32), noting that, dimensionally, this quantity represents a time. For instance, in [26], \tilde{T} is defined either as the previous time step or as the ratio between a characteristic length of the domain and the fluid celerity. In this work, we set $\tilde{T} = \Delta t / \text{Re}$, namely as the ratio between the last computed time step and the Reynolds number Re .

As a finer-level consideration, relevant to more advanced modeling scenarios such as fluid dynamics in challenging regimes, it may be necessary to modify the time adaptation strategy proposed in Algorithm 2. Drawing inspiration from spatial adaptation, one can embed the direct time step determination in (35) into an iterative loop. Specifically, following the notation introduced in Algorithm 2, the characteristic temporal scale can be redefined as $\tilde{T} = \Delta t^{*,(k)} / \text{Re}$, where $\Delta t^{*,(k)}$ denotes the k -th candidate for the new time step Δt^* , given by

$$\Delta t^{*,(k)} = \sqrt{\frac{\text{Re}}{\Delta t^{*,(k-1)}} \frac{\tau_\ell}{\rho_{n,T}^2}}. \quad (39)$$

While this approach increases the computational cost due to the additional iterative loop, it allows for enhanced robustness. In this case, an extra stopping criterion must be introduced, for instance based on the stagnation of the time error estimator.

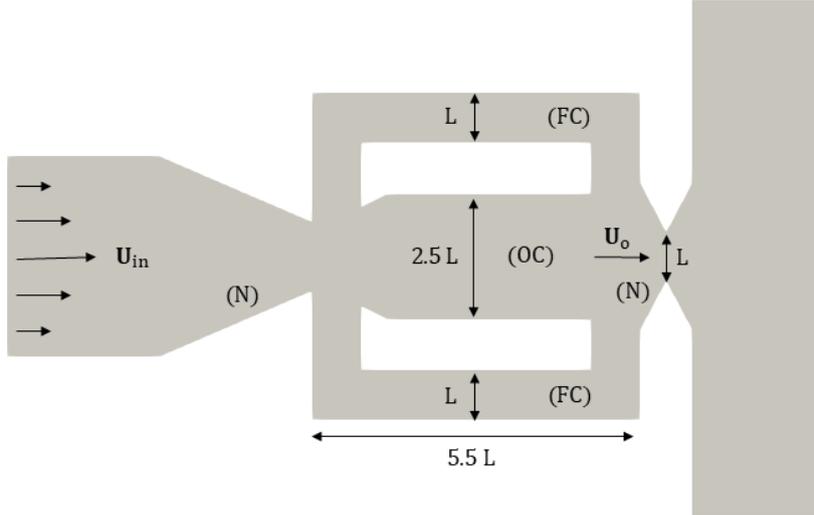


Figure 7: Use case 3: not to scale sketch of a fluidic oscillator.

7.3 Simulation outcomes for use case 3

The space-time adaptive strategy introduced in Sect. 6.2 is here applied to the specific case of a fluidic oscillator. This device generates oscillating flow patterns without involving any moving mechanical components, relying solely on its geometric configuration. The absence of moving parts offers several advantages, including lower manufacturing costs, increased durability, and enhanced resistance to aggressive working fluids. Fluidic oscillators find application in a range of fields, such as combustion [89] and flow control systems [90]. For a comprehensive overview of their development, we refer the reader to [91].

Figure 7 shows the geometry of a standard fluidic oscillator which includes the oscillation chamber (OC), the feedback channels (FC), and the nozzles (N). The fluidic oscillator geometry is defined parametrically, with all dimensions expressed as multiples of a reference length L . The domain Ω in Fig. 7 measures $40L$ in length and $45L$ in height, as in [92], with an inlet nozzle narrowing from $2L$ to $0.7L$. While L is an input parameter, it does not affect the flow behavior, which depends only on the preserved geometric ratios. Here, we set $L = 1$ for simplicity.

We approximate equations (36) in such a geometry on the time interval $(0, T]$ with $T = 3.173$, after setting $\rho = 1$, $\nu = 0.1$, $\mathbf{f} = \mathbf{0}$. Concerning the boundary data, we choose $\partial\Omega_N = \emptyset$, while assigning $\varphi = \mathbf{U}_{in}$ on the inflow boundary, with \mathbf{U}_{in} a parabolic profile, and $\varphi = \mathbf{0}$ on the remaining part of $\partial\Omega$. In particular, the inlet velocity is set such that the Reynolds number at the outlet nozzle satisfies $\text{Re} = U_o L / \nu = 5000$, where U_o denotes the resulting average outlet velocity (see Fig. 7). This choice ensures that the proposed methodology operates under the same flow regime as the reference study [92]. Furthermore, the selected final time is sufficient to capture the full development of the flow dynamics.

We run Algorithm 3 on this use case⁵, starting from an isotropic uniform initial mesh, \mathcal{T}_h^0 , consisting of 54.353 elements, and with an initial time step $\Delta t^1 = 0.0015$, which is preserved for the first two iterations of the time adaptation loop. For spatial mesh adaptation, the parameters are set as $\text{TOL} = 15$, $N_{\max} = 5$, $\Delta_S = 0.001$, $\text{Delta} = \text{'estimator'}$, $h_{\min} = 0.01$, and $h_{\max} = 5$. The time-step prediction is driven by the parameters $\tau_\ell = 0.65$ and $\tilde{T} = \Delta t^1 / \text{Re}$, with the bounds $\Delta t_{\min} = \Delta t^1$ and $\Delta t_{\max} = 2$ chosen to ensure that the CFL condition is always satisfied.

Figure 8 displays the velocity magnitude colormap along with the corresponding anisotropically adapted mesh during the transient regime. The mesh effectively captures the flow gradients, ensuring high-resolution representation, even in regions where the mesh remains relatively coarse, while maintaining a contained computational cost. In particular, as shown in the left panel of Fig. 9, the mesh cardinality, after an initial drop due to the deliberately fine initial discretization, stabilizes on average over time. However, a gradual increase in the number of tetrahedra can be observed as the flow develops more extensively within the final chamber. Concerning the adaptive selection of the time step, an oscillatory behavior of Δt^{n+1} is observed, without convergence to a steady value. This is clearly illustrated in the right panel of Fig. 9, where the absence of a settling trend is confirmed by the least-squares approximation, shown as a dashed line.

The reliability of the space-time mesh adaptation procedure is confirmed by evaluating the Strouhal number characterizing the flow oscillation frequency. The computed value on the spatially and temporally adapted mesh is 0.0214, which compares well with the reference value of 0.02 reported in the literature [92]. Additional quantitative details of the simulation are summarized in the first row of Table 1, which includes the final mesh cardinality, the maximum aspect ratio $s_K = \lambda_{1,K} / \lambda_{2,K}$, the number of time steps, and the total CPU time.

	$\#\mathcal{T}_h^f$	$\max s_K$	$\#\Delta t$	CPU time
Space-time mesh adaptation	10795	43	595	54 min
Space adaptation	10796	39	2524	225 min

Table 1: Use case 3: quantitative information for the spatial-temporal and the spatial only mesh adaptation.

To highlight the benefits of time adaptation, we repeat the previous simulation by disabling line 1 in Algorithm 3 and fixing the time step to $\Delta t = \Delta t^1$. The second row of Table 1 reports the key output data from this run. A comparison between the two configurations confirms the superior performance of the full adaptation strategy, resulting in approximately a 75% reduction in both the number of time steps and total CPU time.

⁵The simulations are performed on a desktop computer equipped with a 12th Gen Intel[®] Core[™] i7-1260P processor, integrated Mesa Intel[®] Graphics, and 16 GB of RAM. The computational setup relies on FreeFEM, employing MUMPS as the direct linear solver and BAMG for spatial mesh adaptation.

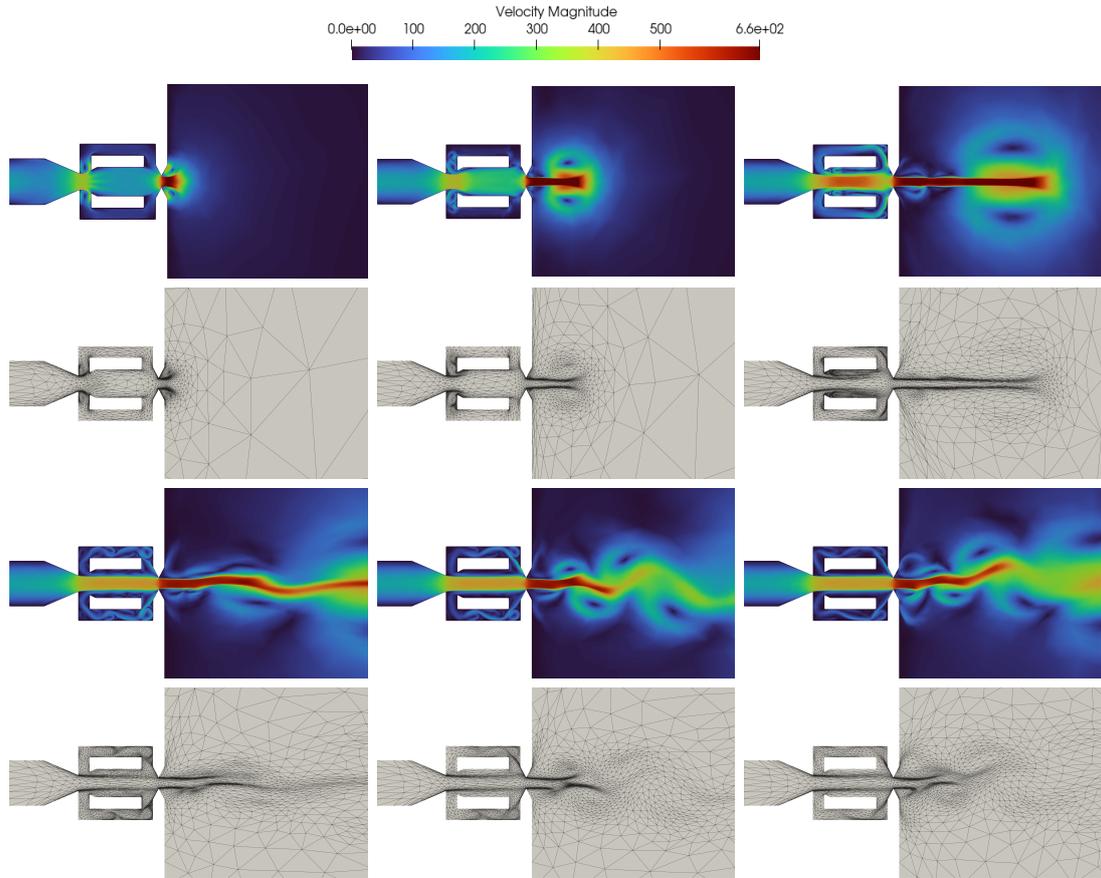


Figure 8: Use case 3: colormap of the velocity magnitude and associated anisotropic adapted mesh at $t=0.0031$, 0.0201 , 0.0582 (first and second row, left-right) and at $t = 0.1118$, 0.1508 , 0.2230 (third and fourth row, left-right).

8 To sum up

This chapter presents a comprehensive framework for anisotropic mesh adaptation driven by recovery-based a posteriori error estimators. Starting from the theoretical formulation in the steady setting, the methodology is successively extended to time-dependent problems via a space-time adaptive strategy that couples spatial mesh adaptation with an adaptive choice of the time step.

The effectiveness of the proposed approach is assessed through three representative engineering applications. In the steady regime, the methodology is applied to both structural topology optimization and cellular material design. In these contexts, anisotropic adaptation enables sharp resolution of interfaces and fine-scale features, using highly stretched elements aligned with the material density gradients. The adoption of a hybrid meshing strategy, which combines anisotropic elements near interfaces with isotropic tetrahedra of prescribed size in bulk regions, proves particularly effective, striking a balance between

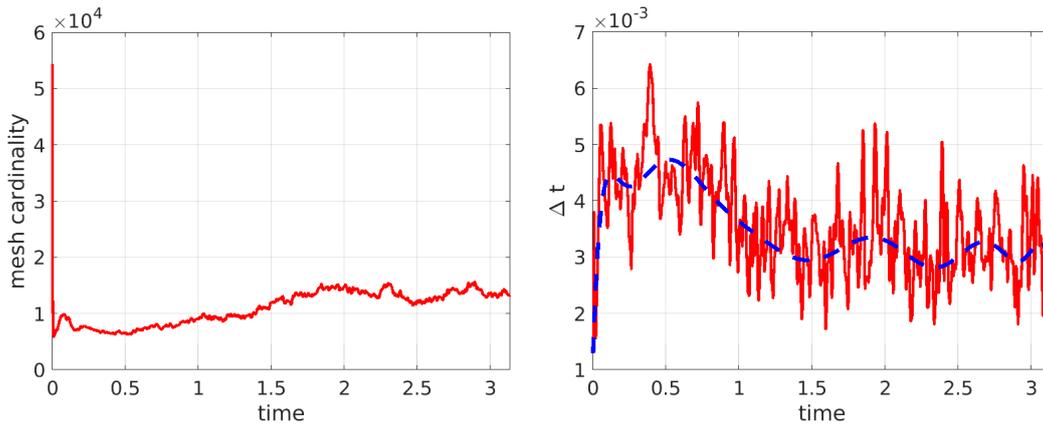


Figure 9: Use case 3: evolution of the mesh cardinality (left) and of the time step (right) over time.

computational accuracy and efficiency, and reliability in terms of mechanical analysis. These features translate into substantial reductions in degrees of freedom and CPU time, with mesh adaptation typically accounting for less than 20% of the total computational cost, while guaranteeing a high quality of the solution and of the associated structural performance.

In the unsteady case, focused on fluid dynamics, the space-time extension allows the method to track transient and localized phenomena through simultaneous adaptation in space and time. The adaptive strategy ensures computational resources are concentrated where needed, improving solution accuracy while avoiding unnecessary over/under-resolution.

These promising results are consistent with those obtained using the same adaptation procedure in other application domains, including aerospace engineering [31], hydrodynamics [26, 93] and turbulent flows [88], biomedical simulations [63, 47, 48], and image analysis [49, 94, 50]. The methodology is also being actively explored in emerging fields such as sustainability and nuclear engineering.

Acknowledgments

All the authors are members of the Gruppo Nazionale Calcolo Scientifico–Istituto Nazionale di Alta Matematica (GNCS–INdAM). ET thanks the INdAM-GNCS Project “Metodi numerici efficienti per problemi accoppiati in sistemi complessi” (CUP E53C24001950001).

References

- [1] W. Bangerth, R. Rannacher, Adaptive Finite Element Methods for Differential Equations, Lectures in Mathematics ETH Zürich, Birkhäuser Verlag, Basel, 2003.

- [2] R. Verfürth, *A Posteriori Error Estimation Techniques for Finite Element Methods*, Oxford University Press, 2013.
- [3] M. Ainsworth, J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, Pure and Applied Mathematics, John Wiley & Sons, New York, 2000.
- [4] J. T. Oden, S. Prudhomme, Goal-oriented error estimation and adaptivity for the finite element method, *Comput. Math. Appl.* 41 (5-6) (2001) 735–756.
- [5] P. Frey, P. George, *Mesh Generation: Application to Finite Elements*, ISTE, John Wiley & Sons, London, 2008.
- [6] S. Perotto, L. Formaggia (Eds.), *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, Vol. 5 of SEMA SIMAI Springer Series, Springer, Cham, 2015.
- [7] R. Sevilla, S. Perotto, K. Morgan (Eds.), *Mesh Generation and Adaptation - Cutting-Edge Techniques*, Vol. 30 of SEMA SIMAI Springer Series, Springer, Cham, 2022.
- [8] <https://www.dealii.org/>.
- [9] <https://fenicsproject.org/>.
- [10] <https://www.openfoam.com/>.
- [11] <https://www.p4est.org/>.
- [12] <https://www.ansys.com/>.
- [13] <https://www.comsol.com/>.
- [14] <https://www.3ds.com/products/simulia/abaqus/cae>.
- [15] O. C. Zienkiewicz, J. Z. Zhu, A simple error estimator and adaptive procedure for practical engineering analysis, *Int. J. Numer. Methods Eng.* 24 (2) (1987) 337–357.
- [16] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique, *Int. J. Numer. Methods Eng.* 33 (7) (1992) 1331–1364.
- [17] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. part 2: Error estimates and adaptivity, *Int. J. Numer. Methods Eng.* 33 (7) (1992) 1365–1382.
- [18] M. Xu, H. Guo, Q. Zou, Hessian recovery based finite element methods for the Cahn-Hilliard equation, *J. Comput. Phys.* 386 (2019) 524–540.
- [19] Y. Li, Recovery-based a posteriori error analysis for plate bending problems, *J. Sci. Comput.* 88 (3) (2021) Paper No. 77, 26.

- [20] P. C. Africa, C. de Falco, S. Perotto, Scalable recovery-based adaptation on Cartesian quadtree meshes for advection-diffusion-reaction problems, *Advances in Computational Science and Engineering* 1 (4) (2023) 443–473.
- [21] A. Abdulhaque, T. Kvamsdal, K. A. Johannessen, M. Kumar, A. M. Kvarving, An efficient and easy-to-implement recovery-based a posteriori error estimator for isogeometric analysis of the Stokes equation, *Comput. Methods Appl. Mech. Engrg.* 424 (2024) Paper No. 116889, 40.
- [22] Y.-D. Wang, Q. Zhan, Y.-Y. Wang, J. Zhang, N.-E. Zhang, W.-Y. Yin, A skeleton-based electromagnetic recovery for explicit adaptive mesh refinement, *IEEE Trans. Microw. Theory Tech.* 72 (9) (2024) 5155–5166.
- [23] E. Stein, S. Ohnismus, Anisotropic discretization- and model-error estimation in solid mechanics by local Neumann problems, *Comput. Methods Appl. Mech. Engrg.* 176 (1-4) (1999) 363–385.
- [24] L. Formaggia, S. Micheletti, S. Perotto, Anisotropic mesh adaption with application to CFD problems, in: *Proceedings of the Fifth World Congress on Computational Mechanics (WCCM V)*, 2002, pp. 1481–1493.
- [25] R. El Khaoulani, P. Bouchard, An anisotropic mesh adaptation strategy for damage and failure in ductile materials, *Finite Elem. Anal. Des.* 59 (2012) 1–10.
- [26] G. M. Porta, S. Perotto, F. Ballio, A space-time adaptation scheme for unsteady shallow water problems, *Math. Comput. Simul.* 82 (12) (2012) 2929–2950.
- [27] Y. Belhamadia, A. Fortin, Y. Bourgault, On the performance of anisotropic mesh adaptation for scroll wave turbulence dynamics in reaction–diffusion systems, *J. Comput. Appl. Math.* 271 (2014) 233–246.
- [28] J. Carpio, J. Prieto, M. Vera, A local anisotropic adaptive algorithm for the solution of low-mach transient combustion problems, *J. Comput. Phys.* 306 (2016) 19–42.
- [29] N. Ferro, S. Micheletti, S. Perotto, Anisotropic mesh adaptation for crack propagation induced by a thermal shock in 2D, *Comput. Methods Appl. Mech. Engrg.* 331 (2018) 138–158.
- [30] B. Lèye, J. Koko, S. Kane, M. Sy, Numerical simulation of saltwater intrusion in coastal aquifers with anisotropic mesh adaptation, *Math. Comput. Simul.* 154 (2018) 1–18.
- [31] S. Micheletti, S. Perotto, L. Soli, Topology optimization driven by anisotropic mesh adaptation: Towards a free-form design, *Comput. Struct.* 214 (2019) 60–72.
- [32] F. Gatti, C. de Falco, S. Perotto, L. Formaggia, M. Pastor, A scalable well-balanced numerical scheme for the modeling of two-phase shallow granular landslide consolidation, *J. Comput. Phys.* 501 (2024) Paper No.112798, 22.

- [33] F. Vaccaro, A. G. Mauri, S. Perotto, S. Brivio, S. Spiga, Modeling and simulation of electrochemical and surface diffusion effects in filamentary cation-based resistive memory devices, *Appl. Math. Model.* 134 (2024) 591–610.
- [34] P. Farrell, S. Micheletti, S. Perotto, An anisotropic Zienkiewicz–Zhu-type error estimator for 3D applications, *Int. J. Numer. Methods Eng.* 85 (6) (2011) 671–692.
- [35] L. Formaggia, S. Perotto, New anisotropic a priori error estimates, *Numer. Math.* 89 (4) (2001) 641–667.
- [36] L. Formaggia, S. Perotto, Anisotropic error estimates for elliptic problems, *Numer. Math.* 94 (1) (2003) 67–92.
- [37] A. Ern, J. L. Guermond, *Theory and Practice of Finite Elements*, Vol. 159, Springer, New York, 2004.
- [38] D. Mavriplis, Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes, in: *16th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2003, p. 3986.
- [39] T. Barth, D. Jespersen, The design and application of upwind schemes on unstructured meshes, in: *27th Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 1989, p. 366.
- [40] B. Van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method, *J. Comput. Phys.* 135 (2) (1997) 227–248.
- [41] P. Houston, B. Senior, E. Süli, *hp*-discontinuous Galerkin finite element methods for hyperbolic problems: error analysis and adaptivity, *Internat. J. Numer. Methods Fluids* 40 (1-2) (2002) 153–169.
- [42] P. Lancaster, K. Šalkauskas, *Curve and Surface Fitting - An Introduction*, Academic Press, Inc., London, 1986.
- [43] I. Babuška, A. Miller, The post-processing approach in the finite element method - part 2: The calculation of stress intensity factors, *Int. J. Numer. Methods Eng.* 20 (6) (1984) 1111–1129.
- [44] V. Akcelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, B. Waanders, Parallel algorithms for PDE-constrained optimization, in: M. A. Heroux, P. Raghavan, H. D. Simon (Eds.), *Parallel Processing for Scientific Computing*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006, pp. 291–322.
- [45] G. Fasshauer, J. Zhang, On choosing “optimal” shape parameters for RBF approximation, *Numer. Algorithms* 45 (1-4) (2007) 345–368.

- [46] G. M. Porta, S. Perotto, F. Ballio, Anisotropic mesh adaptation driven by a recovery-based error estimator for shallow water flow modeling, *Internat. J. Numer. Methods Fluids* 70 (3) (2012) 269–299.
- [47] N. Ferro, F. Mezzadri, D. Carbonaro, E. Galligani, D. Gallo, U. Morbiducci, C. Chiastra, S. Perotto, Designing novel vascular stents with enhanced mechanical behavior through topology optimization of existing devices, *Finite Elem. Anal. Des.* 244 (2025) Paper No. 104304, 18.
- [48] N. Ferro, S. Perotto, D. Bianchi, R. Ferrante, M. Mannisi, Design of cellular materials for multiscale topology optimization: application to patient-specific orthopedic devices, *Struct. Multidiscip. Optim.* 65 (3) (2022) 79.
- [49] M. Giacomini, S. Perotto, Anisotropic mesh adaptation for region-based segmentation accounting for image spatial information, *Comput. Math. Appl.* 121 (2022) 1–17.
- [50] F. Clerici, N. Ferro, S. Marconi, S. Micheletti, E. Negrello, S. Perotto, Anisotropic adapted meshes for image segmentation: application to three-dimensional medical data, *SIAM J. Imaging Sci.* 13 (4) (2020) 2189–2212.
- [51] S. Micheletti, S. Perotto, Anisotropic adaptation via a Zienkiewicz–Zhu error estimator for 2D elliptic problems, in: *Numerical Mathematics and Advanced Applications*, Springer Berlin, Heidelberg, 2010, pp. 645–653.
- [52] P. George, H. Borouchaki, P. Frey, P. Laug, E. Saltel, Mesh generation and mesh adaptivity, *Encyclopedia of Computational Mechanics* (2004).
- [53] M. Bendsøe, Optimal shape design as a material distribution problem, *Struct. Optim.* 1 (4) (1989) 193–202.
- [54] M. Zhou, G. Rozvany, The coc algorithm, part i: A new approach to optimal structural design, *Comput. Struct.* 38 (3) (1991) 555–561.
- [55] M. Mlejnek, Optimization of structural layouts by the homogenization method, *Comput. Mech.* 10 (2) (1992) 96–106.
- [56] G. Allaire, F. Jouve, A. Toader, A level set method for structural optimization, *Comput. Mech.* 31 (3) (2002) 189–204.
- [57] M. Wang, X. Wang, D. Guo, Level set method for structural topology optimization, *Comput. Methods Appl. Mech. Eng.* 192 (1-2) (2003) 227–246.
- [58] J. Sokółowski, A. Zochowski, Topological derivative in shape optimization, *SIAM J. Control Optim.* 37 (4) (1999) 1251–1272.
- [59] B. Bourdin, A. Chambolle, Design of structures with phase-field models, *J. Comput. Phys.* 173 (2) (2003) 720–741.

- [60] Y. Xie, G. Steven, A simple evolutionary procedure for structural optimization, *Comput. Struct.* 49 (5) (1993) 885–896.
- [61] O. Sigmund, *Topology Optimization: Theory, Methods, and Applications*, Springer Berlin, Heidelberg, 2013.
- [62] O. Sigmund, J. Petersson, Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima, *Struct. Multidisc. Optim.* 16 (1) (1998) 68–75.
- [63] D. Carbonaro, F. Mezzadri, N. Ferro, G. De Nisco, A. L. Audenino, D. Gallo, C. Chiastra, U. Morbiducci, S. Perotto, Design of innovative self-expandable femoral stents using inverse homogenization topology optimization, *Comput. Methods Appl. Mech. Eng.* 416 (2023) 116288.
- [64] D. Cortellessa, N. Ferro, S. Perotto, S. Micheletti, Enhancing level set-based topology optimization with anisotropic graded meshes, *Appl. Math. Comput.* 447 (2023) 127903.
- [65] N. Ferro, S. Micheletti, S. Perotto, An optimization algorithm for automatic structural design, *Comput. Methods Appl. Mech. Eng.* 372 (2020) 113335.
- [66] M. Artina, M. Fornasier, S. Micheletti, S. Perotto, Anisotropic mesh adaptation for crack detection in brittle materials, *SIAM J. Sci. Comput.* 37 (4) (2015) B633–B659.
- [67] O. Sigmund, N. Aage, E. Andreassen, On the (non-)optimality of Michell structures, *Struct. Multidiscip. Optim.* 54 (2016) 361–373.
- [68] E. Andreassen, C. Andreasen, How to determine composite material properties using numerical homogenization, *Comput. Mater. Sci.* 83 (2014) 488–495.
- [69] G. Allaire, Homogenization and two-scale convergence, *SIAM J. Math. Anal.* 23 (6) (1992) 1482–1518.
- [70] O. Sigmund, Materials with prescribed constitutive parameters: an inverse homogenization problem, *Int. J. Solids Struct.* 31 (17) (1994) 2313–2329.
- [71] M. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, *Comput. Methods Appl. Mech. Eng.* 71 (2) (1988) 197–224.
- [72] N. Ferro, S. Micheletti, S. Perotto, Density-based inverse homogenization with anisotropically adapted elements, in: *Numerical Methods for Flows*, Vol. 132 of *Lect. Notes Comput. Sci. Eng.*, Springer, Cham, 2020, pp. 211–221.
- [73] G. Speroni, N. Ferro, A novel metric - based mesh adaptation algorithm for 3D periodic domains, *MOX Report* 61 (2024).

- [74] M. Gavazzoni, N. Ferro, S. Perotto, S. Foletti, Multi-physics inverse homogenization for the design of innovative cellular materials: Application to thermo-elastic problems, *Math. Comput. Appl.* 27 (1) (2022).
- [75] O. Sigmund, On the optimality of bone microstructure, in: *Proceedings of the IUTAM Symposium on Synthesis in Bio Solid Mechanics*, Springer Netherlands, Dordrecht, 1999, pp. 221–234.
- [76] S. Micheletti, S. Perotto, Space-time adaptation for purely diffusive problems in an anisotropic framework, *Int. J. Numer. Anal. Model.* 7 (1) (2010) 125–155.
- [77] B. Sauvage, F. Alauzet, A. Dervieux, A space and time fixed point mesh adaptation method, *J. Comput. Phys.* 519 (2024) Paper No. 113389, 29.
- [78] F. Gatti, M. Fois, C. de Falco, S. Perotto, L. Formaggia, Parallel simulations for fast-moving landslides: space-time mesh adaptation and sharp tracking of the wetting front, *Internat. J. Numer. Methods Fluids* 95 (8) (2023) 1286–1309.
- [79] J. M. Cascón, L. Ferragut, M. I. Asensio, Space-time adaptive algorithm for the mixed parabolic problem, *Numer. Math.* 103 (2006) 367–392.
- [80] T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza, Model order reduction in fluid dynamics: challenges and perspectives, in: *Reduced Order Methods for Modeling and Computational Reduction*, Vol. 9 of MS&A, Springer, Cham, 2014, pp. 235–273.
- [81] S. Brunton, B. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* 52 (1) (2020) 477–508.
- [82] T. Coupez, G. Jannoun, N. Nassif, H. Nguyen, H. Dignonnet, E. Hachem, Adaptive time-step with anisotropic meshing for incompressible flows, *J. Comput. Phys.* 241 (2013) 195–211.
- [83] F. Alauzet, A. Loseille, G. Olivier, Time-accurate multi-scale anisotropic mesh adaptation for unsteady flows in CFD, *J. Comput. Phys.* 373 (2018) 28–63.
- [84] R. Temam, *Navier–Stokes Equations: Theory and Numerical Analysis*, Vol. 343, American Mathematical Society, 2024.
- [85] G. K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, 2000.
- [86] D. Boffi, F. Brezzi, M. Fortin, *Mixed Finite Element Methods and Applications*, Vol. 44, Springer Berlin, Heidelberg, 2013.
- [87] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Vol. 132, Birkhäuser Verlag, Basel, 1992.

- [88] E. Temellini, N. Ferro, G. Stabile, E. Delgado Ávila, T. Chacón Rebollo, S. Perotto, Space-time mesh adaptation for the VMS-Smagorinsky modeling of high Reynolds number flows, *J. Comput. Phys.* (2025) doi:10.1016/j.jcp.2025.114123.
- [89] D. Guyot, B. Bobusch, C. Paschereit, S. Raghu, Active combustion control using a fluidic oscillator for asymmetric fuel flow modulation, in: 44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 2008, p. 4956.
- [90] R. Wozidlo, F. Ostermann, H. Schmidt, Fundamental properties of fluidic oscillators for flow control applications, *AIAA J.* 57 (3) (2019) 978–992.
- [91] J. Gregory, M. Tomac, A review of fluidic oscillator development and application for flow control, in: 43rd AIAA Fluid Dyn. Conf., 2013, p. 2474.
- [92] J. Seo, C. Zhu, R. Mittal, Flow physics and frequency scaling of sweeping jet fluidic oscillators, *AIAA J.* 56 (6) (2018) 2208–2219.
- [93] B. Esfandiar, G. Porta, S. Perotto, A. Guadagnini, Impact of space-time mesh adaptation on solute transport modeling in porous media, *Water Resour. Res.* 51 (2) (2015) 1315–1332.
- [94] A. S. Chiappa, S. Micheletti, R. Peli, S. Perotto, Mesh adaptation-aided image segmentation, *Commun. Nonlinear Sci. Numer. Simul.* 74 (2019) 147–166.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 34/2025** Bucelli, M.; Dede', L.
Coupling models of resistive valves to muscle mechanics in cardiac fluid-structure interaction simulations
- 33/2025** Di Battista, I.; De Sanctis, M.F.; Arnone, E.; Castiglione, C.; Palummo, A.; Sangalli, L.M.
A semiparametric space-time quantile regression model
- 32/2025** De Sanctis, M.F.; Di Battista, I.; Arnone, E.; Castiglione, C.; Palummo, A.; Bernardi, M.; Ieva, F.; Sangalli, L.M.
Exploring nitrogen dioxide spatial concentration via physics-informed multiple quantile regression
- 31/2025** Botteghi, N.; Fresca, S.; Guo, M.; Manzoni, A.
HyperRL: Parameter-Informed Reinforcement Learning for Parametric PDEs
- 30/2025** Rosafalco, L.; Conti, P.; Manzoni, A.; Mariani, S.; Frangi, A.
Online learning in bifurcating dynamic systems via SINDy and Kalman filtering
- 29/2025** Centofanti, E.; Ziarelli, G.; Parolini, N.; Scacchi, S.; Verani, M.; Pavarino, L. F.
Learning cardiac activation and repolarization times with operator learning
- 28/2025** Ciaramella, G.; Gander, M.J.; Mazzieri, I.
Discontinuous Galerkin time integration for second-order differential problems: formulations, analysis, and analogies
- 27/2025** Antonietti P.F.; Artoni, A.; Ciaramella, G.; Mazzieri, I.
A review of discontinuous Galerkin time-stepping methods for wave propagation problems
- 24/2025** Bartsch, J.; Borzi, A.; Ciaramella, G.; Reichle, J.
Adjoint-based optimal control of jump-diffusion processes
- 22/2025** Leimer Saglio, C. B.; Pagani, S.; Antonietti P. F.
A p -adaptive polytopal discontinuous Galerkin method for high-order approximation of brain electrophysiology