



MOX-Report No. 61/2024

A novel metric - based mesh adaptation algorithm for 3D periodic domains

Speroni, G.; Ferro, N.

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

mox-dmat@polimi.it

<https://mox.polimi.it>

A novel metric-based mesh adaptation algorithm for 3D periodic domains

Giacomo Speroni¹, Nicola Ferro¹

September 5, 2024

¹MOX - Dipartimento di Matematica
Politecnico di Milano
Piazza L. da Vinci, 32, I-20133 Milano, Italy
{giacomo.speroni, nicola.ferro}@polimi.it

Abstract

We present a novel metric-based mesh adaptation algorithm, named **3DPAMA**, to be employed for discretization of three-dimensional periodic domains. The proposed method – based on mathematically rigorous assumptions – utilizes established techniques for unconstrained mesh adaptation and resorts to localized manipulations on the external boundary of the mesh. In particular, the scheme comprises four steps: (i) a non-periodic initial mesh adaptation, (ii) the splitting of the obtained volumetric grid into interior and exterior tessellations, (iii) minimal local operations to yield a periodic external surface, and (iv) the assembly of the final adapted grids. To demonstrate the robustness, efficacy, and flexibility of the proposed methodology, **3DPAMA** algorithm is employed in a continuous finite element setting to tackle test cases established in the literature as well as challenging scenarios that involve various periodic requirements, domain geometries, and metric fields. Finally, **3DPAMA** is employed in a practical use case where mesh adaptation is tightly coupled with the solution of a time-dependent partial differential equation.

1 Introduction

Many physical phenomena, such as fluid dynamics, combustion, heat transfer, structural mechanics, can be modeled through partial differential equations (PDEs) whose approximated solution can be computed resorting to numerical schemes. To discretize PDEs (and systems of PDEs), it is possible to resort to several approaches, e.g., finite element [1, 2], boundary element [3], and virtual element [4] methods, as well as finite volumes [5]. Such techniques strongly relies on the discretization of the spatial physical domain via a partition, called mesh, whose elements allows for the definition of a discrete space and for the computation of the approximated fields of interest. In this context, when numerical solutions exhibit intricate features (e.g., multiscale patterns, boundary and/or internal layers, shocks), the mesh should be appropriately refined with the goal

to output accurate solutions. Indeed, performing a uniform refinement of the tessellation of the domain yields approximations progressively more precise, while leading to a significant increase in the number of degrees of freedom associated with the numerical scheme and, eventually, to a high computational burden. Possible remedies, which aim to balance accuracy requirements and computational costs, are h -adaptivity [6], frequently simply referred to mesh adaptivity, that performs an informed re-partition of the domain, r -adaptivity [7], that optimally dislocates the element vertices, p -adaptivity [8], that selects the local polynomial degree of the discrete space, and hp -adaptivity [9] that combines h - and p -adaptivity.

Among the possible choices, mesh adaptivity allows to design computational meshes that are automatically tailored to the physical problem at hand, thus allowing for an efficient numerical approximation of complex phenomena [10]. With this aim, mesh adaptation techniques can be informed through different indicators and estimators that identify the portions of the domain where mesh refinement and coarsening operations, such as edge swaps, edge collapses, edge splits, point insertion, face swaps [11], are expected to be undertaken. For instance, a posteriori error estimators represent a rigorous way to detect such regions and to lead the mesh adaptation process, as extensively investigated in [12, 13, 14, 15]. In response to the great success of mesh adaptation, several techniques have been efficiently implemented in different software suites and are successfully adopted for several engineering applications, e.g., fluid flow mechanics [16], crack propagation [17, 18], topology optimization [19, 20], and multi-scale modeling [21].

In fields like flows in turbomachines [22], crystal plasticity [23], homogenization in micro-mechanical models [24], design of microstructure [25, 26], the physical problems under investigation may feature a periodicity constraint on the boundary of the domain to be enforced through periodic boundary conditions. This circumstance naturally prompts the need for the generation and adaptation of periodic meshes, namely tessellations that include pairs of matching discretized boundaries, where periodic boundary conditions can be imposed in a straightforward way. However, the implementation of 3-dimensional mesh adaptation techniques for periodic domain is still an open issue. The difficulty – mostly negligible in 2D cases – arises in the fact that the refinement and coarsening operations needed for mesh adaptation must be extended to a periodic setting, thus requiring scrupulous care and making the well-established mesh adaptation frameworks and software not fully appropriate. For instance, if a mesh edge lying on the periodic boundary undergoes splitting, swapping, or collapsing, all the corresponding periodic counterparts must be modified accordingly [27]. Furthermore, there is a greater implementation challenge in 3D scenarios, since the splitting of faces is not unique for some configurations. In order to overcome these challenges, it is often customary to i) sacrifice the use of mesh adaptation for periodic grids and high accuracy requirements, by resorting to uninformed structured tessellation that can account for periodic constraints by design, e.g., in [26]; ii) perform mesh adaptation by freezing the discretized domain boundary, leading to a significant computational burden and to the degradation of the performance of the mesh generator and of the solution accuracy; iii) ditch the assignment of exact periodicity and rely on a weak imposition of the periodic conditions,

using Lagrange multipliers or mortar techniques for gluing periodic surfaces with non-matching meshes (see, e.g., [28]). However, techniques i)-iii) only circumvent the real adaptation issues and do not often represent the ideal discretization tools.

Up to the authors' knowledge, the literature on mesh adaptation for 3-dimensional periodic domain is very limited. For instance, in [29], the grid is made conforming with respect to a separation manifold through local operations and r -adaptivity. Regarding h -adaptation, in [27], a 2- and 3-dimensional periodic mesh adaptation technique is implemented by resorting to a geometric migration of the grid entities and to the introduction of ghosts elements. This procedure is successful, but requires special care if two or more periodic conditions are to be enforced. Also, it leads to the modification of the shape of the computational domain, affecting its popularity in the engineering community and complicating interpolation operations. The latter issue is tackled in [22], where the migration task is followed by an anti-migration to restore the original domain in RANS turbomachinery applications relying on a Hessian-based anisotropic mesh adaptation.

In this paper, we introduce a mathematically rigorous and flexible algorithm that exploits the metric framework, allowing for the periodic tetrahedral tessellation of any generic 3D domain. The proposed strategy unlocks the possibility to use the well-known techniques for unconstrained mesh adaptation with the additional handling of the boundary discretization that accounts for the periodicity requests. Differently from [27, 22], we propose to enforce the periodicity via an internal splitting of the mesh thus avoiding any intermediate additional steps that involve changes or movement in the reference domain. The algorithm here presented conveniently performs only one global mesh adaptation and treats the boundary elements in an automatic way. The method presents some additional desirable features that accommodate the user's needs: it handles generic domain shapes and any pair of boundary conditions, it can be fed with different source of information (e.g., analytic metrics or feature-based indicators or estimators) to lead the mesh adaptation, and, finally, it exploits well-established software for an easy integration in the code flow.

The paper is organized as follows. In Section 2, we present the theoretical framework of interest and we present a step-by-step procedure to generate a periodic mesh. Section 3 is devoted to an extensive test case campaign based on the finite element method in order to assess the feasibility and effectiveness of the algorithm in different scenarios that can be relevant for various use cases. Finally, in Section 4, we gather the conclusions and we outline some future perspectives that can broaden the range of applicability of the current work.

2 The methodology

In this work, we focus on the adaptation of tetrahedral grids with a specific focus on their employment for the finite element method. Throughout the whole paper, we employ the notation:

- $\mathcal{T}_h = \{K\}$ to refer to a volumetric mesh composed of tetrahedra K ;

- $S_h = \{T\}$ to indicate a surface discretized via triangles T ;
- $L_h = \{l\}$ for a 1-dimensional partition constituted by the contiguous segments l .

Moreover, we consider a metric-based framework [14]. Such framework relies on the definition of the metric, $\mathcal{M} : \Omega \rightarrow \mathbb{R}^{3 \times 3}$, namely a symmetric positive definite tensor field defined over the domain $\Omega \subset \mathbb{R}^3$. The metric tensor can be regarded as a size map that induces the mesh $\mathcal{T}_h = \{K\}$ of Ω , as it prescribes the geometry (namely, size, shape and orientation) of each generic tetrahedron $K \in \mathcal{T}_h$ [30]. This fact becomes evident when metric \mathcal{M} is spectrally decomposed as

$$\mathcal{M} = R^T \Lambda R, \quad \text{with} \quad R = (\mathbf{r}_1 | \mathbf{r}_2 | \mathbf{r}_3)^T, \quad \Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix},$$

where \mathbf{r}_i and λ_i , $i = 1, 2, 3$, are the eigenvectors and eigenvalues of \mathcal{M} . In this decomposition: vectors $\mathbf{r}_i|_K$ correspond to the element K orientation; values $\lambda_i|_K \simeq 1/h_i^2$ are associated with the lengths, h_i , of the edges of K [14]; the element shape is determined by the so-called maximum aspect ratio, defined as [31, 32]

$$s|_K = \max_{i=1,\dots,3} \left(\frac{\lambda_i|_K^2}{\prod_{j \neq i} \lambda_j|_K} \right)^{2/3}, \quad (1)$$

that attains value equal to 1 for isotropic elements, whereas highlights marked anisotropy for $s|_K \gg 1$.

The concept of metric (and of unit mesh [14]) is widely employed to implement common mesh generators. In particular, in order to build a tessellation \mathcal{T}_h that is induced by a prescribed metric \mathcal{M} , mesh generators are typically fed with a support mesh \mathcal{T}_h^0 of Ω and with a polynomial piecewise continuous approximation of \mathcal{M} , such that $\mathcal{M} \approx \mathcal{M}_h^1 \in [V_h^1(\mathcal{T}_h^0)]^{3 \times 3}$, with

$$V_h^p(\mathcal{T}_h^0) = \{u_h \in C^0(\bar{\Omega}) : u_h|_K \in \mathbb{P}_p(K), \forall K \in \mathcal{T}_h^0\}, \quad p \geq 1,$$

where $\mathbb{P}_p(K)$ is the space of polynomials of degree p defined over the tetrahedron K . Within this formalism, in a metric-based framework, a grid adaptation strategy entails inputting a solution-informed metric $\mathcal{M}_h^{1,\text{adapt}} \in [V_h^1(\mathcal{T}_h^0)]^{3 \times 3}$ that induces an adapted mesh $\mathcal{T}_h^{\text{adapt}}$, whose elements are oriented, shaped and sized according to the eigenvectors $\mathbf{r}_{h,i}^{\text{adapt}}$ and eigenvalues $\lambda_{h,i}^{\text{adapt}}$ of $\mathcal{M}_h^{1,\text{adapt}}$.

The methods that lead to the computation of $\mathbf{r}_{h,i}^{\text{adapt}}$ and $\lambda_{h,i}^{\text{adapt}}$ and, in turn, to the definition of $\mathcal{M}_h^{1,\text{adapt}}$ are diverse and range from heuristics to mathematically rigorous error estimators on the discrete solution. As this topic is out of the scope of this manuscript, we mention the works in [33, 14, 34, 35] for more details, while we refer to Section 3 for some metric computation cases.

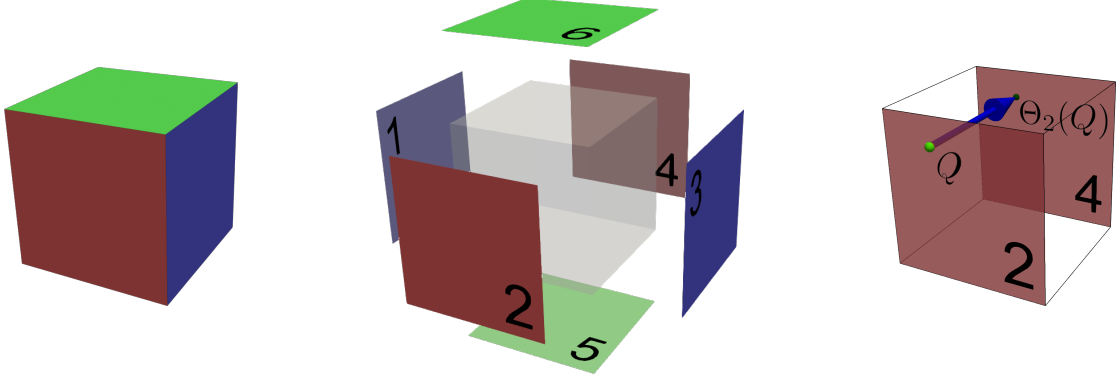


Figure 1: Reference domain Ω (left), surfaces S_i , $i = 1, \dots, 6$ (center), and example of the transformation Θ_2 applied to a point $Q \in S_2$.

2.1 A metric-based mesh adaptation for periodic domains

Without loss of generality, let us consider the unitary cube $\Omega = (0, 1)^3$, whose boundary $\partial\Omega$ is composed of the six non-overlapping faces denoted by $S_i \subset \partial\Omega$, where index $i \in L = \{1, \dots, 6\}$ is used to label the face, as depicted in Fig. 1, left and center panels. We assume that $\partial\Omega$ can be decomposed as $\Gamma_P \cup \Gamma$, where Γ_P represents the boundary portion where periodic conditions are to be enforced, according to specific problem at hand. Specifically, we define $\Gamma_P = \cup_{i \in I} S_i$ for a given set of indices $I \subset L$, whereas Γ comprises the remaining surfaces S_i , for $i \in L \setminus I$.

In order to link two faces, S_i and S_j , $i, j \in I$, that require periodicity, we define the bijective correspondence $i \leftrightarrow j$ as an idempotent map, $\sigma : \mathbb{N} \rightarrow \mathbb{N}$, that associates the indices of two corresponding periodic faces, i.e., $i \mapsto \sigma(i) = j \mapsto \sigma(\sigma(i)) = i$, for $i, j \in I$. Thus, we can extract a reduced subset of indices $\mathring{I} \subset I$, so that the paired faces are identified by the indices $(i, \sigma(i))$, with $i \in \mathring{I}$, and it holds that

$$\Gamma_P = \bigcup_{i \in \mathring{I}} (S_i \cup S_{\sigma(i)}). \quad (2)$$

Once the indices of the periodic faces are paired through σ , we introduce the bijective map Θ_i that associates the points in S_i to the corresponding ones in $S_{\sigma(i)}$, for $i \in \mathring{I}$, in accordance with the periodicity requirement. For instance, in the case of Fig. 1, right panel, we consider the pair of periodic faces $(2, \sigma(2) = 4)$ and we sketch how map Θ_2 transforms a generic point $Q \in S_2$ into the point $\Theta_2(Q) \in S_4$ through a rigid translation along the direction normal to S_2 . Indeed, we remark that the pairing of surfaces with indices $(i, \sigma(i))$ leads to the identification of the inverse map of Θ_i with $\Theta_{\sigma(i)}$.

Maps σ and Θ_i permit to express a generic periodic boundary condition in a straightforward way as

$$v|_{S_i} = v|_{S_{\sigma(i)}} \iff v(\mathbf{x}) = v(\Theta_i(\mathbf{x})), \quad \forall \mathbf{x} \in S_i, \quad i \in \mathring{I},$$

where v is a generic periodic field defined on Γ_P and $\mathbf{x} \in \mathbb{R}^3$ is a generic point with Cartesian coordinates (x, y, z) .

The same considerations hold for a discrete setting. After discretizing the domain Ω with mesh \mathcal{T}_h^0 and introducing the discrete spaces $V_h^P(\mathcal{T}_h^0)$, periodic boundary conditions can be enforced on the function $v_h \in V_h^P(\mathcal{T}_h^0)$ by assigning that

$$v_h|_{S_{h,i}} = v_h|_{S_{h,\sigma(i)}} \iff v_h(\mathbf{x}) = v_h(\Theta_i(\mathbf{x})), \quad \forall \mathbf{x} \in S_{h,i}, \quad i \in \mathring{I}, \quad (3)$$

with $S_{h,i} = \{T\}$ the discretization of face S_i through triangles T .

Constraints (3) are satisfied if:

- the discretized paired faces $S_{h,i}$ and $S_{h,\sigma(i)}$ have vertices and triangles that match according to map Θ_i (and $\Theta_{\sigma(i)}$);
- the finite element function v_h has equal values attached to the corresponding degrees of freedom (DOFs) that lie on $S_{h,i}$ and $S_{h,\sigma(i)}$.

The second requirement is straightforward, since it only implies that the DOFs vector associated with a discrete function defined on face $S_{h,i}$ is duplicated for the corresponding vector associated with $S_{h,\sigma(i)}$. Vice versa, the first task needs special attention as the discretization of the paired faces cannot be performed independently.

Indeed, the construction of a periodic grid matching the first requirement is far from trivial and adapting a mesh under these demands is even more difficult. For instance, in the metric framework introduced at the beginning of the section, feeding a mesh generator with a periodic metric $\mathcal{M}_h^{1,\text{adapt}}$ is not sufficient to guarantee that the resulting mesh is periodic. In fact, mesh generators enforce the metric requests only in a loose sense. For this reason, we propose a 4-step procedure to generate 3D meshes adapted according to an input metric, $\mathcal{M}_h^{\text{in}}$, defined on the support mesh \mathcal{T}_h^0 , that are perfectly periodic and where constraints (3) can be automatically enforced.

For clarity reasons, in this section, we keep the unitary cube as the domain Ω and we consider the case where $\Gamma_P = \partial\Omega$, with (1, 3), (2, 4), and (5, 6) pairs of periodic faces, numbered according to Fig. 1, center panel. Additional comments regarding different types and numbers of periodic conditions are in Section 3.2.

Phase 1. Non-periodic 3D mesh adaptation The first step of the procedure consists of adapting the initial support mesh \mathcal{T}_h^0 via a standard volumetric mesh generator, fed with the input metric $\mathcal{M}_h^{\text{in}} \in [V_h^1(\mathcal{T}_h^0)]^{3 \times 3}$, and without any constraints on periodicity. The resulting grid, $\mathcal{T}_h^{\text{adapt}}$, features shape, size, and orientation of the elements as prescribed by the input metric. However, $\mathcal{T}_h^{\text{adapt}}$ has to be regarded only as an educated guess for the periodic mesh, since no strong periodicity is attained nor guaranteed and further manipulation is needed. Figure 2 graphically exemplifies this phase, by showing the initial support mesh (left panel), the resulting adapted grid (center panel), and a zoom of faces $S_{h,2}$ and $S_{h,4}$ (right panel) where vertices and edges do not correspond, making $\mathcal{T}_h^{\text{adapt}}$ not suitable for the enforcement of periodic boundary conditions.

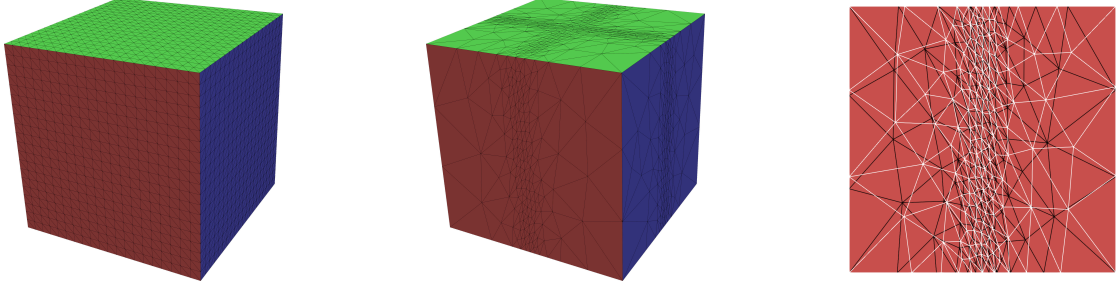


Figure 2: Phase 1: initial support mesh \mathcal{T}_h^0 (left), non-periodic adapted mesh $\mathcal{T}_h^{\text{adapt}}$ (center), mesh $S_{h,2}$ superimposed to $S_{h,4}$ (right, red-colored, with white- and black-highlighted edges, respectively).

Phase 2. Mesh splitting Mesh $\mathcal{T}_h^{\text{adapt}}$ is split into the portions $\mathcal{T}_h^{\text{int}}$ and $\mathcal{T}_h^{\text{ext}}$, so that

$$\mathcal{T}_h^{\text{adapt}} = \mathcal{T}_h^{\text{int}} \oplus^V \mathcal{T}_h^{\text{ext}}, \quad (4)$$

with $\mathcal{T}_h^{\text{ext}}$, referred to as the external mesh, collecting the tetrahedra of $\mathcal{T}_h^{\text{adapt}}$ that share a vertex, an edge, or a face with $S_{h,i}$ for $i \in I$, and such that $\mathcal{T}_h^{\text{int}}$, named the internal mesh, is characterized by a single connected component made up of the remaining elements (see Fig. 3 for a sketch of $\mathcal{T}_h^{\text{int}}$ in the left panel, a cut view of $\mathcal{T}_h^{\text{ext}}$ in the center panel, and a representation of the sum operation in (4) in the right panel). We observe that meshes $\mathcal{T}_h^{\text{ext}}$ and $\mathcal{T}_h^{\text{int}}$ define two polygonal open sets, Ω_h^{ext} and Ω_h^{int} , do not overlap by construction, and share a set of faces, edges and vertices that constitutes the separating surface mesh $\Gamma_h^{\text{int,ext}} = \overline{\Omega}_h^{\text{int}} \cap \overline{\Omega}_h^{\text{ext}}$.

We remark that the sum operation \oplus^V in (4) is to be intended as a mesh operator that merges two 3D volumetric tessellations, \mathcal{T}_h^1 and \mathcal{T}_h^2 , provided that these grids share identical triangles, edges and vertices in correspondence with a surface interface $\Gamma_h^{1,2}$. In the sequel, upon simple modifications, the sum \oplus^V will be customized to surface and linear meshes by introducing the operators \oplus^S and \oplus^L , respectively.

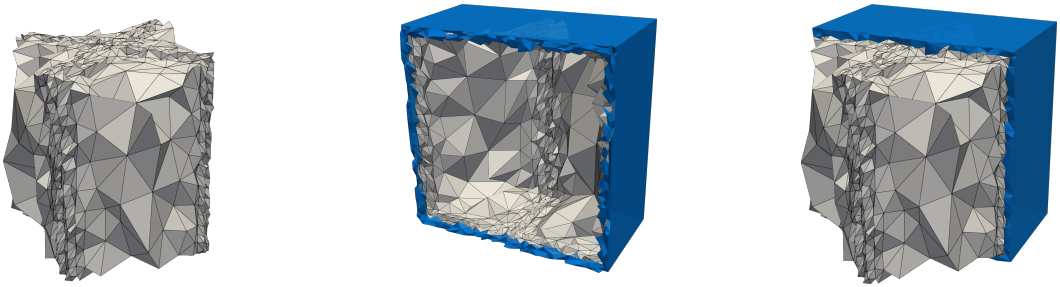


Figure 3: Phase 2: internal mesh $\mathcal{T}_h^{\text{int}}$ (left), cut view of the external mesh $\mathcal{T}_h^{\text{ext}}$ (center), and vertically-clipped view of $\mathcal{T}_h^{\text{adapt}}$ as the sum $\mathcal{T}_h^{\text{int}} \oplus^V \mathcal{T}_h^{\text{ext}}$ (right).

Phase 3. 1D-2D periodicity on the surfaces The third phase is applied to enforce periodicity on the portion $\mathcal{T}_h^{\text{ext}}$ of the splitting in (4) and, thus, represents the main core and the novelty of the proposed procedure. We observe that the boundary of the external mesh, namely $\partial\mathcal{T}_h^{\text{ext}}$, can be split into the interface surface mesh $\Gamma_h^{\text{int,ext}}$ as defined in Phase 2, and in the remaining portion, $\Gamma_h^{\text{ext}} = \partial\mathcal{T}_h^{\text{ext}} \setminus \Gamma_h^{\text{int,ext}}$, that, by design, coincides with a non-periodic discretization of Γ_P .

In order to locally modify Γ_h^{ext} and to yield a periodic mesh of Γ_P , we operate a selection of the surface and linear meshes to be adjusted to accommodate the periodic requirements. In particular, we first pick $S_{h,i}$, with $i \in \mathring{I}$, to be considered as the reference discrete faces for each pair of periodic surfaces (see Fig. 4, first panel). This extraction leads to take into consideration only face $S_{h,i}$ of the pair $(i, \sigma(i))$, while face $S_{h,\sigma(i)}$ will inherit all the modifications at the end of the process.

Similarly, we select the linear meshes $L_{h,ij} = S_{h,i} \cap S_{h,j}$, which discretize the intersection between the i -th and the j -th extracted reference surface, with $i, j \in \mathring{I}$, and for $i < j$, since $L_{h,ij} = L_{h,ji}$ (Fig. 4, second panel). These three linear meshes are independent and can be employed to build the frame of the periodic mesh. Specifically, we can assemble the linear periodic frame, $L_{h,P}$, that discretize the cube edges by

$$L_{h,P} = \bigoplus_{i < j, i, j \in \mathring{I}}^L [L_{h,ij} \cup \Theta_i(L_{h,ij}) \cup \Theta_j(L_{h,ij}) \cup \Theta_i(\Theta_j(L_{h,ij}))]. \quad (5)$$

The use of maps Θ_i and Θ_j ¹ to transfer the reference linear meshes across the periodic boundary guarantee that $L_{h,P}$ is periodic (see Fig. 4, third panel).

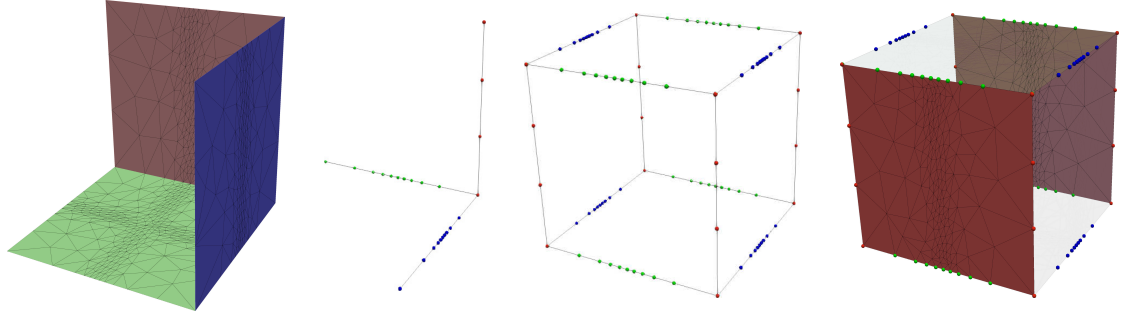


Figure 4: Phase 3: reference surface (first panel) and linear (second panel) meshes, linear periodic frame $L_{h,P}$ (third panel), and the pair $\tilde{S}_{h,2}$ and $\Theta_2(\tilde{S}_{h,2})$ of periodic surface meshes (fourth panel).

Once the linear mesh $L_{h,P}$ is constructed, we consider the extracted reference faces $S_{h,i}$, $i \in \mathring{I}$, and we generate the discretizations $\tilde{S}_{h,i}$ by adjusting only the vertices of the outer layer of triangles of $S_{h,i}$ so that they match $L_{h,P}$. This expedient allows us to guarantee periodicity at the face intersections while keeping most of the existing

¹The application of map Θ_i to a discrete partition L_h is here intended as an operator that transforms all the edges and the vertices in L_h through Θ_i .

tessellation – adapted at Phase 1 according to $\mathcal{M}_h^{\text{in}}$ – intact. Finally, the periodic external surface is assembled as

$$\Gamma_{h,P}^{\text{ext}} = \bigoplus_{i \in \tilde{I}}^S [\tilde{S}_{h,i} \cup \Theta_i(\tilde{S}_{h,i})],$$

where the use of map Θ_i ² allows transferring the extracted reference surface meshes by periodicity (see Fig. 4, fourth panel, for the visualization of $\tilde{S}_{h,2}$ and $\tilde{S}_{h,4}$).

Phase 4. 3D periodic mesh assembly This phase concludes the process by assembling all the portions of the mesh. In particular, the surface mesh $\Gamma_h^{\text{env}} = \Gamma_{h,P}^{\text{ext}} \oplus^S \Gamma_h^{\text{int,ext}}$ is a discretization of the boundary of the external domain Ω_h^{ext} (see Fig. 5, left panel). However, Γ_h^{env} does not coincide with $\partial\mathcal{T}_h^{\text{ext}}$ obtained in Phase 2 because of the modifications to $S_i, i \in I$ occurred in Phase 3. Consequently, the volumetric mesh $\mathcal{T}_h^{\text{ext}}$ needs to be updated to properly match Γ_h^{env} . With this aim, we keep the external envelope Γ_h^{env} fixed and we create a new tetrahedralization of Ω_h^{ext} , namely we generate the volumetric mesh, $\mathcal{T}_{h,P}^{\text{ext}}$ as in Fig. 5, right panel. To ensure consistency with respect to the tentative adaptation in Phase 1, this volumetric mesh generation is guided through the metric $\mathcal{M}_h^{\text{in}}|_{\Omega_h^{\text{ext}}}$.

As a result, grid $\mathcal{T}_{h,P}^{\text{ext}}$ i) exactly matches the interface with the internal mesh $\mathcal{T}_h^{\text{in}}$, and ii) is periodic by construction, being $\Gamma_h^{\text{int,ext}}$ and $\Gamma_{h,P}^{\text{ext}}$ of Phases 2 and 3 kept fixed, respectively.

Finally, we assemble the global 3D mesh of Ω (Fig. 6, left panel), by setting

$$\mathcal{T}_{h,P}^{\text{adapt}} = \mathcal{T}_h^{\text{int}} \oplus^V \mathcal{T}_{h,P}^{\text{ext}}.$$

We highlight that the adapted mesh $\mathcal{T}_{h,P}^{\text{adapt}}$, shown in Fig. 6, left panel, is periodic by construction, automatically inheriting the property from $\mathcal{T}_{h,P}^{\text{ext}}$. This fact is particularly evident in the zoom in the right panel, where paired surface meshes $\tilde{S}_{h,2}$ and $\tilde{S}_{h,4}$ (red-colored), and $\tilde{S}_{h,1}$ and $\tilde{S}_{h,3}$ (blue-colored) are overlapped with no vertex or triangle mismatch.

Remark 1 (Extensions) *The procedure here detailed can be effectively employed also when $I \neq L$, namely when periodicity is not to be enforced on all boundary faces. In this case, Phase 3 has to be modified in order to appropriately choose the reference edges, $L_{h,ij}$, and the reference surfaces, $S_{h,i}$ to be periodically transferred through Θ_i , with $i \in \tilde{I}$, being $\#\tilde{I} < 3$, where $\#$ indicates the set cardinality. This particularization is practically explored in Section 3.1.*

Moreover, this algorithm can be extended to generic domains, without orthogonal or straight edges. We refer to Section 3.2 for some examples.

²The application of map Θ_i to a discrete surface S_h is here intended as an operator that transforms all the vertices coordinates and the edges in S_h through Θ_i .

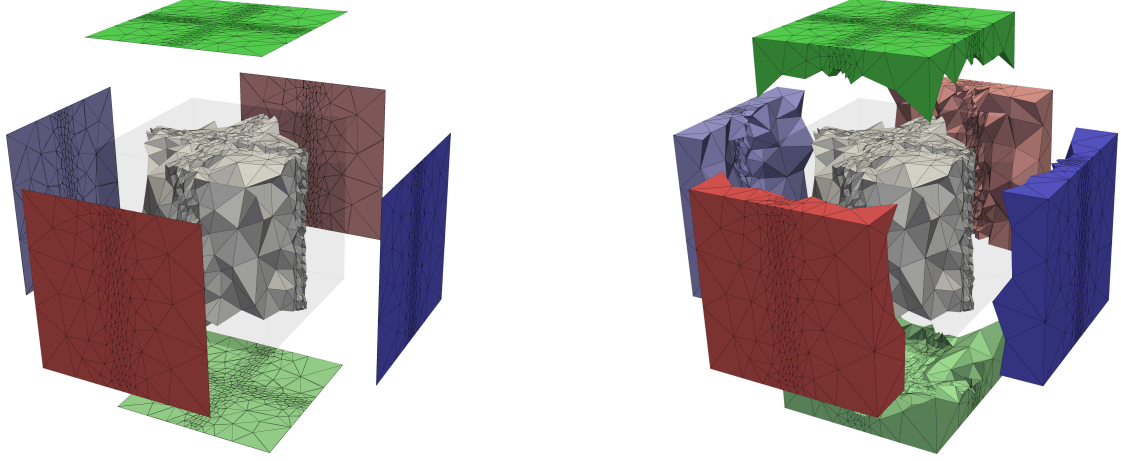


Figure 5: Phase 4: exploded views of $\Gamma_{h,P}^{\text{ext}}$ together with $\Gamma_h^{\text{int,ext}}$ (left), and of the periodic volumetric external mesh $\mathcal{T}_{h,P}^{\text{ext}}$ (right).

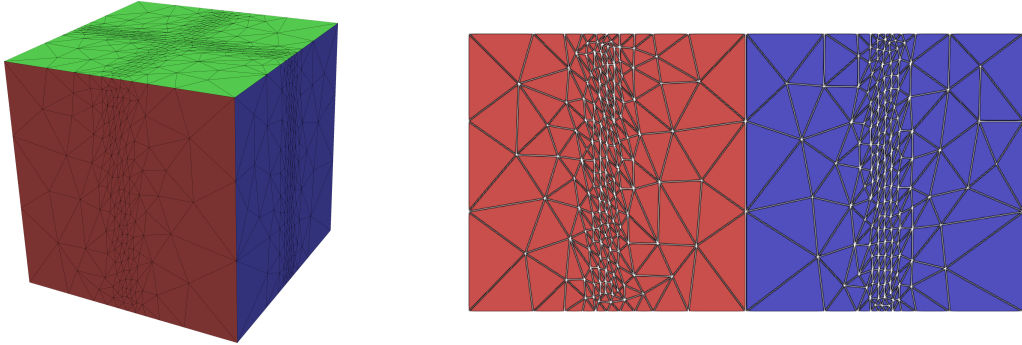


Figure 6: Phase 4: periodic adapted mesh $\mathcal{T}_{h,P}^{\text{adapt}}$ (left), mesh $\tilde{S}_{h,2}$ superimposed to $\tilde{S}_{h,4}$ (right, red-colored, with white- and black-highlighted edges, respectively), $\tilde{S}_{h,1}$ superimposed to $\tilde{S}_{h,3}$ (right, blue-colored, with white- and black-highlighted edges, respectively).

2.2 The 3DPAMA algorithm

The phases explained in Section 2.1 can be summarized in the pseudocode listed in Algorithm 1, named 3DPAMA (3D Periodic Anisotropic Mesh Adaptation).

Algorithm 1 3DPAMA: 3D Periodic Anisotropic Mesh Adaptation

Input: \mathcal{T}_h^0 , I , σ , $\{\Theta_i\}_{i \in I}$, method, f , tol, MTOL, itmax

```

1: Set: it = 0, errM = 1 + MTOL, flag = 0;
2: while (errM > MTOL & it < itmax) do
3:    $\mathcal{M}_h^{\text{in}} = \text{ComputeMetric}(f, \text{tol}, \text{method});$ 
4:    $\mathcal{T}_h^{\text{adapt}} = \text{Phase1}(\mathcal{T}_h^0, \mathcal{M}_h^{\text{in}});$ 
5:    $\text{errM} = \left| \#\mathcal{T}_h^{\text{adapt}} - \#\mathcal{T}_h^0 \right| / \#\mathcal{T}_h^0;$ 
6:    $\mathcal{T}_h^0 = \mathcal{T}_h^{\text{adapt}};$ 
7:   it = it + 1;
8: end while
9:  $\mathcal{M}_h^{\text{in}} = \text{Project}(\mathcal{M}_h^{\text{in}}, \mathcal{T}_h^{\text{adapt}});$ 
10:  $\left[ \mathcal{T}_h^{\text{int}}, \mathcal{T}_h^{\text{ext}}, \Gamma_h^{\text{int,ext}} \right] = \text{Phase2}(\mathcal{T}_h^{\text{adapt}});$ 
11:  $\left[ L_{h,P}, \Gamma_{h,P}^{\text{ext}} \right] = \text{Phase3}(\mathcal{T}_h^{\text{ext}}, I, \sigma, \{\Theta_i\}_{i \in I});$ 
12:  $\mathcal{T}_{h,P}^{\text{adapt}} = \text{Phase4}(\mathcal{T}_h^{\text{int}}, \Gamma_{h,P}^{\text{ext}}, \Gamma_h^{\text{int,ext}}, \mathcal{M}_h^{\text{in}});$ 
13: flag = VerifyP( $\mathcal{T}_{h,P}^{\text{adapt}}$ );
```

Output: $\mathcal{T}_{h,P}^{\text{adapt}}$, flag

The input parameters of the procedure are: the support mesh, \mathcal{T}_h^0 , the set of indices I , maps σ and Θ_i , with $i \in I$. Moreover, for the metric computation, 3DPAMA takes the variable method, the function f , and the tolerance tol, employed to compute the metric tensor, and the values MTOL and itmax to impose a stop criterion.

The algorithm implements Phases 1–4 in Section 2.1 in the routines Phase1–Phase4. In particular, in the spirit of the state-of-the-art mesh generator implementations [10], Phase1 is called in a **while** loop (lines 2–8), in order to reach a desired refinement and precision of the first adapted mesh $\mathcal{T}_h^{\text{adapt}}$. This cycle involves the computation of the metric through ComputeMetric (line 3). This routine implements different methods for metric computation that are recurrent in finite element analysis, depending on the

specified `method`. In particular, if `method` is set to ‘`Analytic`’, f is interpreted as an analytic tensor representing \mathcal{M}^{in} , the discrete metric is computed by interpolating f onto $[V_h^1(\mathcal{T}_h^0)]^{3 \times 3}$, and parameter `tol` is discarded. Conversely, if `method` is chosen as ‘`Hessian`’ or ‘`ZZ`’, a Hessian-derived error surrogate or an anisotropic variant of the Zienkiewicz-Zhu estimator (see [15, 31] and Section 3.3 for more details) based on function f is implemented, respectively, with tolerance `tol` tuning the refinement level. **Phase1** is then called to generate the non-periodic tentative adapted mesh $\mathcal{T}_h^{\text{adapt}}$ (line 4). The **while** loop is stopped if the relative difference between the cardinality of two successive adapted meshes is below the threshold `MTOL` or if the maximum number of iterations `itmax` is exceeded. When the convergence criterion is reached, metric $\mathcal{M}_h^{\text{in}}$ is projected onto the adapted mesh $\mathcal{T}_h^{\text{adapt}}$ (line 9) and Phases 2–4 are sequentially called, constructing the periodic adapted mesh $\mathcal{T}_{h,P}^{\text{adapt}}$ (lines 10–12).

As a further control, the algorithm automatically checks that the periodicity is correctly imposed through a verification routine that returns a boolean value, being `flag` = 1 if the procedure succeeds, 0 otherwise (line 13).

The control value `flag` and the periodic mesh $\mathcal{T}_{h,P}^{\text{adapt}}$ are finally returned as the output of **3DPAMA**.

We remark that any 3D mesh generation, adaptation and manipulation software can be used to implement routines **Phase1**, **Phase2**, **Phase3**, and **Phase4**. We choose to implement the algorithm mainly in **FreeFEM** [36, 37], leveraging its built-in functions and libraries for mesh manipulation, projection and periodicity verification. Additionally, we rely on **mmg3d** [38, 39] for the metric-based mesh adaptation invoked in Phase 1, and on **Tetgen** [40, 41] combined with the `-nosurf` option of **mmg3d** for the creation of $\mathcal{T}_{h,P}^{\text{ext}}$ in Phase 4.

3 Test cases

We assess **3DPAMA** by proposing different test cases with the goal of stressing the efficacy and the flexibility of the algorithm. For this purpose, we consider a continuous finite element setting and we focus on adaptation scenarios where the input metric is defined according to different methods and where the considered geometries and periodicity conditions may vary.

3.1 An analytic metric

As a first test case, we propose a literature comparison by adapting the analysis done in [27]. Let us consider the domain $\Omega = (0, 1)^3$ coupled with a single set of periodic boundary conditions, namely, with reference to Fig. 1, center panel, $I = \{2, 4\} \neq L$, with $\sigma(2) = 4$ and with surfaces $S_2 = \{(x, y, z) \in \partial\Omega, x = 1\}$ and $S_4 = \{(x, y, z) \in \partial\Omega, x = 0\}$ paired through the maps $\Theta_2(x, y, z) = [1 - x, y, z]^T$ and $\Theta_4(x, y, z) = [x + 1, y, z]^T$. These transformations impose a translational periodicity, acting as a rigid movement of points in S_2 towards the corresponding counterpart in S_4 along the normal direction.

Following the approach in [27], we initiate **3DPAMA** with a tessellation \mathcal{T}_h^0 of Ω consisting of 4 tetrahedra, and we set `method` = ‘**Analytic**’, the metric being

$$f = \mathcal{M}^{\text{in}} = \begin{pmatrix} 1/h^2 & 0 & 0 \\ 0 & 1/h^2 & 0 \\ 0 & 0 & 1/h^2 \end{pmatrix}, \quad (6)$$

with $h = 0.1$. Tensor \mathcal{M}^{in} is constant and diagonal, meaning that the induced periodic adapted mesh is expected to be isotropic with edge size close to h .

The adapted grid in Fig. 7 is delivered by 10 consecutive calls to Algorithm 1. The implementation of such an iterative procedure is necessary to guarantee high quality results when starting from an extremely coarse initial mesh \mathcal{T}_h^0 . Output tessellation $\mathcal{T}_{h,P}^{\text{adapt}}$ turns out to be unstructured and perfectly periodic, featuring 13463 isotropic tetrahedra (the maximum aspect ratio $s|_K$ in (1) being ~ 1 for all elements), with the majority of the edges matching the target size h .

For a more thorough analysis, we quantify the performance of **3DPAMA** by computing the quality Q of the adapted tetrahedra as in [38], being

$$Q(K) = \alpha_d \frac{|K| \sqrt{\det \mathcal{M}_K^{\text{in}}}}{\left(\sum_{i=1}^6 \mathbf{e}_{i,K}^T \mathcal{M}_K^{\text{in}} \mathbf{e}_{i,K} \right)^{3/2}} \quad \forall K \in \mathcal{T}_{h,P}^{\text{adapt}},$$

where $\mathbf{e}_{i,K}$, $i = 1, \dots, 6$, are the edges of K , $\alpha_d = 72\sqrt{3}$ is a normalisation constant, $\mathcal{M}_K^{\text{in}}$ is the average of $\mathcal{M}_h^{\text{in}}$ over tetrahedron K , and $|K|$ is the element volume. Indicator Q is commonly employed for assessing whether the considered grid is suitable for reliable finite element approximations and measures how closely the mesh elements agree with the prescriptions of the input metric. A good mesh quality is ensured for $Q > 0.12$, for almost every element, $Q = 1$ holding for a regular tetrahedron.

Additionally, we compute the edge length, $\ell^{\mathcal{M}^{\text{in}}}$, associated with the generic oriented edge $\mathbf{e}_{i,K}$ of K connecting points A and B , that is adimensionalized with respect to the metric \mathcal{M}^{in} , as

$$\ell^{\mathcal{M}^{\text{in}}}(\mathbf{e}_{i,K}) = \int_0^1 \sqrt{\mathbf{e}_{i,K}^T \mathcal{M}^{\text{in}}(A + t\mathbf{e}_{i,K}) \mathbf{e}_{i,K}} dt, \quad K \in \mathcal{T}_{h,P}^{\text{adapt}}, i = 1, \dots, 6.$$

A good enforcement of the edge size induced by the metric is guaranteed if $\ell^{\mathcal{M}^{\text{in}}}(\mathbf{e}_{i,K})$ is close to 1.

Tessellation $\mathcal{T}_{h,P}^{\text{adapt}}$ in Fig. 7 features more than 88.8% edges characterized by adimensional length $\ell^{\mathcal{M}^{\text{in}}}$ between 0.71 and 1.41, and 97.4% of tetrahedra with a quality factor $Q > 0.12$ ³. These results are comparable to those reported in [27], where quality and lengths are computed according to [44], and prove the effectiveness of **3DPAMA** in dealing with a simple analytic isotropic input metric.

³Statistics $Q > 0.12$ and quantities $\ell^{\mathcal{M}^{\text{in}}}$ are automatically computed by the analysis tool implemented in **mng3d**. We refer to [42, 39, 43] for more details.

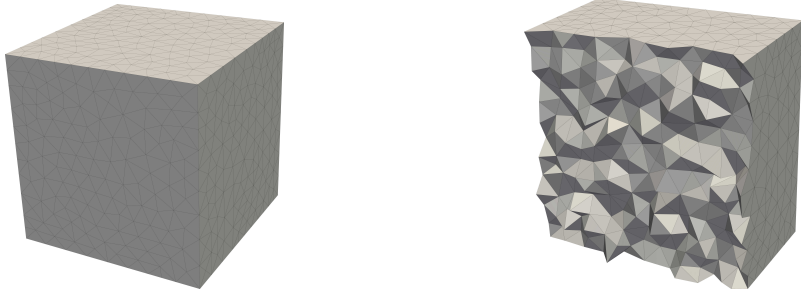


Figure 7: An analytic metric: external (left) and vertically-clipped (right) view of the adapted periodic mesh for the isotropic metric in (6).

In order to challenge the algorithm with a more difficult metric, the isotropically adapted mesh in Fig. 7 is then employed as the starting support mesh for another **3DPAMA** run, when an anisotropic analytic metric field is given. We select

$$f = \mathcal{M}^{\text{in}} = R^T \Lambda R, \quad (7)$$

where

$$\Lambda = \begin{pmatrix} h_{\max}^{-2} & 0 & 0 \\ 0 & h_{\max}^{-2} & 0 \\ 0 & 0 & h_3^{-2} \end{pmatrix}, \quad R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{z_2}{r} & -\frac{y_2}{r} \\ 0 & \frac{y_2}{r} & \frac{z_2}{r} \end{pmatrix},$$

with

$$y_2 = 2y - 1, \quad z_2 = 2z - 1, \quad r = \sqrt{y_2^2 + z_2^2},$$

$$h_3 = h_{\max} \left| 1 - e^{-2|r^2 - 0.36|} \right| + 0.0008, \quad h_{\max} = 0.1.$$

Metric \mathcal{M}^{in} – periodic along surface $\Gamma_P = S_2 \cup S_4$, since $\mathcal{M}^{\text{in}}|_{x=0} = \mathcal{M}^{\text{in}}|_{x=1}$ – prescribes a marked refinement of the mesh in correspondence with a cylindrical surface of radius 0.6, aligned with the x -direction.

Algorithm 1 returns the tessellation shown in Fig. 8, left panel, that is perfectly periodic, being the **3DPAMA** output **flag** value equal to 1. In the left and central panels, as expected by the imposition of the metric field \mathcal{M}^{in} , it is possible to appreciate that the elements are crowded in correspondence with cylindrical surface embedded in the cube. Moreover, by inspecting the horizontal cut in the right panel, we can spot anisotropic tetrahedra which are highly elongated in the x -direction, featuring a maximum aspect ratio $\max_K s|_K$ equal to $66.42 \gg 1$. Such elements are not noticeably affected by the operations in Phases 2–4, confirming that **3DPAMA** only resorts to minimal modifications of the boundary elements to yield the final result.

A quantitative analysis is reported in Table 1, gathering the performance indices on vertex and tetrahedra cardinality, quality, and adimensional length. Also in this scenario,

the values are comparable to those in [27] and show the effectiveness of **3DPAMA** in delivering high-quality anisotropically adapted periodic meshes matching the prescription of the input metric.

Vertex count	$\#\mathcal{T}_{h,P}^{\text{adapt}}$	$Q > 0.12$	$0.71 < \ell^{\mathcal{M}^{\text{in}}} < 1.41$
20126	111130	88.57%	75.21%

Table 1: An analytic metric: quantitative analysis characterizing the adapted periodic mesh obtained for the anisotropic metric in (7).

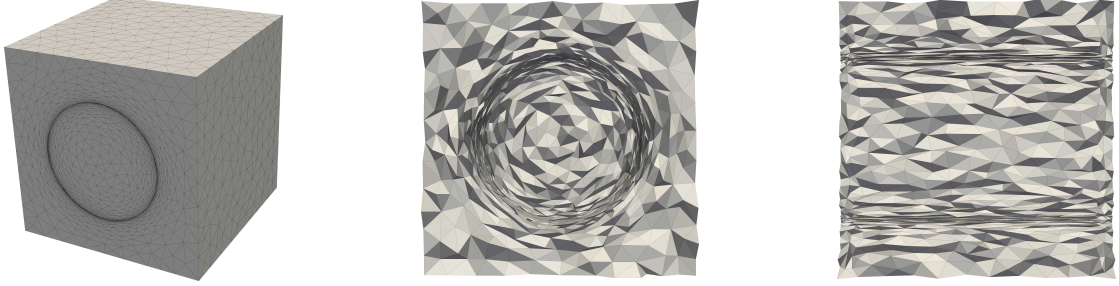


Figure 8: An analytic metric: external view (left), frontal (center), and lateral (right) section of the adapted periodic mesh for the anisotropic metric in (7).

3.2 A Hessian-based error estimator

We here extend the range of applicability of **3DPAMA**. In particular, the second test case is intended to generate a periodic mesh that is adapted to the function $f_\mu \in C^\infty(\Omega)$ defined over the domain $\Omega = (0, 2) \times (0, 1) \times (0, 1)$ as

$$f_\mu(x, y, z) = y(1 - y)\eta_\mu(x),$$

with

$$\eta_\mu(x) = \Psi_\mu\left(\frac{x-a}{b-a}\right)\Psi_\mu\left(\frac{d-x}{d-c}\right), \quad \Psi_\mu(x) = \frac{\phi_\mu(x)}{\phi_\mu(x) + \phi_\mu(1-x)}, \quad (8)$$

$$\phi_\mu(x) = \begin{cases} e^{-\frac{1}{\mu x}}, & x > 0 \\ 0, & x \leq 0, \end{cases}$$

and $a = 0$, $b = 2/3$, $c = 4/3$ and $d = 2$. Scalar function η_μ takes value 1 in the interval $[b, c]$ and vanishes outside the interval (a, d) , with parameter μ defining the steepness of the transition from 0 to 1 (see Fig. 9, left panel). As a consequence, function f_μ features marked internal layers along the planes $x = 1/3$ and $x = 5/3$, as $\mu \rightarrow 0$, and presents periodicity along all the boundary faces by construction (see Fig. 9, right panel, for the visualization on the plane $z = 1$).

We run **3DPAMA** algorithm by considering: a uniform starting mesh \mathcal{T}_h^0 consisting of 780000 elements; a full set of periodic boundary conditions, namely $I = L$, with faces

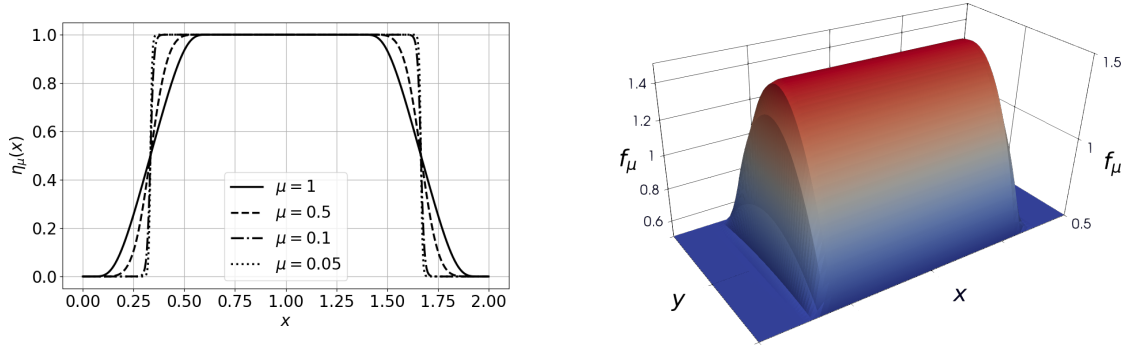


Figure 9: A Hessian-based error estimator: plot of function η_μ for different values of μ (left) and 3D plot of function f_μ on the plane $z = 1$ for $\mu = 0.20$ (right).

S_i , $i = 1, \dots, 6$, paired as $(1, \sigma(1) = 3)$, $(2, \sigma(2) = 4)$, $(5, \sigma(5) = 6)$, in accordance with the notation in Fig. 1, center panel; maps Θ_i as the rigid translations along the directions normal to S_i .

For the metric computation, we select `method` = ‘Hessian’, f coinciding with f_μ , the tolerance `tol` set to 0.008, `MTOL` = $1e - 2$, and `itmax` = 1. In this case, the routine `ComputeMetric` in line 3 of Algorithm 1 coincides with the `mshmet` plugin, provided in the `FreeFEM` library [36] that implements a feature-based error estimator recurrent in the literature on mesh adaptation, see, e.g., [45]. In more detail, starting from function f_μ and the associated interpolation on the discrete space $V_h^1(\mathcal{T}_h^0)$, the routine computes an optimal metric $\mathcal{M}_h^{\text{in}}$ that is based on the discrete Hessian and that surrogates an L^2 -norm approximation error associated with f_μ . As a result, metric $\mathcal{M}_h^{\text{in}}$ highlights the portions of the domain where the mesh should be refined or coarsened according to the information on f_μ gathered through the Hessian analysis. The user-defined tolerance parameter `tol` can be used to tune the severity of the adaptation operations, leading to a family of anisotropically adapted meshes that have increasingly higher tetrahedra counts for smaller values of `tol`.

Figure 10, left panel, shows the exactly periodic output of the `3DPAMA` algorithm for `tol` = 0.008 and $\mu = 0.05$. As expected by the plot of function f_μ in Fig. 9, right panel, we observe that the elements are massively crowded in the portions of the domain where internal and boundary layers develop. Specifically, the zooms in the circles highlight the use of differently-sized elements, the smallest tetrahedra being deformed ($\max_K s|_K = 6.48$) and aligned with the high gradient of the function.

We corroborate this analysis by plotting the convergence trend of the $L^2(\Omega)$ -error associated with the finite element discretization of f_μ as a function of $(\#\mathcal{T}_{h,P}^{\text{adapt}})^{-1/3}$, according to [45]. To this aim, we run `3DPAMA` for the values of $\mu = 0.20, 0.10$ and 0.05 in order to tune the steepness of the layers, while we choose values of the tolerance `tol` between 0.032 and 0.001 to vary the cardinality of the resulting adapted periodic meshes. The logarithmic plot in Fig. 10, right panel, confirms that the error is reduced following the theoretical quadratic order (see, for instance, [34]) for all the chosen values of μ , meaning that the boundary manipulations for periodicity enforcement do not jeopardize

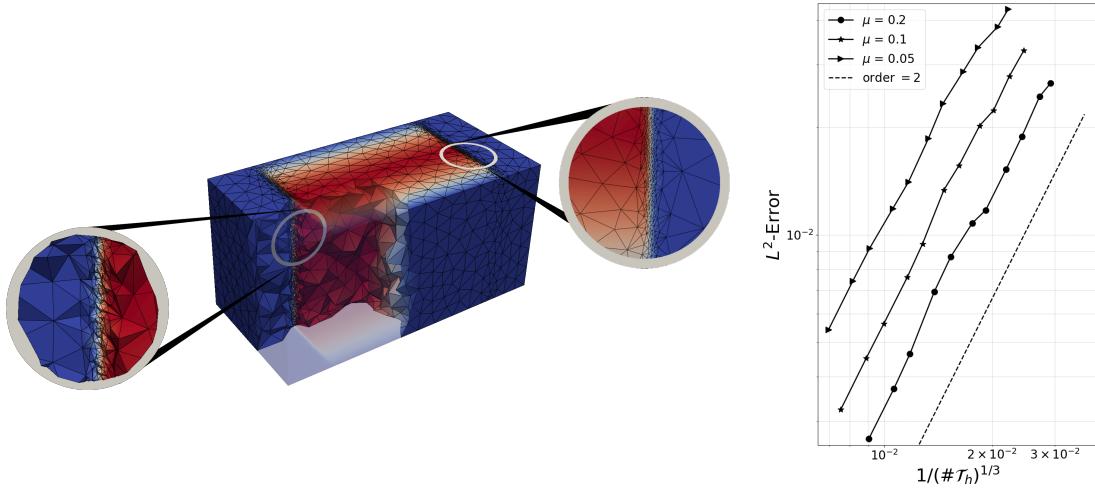


Figure 10: A Hessian-based error estimator: function f_μ superimposed to the anisotropically adapted periodic mesh $\mathcal{T}_{h,P}^{\text{adapt}}$ together with two enlarged views (left), $L^2(\Omega)$ -norm of the interpolation error as a function of $(\#\mathcal{T}_{h,P}^{\text{adapt}})^{-1/3}$ (right).

the quality of the grids and of the approximations. We also notice that the smaller the values of μ , the higher the interpolation error. This fact is reasonable, since steep gradients are responsible for high discretization error and require a large number of elements.

3.2.1 Other Geometries

This test case can be further exploited to demonstrate the flexibility of the proposed algorithm. Indeed, besides the capability to handle different numbers of periodic boundary conditions as shown in Section 3.1 where $\#I = 2$ and in Section 3.2 where $\#I = 6$, it is possible to vary geometries, aiming towards realistic engineering applications on complex domains.

In Table 2, we summarize progressively more convoluted examples. The table is structured as follows: the first column reports the name of the example; the second column highlights the number of periodic faces involved in the simulation; in the third column, we gather the periodicity maps Θ_i ; and the last column shows a clipped plot of the anisotropically adapted periodic mesh.

The first three examples present slight modifications to the test case of the previous section. In particular, we consider the domain $\Omega_0 = (0, 2) \times (0, 1) \times (0, 1)$ with the same pairs of boundary labels given in Fig. 1. Then, for each experiment, we apply a different displacement function $T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ to obtain the deformed domain $\Omega = T(\Omega_0)$ and function $g_\mu = f_\mu \circ T$.

The first test case is obtained by applying the displacement map $T(x, y, z) = [x+z, y+z, z]^T$. The resulting domain Ω corresponds to an inclined prism, with faces S_1, \dots, S_4 tilted uniformly by 45° with respect to the plane $z = 0$. We run 3DPAMA algorithm

with a starting mesh \mathcal{T}_h^0 consisting of 780000 elements that discretize the domain Ω , a complete set of periodic boundary conditions, with maps Θ_i listed in Table 2, first row third column, `method` = ‘Hessian’, f coinciding with $g_\mu = f_\mu(T(x, y, z))$, with $\mu = 0.05$, tolerance `tol` set to 0.008, `MTOL` = $1e-2$, and `itmax` = 1. The resulting adapted mesh shown in the first row fourth column is fully periodic and consists of 131863 elements, which are unevenly distributed across Ω . As prescribed by f , the smallest tetrahedra align with the inclined internal layers.

In the second example, map T coincides with $T(x, y, z) = [x + 0.1 \sin(2\pi z), y + 0.1 \sin(2\pi z), z]^T$ so that domain $\Omega = T(\Omega_0)$ has faces S_1, \dots, S_4 curved by a sinusoidal perturbation dependent on z . We run **3DPAMA** algorithm with the same parameters as the previous example, and by choosing $f = g_\mu = f_\mu(T(x, y, z))$. The periodic adapted mesh shown in Table 2, second row fourth column, is composed of 125899 tetrahedra. In this case, the refinement occurs in correspondence with the layers, as already observed in the other case, but also depends on the curvature of the geometry. This fact is automatically taken into consideration by the employed routines, proving that the boundary mesh manipulations in Phases 2 and 3 do not compromise the approximation of the geometry of the domain.

Concerning the third test case, for the sake of simplicity we omit the definition of map T . For the construction of domain Ω , we start from the reference domain Ω_0 and we add two hemispheres on the top surface, each characterized by a radius $R = 1/4$ centered at $(1/3, 1/2, 1)$ and $(5/3, 1/2, 1)$, respectively. In order to maintain the periodicity of the domain, we further modify the domain by removing the same hemispherical shapes from the bottom part of the domain, creating a Lego-type brick. With this configuration, we run **3DPAMA** algorithm by setting the same parameters as in the first example, starting from a mesh \mathcal{T}_h^0 composed of 879800 elements and by choosing $f = f_\mu$ as in Sect. 3.2. The procedure successfully complete the requested mesh adaptation and delivers an adapted periodic mesh of 311395 tetrahedra, shown in Table 2, third row fourth column.

Finally, the last example involves a change of coordinates and reference system. In particular, we consider a sphere centered in $(0, 0, 0)$ as domain Ω , with an external radius of length equal to 3 and an internal spherical hole with a unitary radius. In this configuration, we consider the spherical coordinates (r, θ, ϕ) , where r is the radial distance, θ is the polar angle, and ϕ is the azimuthal angle. Denoting the internal and external spherical surfaces with S_1 and S_2 , respectively, the periodic boundary conditions that we apply consist in the translational periodicity of S_1 towards S_2 along the radial direction, formalized in maps Θ_1 and Θ_2 , reported in Table 2, fourth row third column. As for the function f , we apply a change of coordinates to f_μ , leading to

$$f = h_\mu(r, \theta, \phi) = \Psi_\mu\left(\frac{r-a}{b-a}\right) \Psi_\mu\left(\frac{d-r}{d-c}\right),$$

with Ψ_μ as in (8), $a = 1, b = 5/3, c = 7/3$ and $d = 3$. Such function features two concentric internal layers, one close to the inner hole at $r = 4/3$ and one proximal to the external surface at $r = 8/3$.

By running **3DPAMA** with a starting mesh \mathcal{T}_h^0 consisting of 1093566 elements, by picking $\mu = 0.5$, and by preserving all the other input parameters, the algorithm effectively

deals with the change of coordinate system and with the holed domain, returning an adapted periodic mesh consisting of 787296 tetrahedra, massively crowded in correspondence with the internal layers of h_μ . With reference to the fourth row fourth column of Table 2, we remark that the mesh adaptation procedure did operate on the external surfaces S_1 and S_2 with few localized operations leading to the surface vertex counts changing from 5780 to 5902, even though these modifications are poorly visible in the final adapted mesh.

3.3 A ZZ-type recovery-based error estimator

The last test case is meant to emphasize the applicability of **3DPAMA** in a practical engineering scenario. In particular, we aim to solve a time-dependent parabolic equation by resorting to a finite element scheme that is based on a tailored computational mesh. Inspired by the 2D setting in [46, 47], we consider a 3D equation that models the heat flow, $u(\mathbf{x}, t)$, conducted in a periodically arranged specimen heated along the z -direction. Specifically, in the space domain $\Omega = (-2, 2) \times (-2, 2) \times (0, 1)$ throughout the time window $J = (0, 4]$, we aim at solving

$$\begin{cases} d_t u(\mathbf{x}, t) - \Delta u(\mathbf{x}, t) = f(\mathbf{x}, t) & \text{in } \Omega \times J, \\ u(\mathbf{x}, t) = 0 & \text{in } \Gamma_D \times J, \\ \text{PBCs} & \text{in } \Gamma_P \times J, \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}) & \text{in } \Omega, \end{cases} \quad (9)$$

where translationals periodic boundary conditions (PBCs) are assigned on the portion Γ_P of the space domain boundary, $\partial\Omega$, with $\Gamma_P = \{(x, y, z) \in \partial\Omega : z = 0\} \cup \{(x, y, z) \in \partial\Omega : z = 1\}$, i.e., the top and bottom faces of the domain, labelled with the pair (5, 6), according to Fig. 1. Moreover, homogeneous Dirichlet boundary conditions are prescribed on $\Gamma_D = \partial\Omega \setminus \Gamma_P$. The source term $f(\mathbf{x}, t)$ and the initial condition $u_0(\mathbf{x})$ are chosen such that the exact solution to (9) is

$$u(\mathbf{x}, t) = \cos\left(\frac{\pi x}{4}\right) \cos\left(\frac{\pi y}{4}\right) \tanh\left[10\left(y - \cos\left(\frac{\pi t}{2}\right) \sin(\pi x)\right)\right]. \quad (10)$$

We observe that the solution is z -independent and exhibits an internal layer changing shape, position and extension in time, following the sinusoidal function

$$y = \cos\left(\frac{\pi t}{2}\right) \sin(\pi x).$$

Considering the standard Sobolev function space $H_{0,\Gamma_D}^1(\Omega)$ [1], the weak formulation of problem (9) reads

find $u(t) = u(\cdot, t) \in H_{0,\Gamma_D}^1(\Omega)$ such that, $\forall t \in J$,

$$\begin{cases} \int_{\Omega} d_t u(t) v \, d\Omega + \int_{\Omega} \nabla u(t) \cdot \nabla v \, d\Omega = \int_{\Omega} f(\mathbf{x}, t) v \, d\Omega & \forall v \in H_{0,\Gamma_D}^1(\Omega) \\ u(0) = u_0(\mathbf{x}), \end{cases} \quad (11)$$

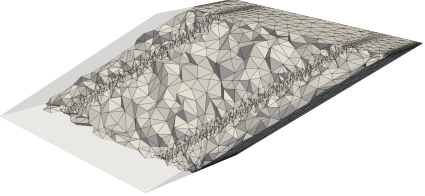
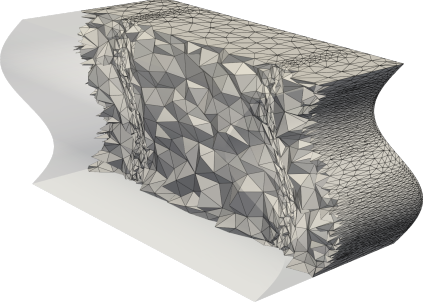
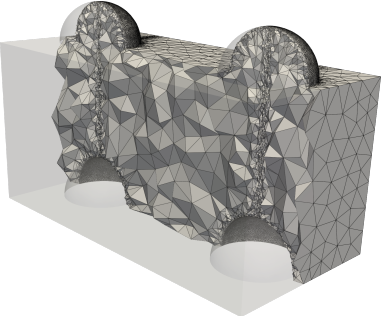
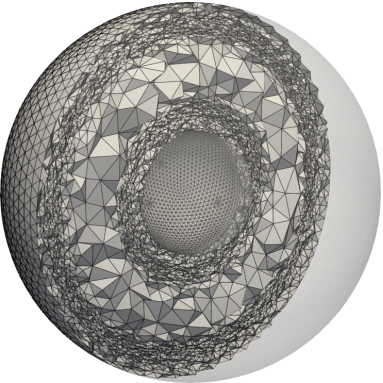
Test case	#I	Periodicity maps	Adapted periodic mesh
Inclined prism	6	$\Theta_1 = [x, y + 1, z]^T$ $\Theta_2 = [x, y - 1, z]^T$ $\Theta_3 = [x - 2, y, z]^T$ $\Theta_4 = [x + 2, y, z]^T$ $\Theta_5 = [x + 1, y + 1, z - 1]^T$ $\Theta_6 = [x - 1, y - 1, z + 1]^T$	
Sinusoidal perturbation	6	$\Theta_1 = [x, y + 1, z]^T$ $\Theta_2 = [x, y - 1, z]^T$ $\Theta_3 = [x - 2, y, z]^T$ $\Theta_4 = [x + 2, y, z]^T$ $\Theta_5 = [x, y, z - 1]^T$ $\Theta_6 = [x, y, z + 1]^T$	
Lego-type brick	6	$\Theta_1 = [x, y + 1, z]^T$ $\Theta_2 = [x, y - 1, z]^T$ $\Theta_3 = [x - 2, y, z]^T$ $\Theta_4 = [x + 2, y, z]^T$ $\Theta_5 = [x, y, z - 1]^T$ $\Theta_6 = [x, y, z + 1]^T$	
Concentric spheres	2	$\Theta_1(r, \theta, \phi) = [r + 2, \theta, \phi]^T$ $\Theta_2(r, \theta, \phi) = [r - 2, \theta, \phi]^T$	

Table 2: A Hessian-based error estimator - other geometries.

with $u(t)$ satisfying PBCs on Γ_P .

Upon discretizing Ω with a tetrahedral mesh \mathcal{T}_h^0 and partitioning J through m equally spaced subintervals of length $\Delta t = t^n - t^{n-1}$, t^n being the generic discrete time instant, $n = 1, \dots, m$, we cast problem (11) in a discrete setting by resorting to the finite element method and to an implicit Euler scheme for the time stepping, yielding for $n = 1, \dots, m$, find $u_h^n \in V_h = V_h^1(\mathcal{T}_h^0) \cap H_{0,\Gamma_D}^1(\Omega)$ such that

$$\begin{cases} \int_{\Omega} \frac{u_h^n - u_h^{n-1}}{\Delta t} v_h + \int_{\Omega} \nabla u_h^n \cdot \nabla v_h \, d\Omega = \int_{\Omega} f(\mathbf{x}, t^n) v_h \, d\Omega & \forall v_h \in V_h \\ u_h^0 = u_{h,0}, \end{cases} \quad (12)$$

with u_h^n the approximation of $u(t)$ at time $t = t^n$ that is periodic along Γ_P , and where $u_{h,0}$ is a suitable approximation of u_0 in V_h .

In order to leverage the capabilities of **3DPAMA** in constructing periodic grids, we generate a sequence of anisotropic meshes $\{\mathcal{T}_{h,P}^n\}_{n=0,\dots,m}$ adapted to the solution of (12) at each time instant.

To this aim, at $t = t^n$, we first compute u_h^n by solving (12) on the grid $\mathcal{T}_{h,P}^{n-1}$ and with a fixed time-step $\Delta t = 1/50$. Successively, we apply a mesh adaptation step, by invoking Algorithm 1 as

$$[\mathcal{T}_{h,P}^n, \text{flag}] = \text{3DPAMA}(\mathcal{T}_{h,P}^{n-1}, I, \sigma, \{\Theta_i\}_{i \in I}, \text{'ZZ'}, u_h^n, \text{tol}, \text{MTOL}, \text{itmax}),$$

The input parameters of this **3DPAMA** call are thus set: $I = (5, 6)$, $\Theta_5 = [x, y, z + 1]^T$, $\Theta_6 = [x, y, z - 1]^T$, $\text{tol} = 1$, $\text{MTOL} = 1e - 3$, and $\text{itmax} = 10$.

Unlike the previous examples, mesh adaptation is here guided by a recovery-based Zienkiewicz-Zhu error estimator [15, 48], known to be suitable for gradient localization. Specifically, exploiting the anisotropic setting introduced in [49], we code a in-house implementation of the anisotropic version proposed in [50, 51] and in [31] for the 2D and 3D case, respectively. This estimator is cheaper to compute and easier to implement with respect to a Hessian approach and have already been successfully employed in different applications, e.g., in fluid dynamics [46, 52] or in topology optimization [19, 25].

The numerical solutions and the corresponding anisotropically adapted periodic grids are depicted in Fig. 11, left panel, for $t = 0$, corresponding to the initial condition defined on the starting uniform mesh characterized by 194800 elements, $t = 4/5, 1, 6/5, 2$, and at the final time $t = 4$. It is possible to observe that the meshes closely follow the dynamics of the system, leading to a marked refinement close to the moving sinusoidal shape and to an evident coarsening in the remaining portions of the domain. The corresponding solutions sharply capture the expected behavior, showcasing high resolution in correspondence with the sinusoidal gradient. Figure 11, right panel, shows the evolution in time of the number of mesh elements and vertices. The plot corroborates the efficacy of **3DPAMA** in effectively tracking the time-dependent phenomenon, the grid cardinality following the time pattern of solution u . Moreover, the oscillatory trend of the number of elements reveals that **3DPAMA** reduces the computational burden associated with a finite element solver based on a fixed grid. Indeed, in correspondence with the time

instants where the dynamics does not require extra-resolution, the algorithm effectively lowers vertex and element counts (see, for instance, the cardinality at $t = 1$ and $t = 3$ that is much lower than the initial one).

Finally, the adaptation capabilities of **3DPAMA** combined with the ZZ recovery-based error estimator are confirmed by Fig. 12 where different sections of the mesh-solution pair are shown for $t = 4$. The zooms in the internal part of the domain reveal that the enforcement of periodicity does not compromise the adaptation performance, the grid featuring crowded and highly elongated elements, with a maximum aspect ratio $\max_K s|_K = 8.92$, located in correspondence with the gradient of the solution, in agreement with the findings in [46, 47].

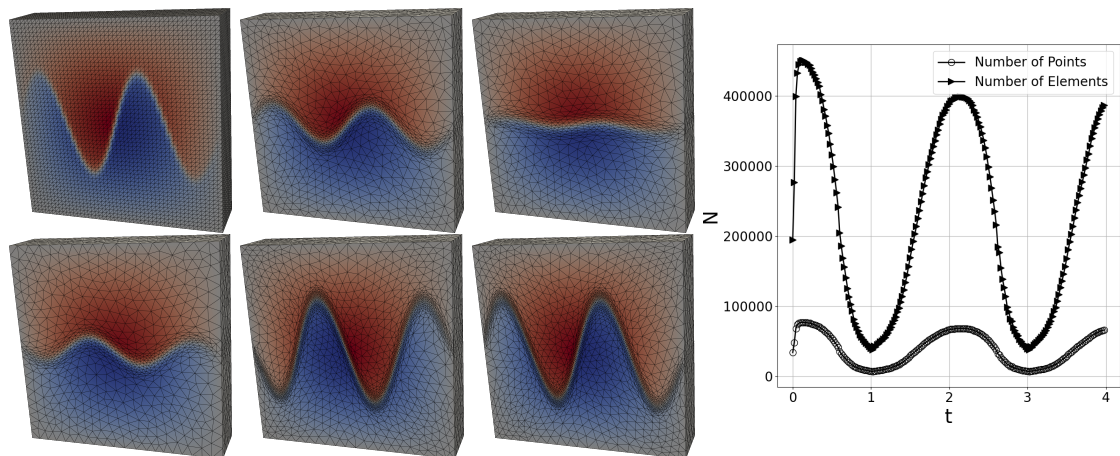


Figure 11: A ZZ-type recovery-based error estimator: solution u_h superimposed to the anisotropically adapted periodic mesh for $t = 0, 4/5, 1, 6/5, 2, 4$, (first panel, left-right, top-bottom), evolution in time of the number of vertices and elements of the adapted meshes (second panel).

4 Conclusions

In this work, we presented a novel algorithm, named **3DPAMA**, designed to leverage standard mesh generation techniques to produce anisotropically adapted periodic 3D tetrahedral tessellations. The steps of the procedure are based on a theoretically sound analysis on the periodicity maps and include localized volumetric and surface operations, to guarantee that the periodic boundary portions have matching mesh entities (i.e., vertices, faces and edges). Differently from previous approaches, the proposed method enforces periodicity through an internal splitting of the mesh, thereby avoiding any intermediate steps that may involve alterations or movements in the reference domain.

The numerical test campaign has demonstrated the effectiveness of the periodic mesh adaptation strategy across various benchmark cases, as well as in comparison with the limited existing literature on this topic. The resulting periodic meshes turned out to

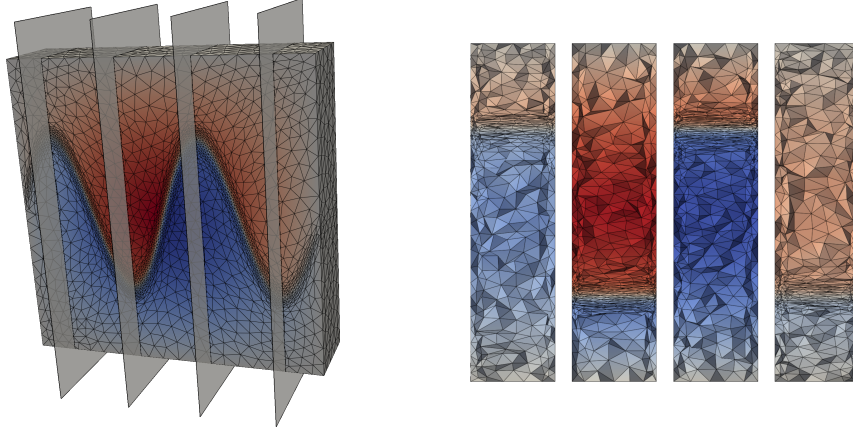


Figure 12: A ZZ-type recovery-based error estimator: solution u_h superimposed to the anisotropically adapted periodic mesh at $t = 4$ (first panel), corresponding cut views for the sections $y = -1.5, -0.5, 0.5, 1.5$ (second panel, left-right).

be valid for the employment in finite element frameworks and featured high quality elements, possibly deformed in presence of anisotropic sources. Moreover, Algorithm **3DPAMA** proved to be:

- robust with respect to different domain geometries and types of periodic boundary conditions;
- flexible in the choice of the method to compute the metric field that drives the adaptation (namely, analytic metric, Hessian-based estimator, Zienkiewicz-Zhu error analysis);
- easy-to-implement, since requiring basic mesh manipulation strategies implemented in well-established software tools. This feature facilitates the integration into existing finite element code workflows.

Future developments of this work can further broaden the applicability and efficiency of our algorithm, making it a versatile tool for a wide range of applications in computational modeling and simulation. In particular, it is planned to work on an automatic detection of the reference maps Θ_i based on the specification of σ only, so as to reduce the input parameters. Moreover, we will target the generalization of the strategy to other metric computation, e.g., by resorting to goal-oriented error indicators. Finally, a parallel implementation of the algorithm is planned, as well as the development of a standalone software package that can be seamlessly interfaced with any finite element suite.

Acknowledgements

GS and NF are members of the Gruppo Nazionale Calcolo Scientifico–Istituto Nazionale di Alta Matematica (GNCS–INdAM). NF acknowledges the INdAM–GNCS 2024 Project “Accuratezza geometrica e adattività di griglia per problemi con interfacce complesse”.

Declarations

The authors declare that they have no conflict of interest.
No data was used for the research described in the article.

References

- [1] A. Ern, J.-L. Guermond, Theory and practice of finite elements, Vol. 159, Springer, 2004.
- [2] A. Quarteroni, A. Valli, Numerical approximation of partial differential equations, Vol. 23, Springer Science & Business Media, 2008.
- [3] G. Chen, J. Zhou, Boundary element methods with applications to nonlinear problems, Vol. 7, Springer Science & Business Media, 2010.
- [4] L. Beirão da Veiga, F. Brezzi, L. D. Marini, A. Russo, The hitchhiker’s guide to the virtual element method, *Math. Models Methods Appl. Sci.* 24 (08) (2014) 1541–1573.
- [5] F. Moukalled, L. Mangani, M. Darwish, F. Moukalled, L. Mangani, M. Darwish, The finite volume method, Springer, 2016.
- [6] W. Bangerth, R. Rannacher, Adaptive finite element methods for differential equations, Springer Science & Business Media, 2003.
- [7] W. Cao, W. Huang, R. D. Russell, An r-adaptive finite element method based upon moving mesh PDEs, *J. Comput. Phys.* 149 (2) (1999) 221–244.
- [8] I. Babuška, M. Suri, The p and h-p versions of the finite element method, basic principles and properties, *SIAM Rev.* 36 (4) (1994) 578–632.
- [9] J. Oden, L. Demkowicz, hp adaptive finite element methods in computational fluid dynamics, *Comput. Methods Appl. Mech. Eng.* 89 (1-3) (1991) 11–40.
- [10] F. Alauzet, A. Loseille, A decade of progress on anisotropic mesh adaptation for computational fluid dynamics, *Comput.-Aided Des.* 72 (2016) 13–39.
- [11] P.-L. George, H. Borouchaki, P.-J. Frey, P. Laug, E. Saltel, Mesh Generation and Mesh Adaptivity, John Wiley & Sons, Ltd, 2004, Ch. 17.

- [12] M. Ainsworth, J. Oden, A posteriori error estimation in finite element analysis, *Comput. Methods Appl. Mech. Eng.* 142 (1) (1997) 1–88.
- [13] J. Peraire, M. Vahdati, K. Morgan, O. C. Zienkiewicz, Adaptive remeshing for compressible flow computations, *J. Comput. Phys.* 72 (2) (1987) 449–466.
- [14] P.-J. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, *Comput. Methods Appl. Mech. Eng.* 194 (48) (2005) 5068–5082.
- [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] B. T. Helenbrook, Mesh deformation using the biharmonic operator, *Int. J. Numer. Methods Eng.* 56 (7) (2003) 1007–1021.
- [17] Ł. Kaczmarczyk, M. M. Nezhad, C. Pearce, Three-dimensional brittle fracture: configurational-force-driven crack propagation, *Int. J. Numer. Methods Eng.* 97 (7) (2014) 531–550.
- [18] N. Ferro, S. Micheletti, S. Perotto, Anisotropic mesh adaptation for crack propagation induced by a thermal shock in 2D, *Comput. Methods Appl. Mech. Eng.* 331 (2018) 138–158.
- [19] S. Micheletti, S. Perotto, L. Soli, Topology optimization driven by anisotropic mesh adaptation: Towards a free-form design, *Comput. Struct.* 214 (2019) 60–72.
- [20] D. Cortellessa, N. Ferro, S. Perotto, S. Micheletti, Enhancing level set-based topology optimization with anisotropic graded meshes, *Appl. Math. Comput.* 447 (2023) 127903.
- [21] M. Piggott, P. Farrell, C. Wilson, G. Gorman, C. Pain, Anisotropic mesh adaptivity for multi-scale ocean modelling, *Philos. Trans. R. Soc. A* 367 (1907) (2009) 4591–4611.
- [22] L. Frazza, A. Loseille, F. Alauzet, Unstructured anisotropic mesh adaptation for 3D RANS turbomachinery applications, in: *AIAA Aviation 2019 Forum*, 2019, p. 3059.
- [23] Z. Zhao, S. Kuchnicki, R. Radovitzky, A. Cuitiño, Influence of in-grain mesh resolution on the prediction of deformation textures in fcc polycrystals by crystal plasticity FEM, *Acta Mater.* 55 (7) (2007) 2361–2373.
- [24] C. González, J. Segurado, J. LLorca, Numerical simulation of elasto-plastic deformation of composites: evolution of stress microfields and implications for homogenization models, *J. Mech. Phys. Solids* 52 (7) (2004) 1573–1593.
- [25] 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.

- [26] J. K. Guest, J. H. Prévost, Optimizing multifunctional materials: design of microstructures for maximized stiffness and fluid permeability, *Int. J. Solids Struct.* 43 (22-23) (2006) 7028–7047.
- [27] C. Dobrzynski, M. Melchior, L. Delannay, J.-F. Remacle, A mesh adaptation procedure for periodic domains, *Int. J. Numer. Methods Eng.* 86 (12) (2011) 1396–1412.
- [28] É. Béchet, N. Moës, B. Wohlmuth, A stable Lagrange multiplier space for stiff interface conditions within the extended finite element method, *Int. J. Numer. Methods Eng.* 78 (8) (2009) 931–954.
- [29] A. Nagarajan, S. Soghrati, Conforming to interface structured adaptive mesh refinement: 3D algorithm and implementation, *Comput. Mech* 62 (2018) 1213–1238.
- [30] P.-L. George, H. Borouchaki, *Delaunay Triangulation and Meshing*. 1. Vyd. Paris, Hermes (1998).
- [31] 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.
- [32] N. Ferro, S. Micheletti, S. Perotto, An optimization algorithm for automatic structural design, *Comput. Methods Appl. Mech. Eng.* 372 (2020) 113335.
- [33] F. Alauzet, P.-J. Frey, Estimateur d’erreur geometrique et metriques anisotropes pour l’adaptation de maillage. partie i: aspects theoriques, Ph.D. thesis, INRIA (2003).
- [34] F. Magoulès, *Computational fluid dynamics*, CRC Press, 2011.
- [35] S. Perotto, L. Formaggia, et al., *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, Springer, 2015.
- [36] F. Hecht, New development in FreeFem++, *J. Numer. Math.* 20 (3-4) (2012) 251–265.
- [37] A. Le Hyaric, P. Jolivet, S. Garnotel, Freefem, version 4.14, <https://github.com/FreeFem/FreeFem-sources>, accessed: 8/8/2024 (2023).
- [38] C. Dapogny, C. Dobrzynski, P.-J. Frey, Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems, *J. Comput. Phys.* 262 (2014) 358–378.
- [39] A. Froehly, mmg, version 5.7.0, <https://github.com/MmgTools/mmg>, accessed: 8/8/2024 (2022).
- [40] H. Si, TetGen, a Delaunay-based quality tetrahedral mesh generator, *ACM Trans. Math. Softw.* 41 (2) (2015) 11.

- [41] H. Si, A Quality Tetrahedral Mesh Generator and 3D Delaunay Triangulator, <https://wias-berlin.de/software/index.jsp?id=TetGen&lang=1>, version 1.6. Accessed: 8/8/2024 (2020).
- [42] G. Balarac, F. Basile, P. Bénard, F. Bordeu, J.-B. Chapelier, L. Cirrottola, G. Caumon, C. Dapogny, P.-J. Frey, A. Froehly, et al., Tetrahedral remeshing in the context of large-scale numerical simulation and high performance computing, *Math. In Action* 11 (1) (2022) 129–164.
- [43] A. Loseille, Adaptation de maillage anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluides. Application à la prédiction haute-fidélité du bang sonique., Ph.D. thesis, Université Pierre et Marie Curie-Paris VI (2008).
- [44] C. Dobrzynski, P.-J. Frey, Anisotropic Delaunay mesh adaptation for unsteady simulations, in: *Proceedings of the 17th international Meshing Roundtable*, Springer, 2008, pp. 177–194.
- [45] L. Chen, P. Sun, J. Xu, Optimal anisotropic meshes for minimizing interpolation errors in L^p -norm, *Math. Comput.* 76 (257) (2007) 179–204.
- [46] S. Micheletti, S. Perotto, Space time adaptation for purely diffusive problems in an anisotropic framework, *Int. J. Numer. Anal. Model.* 7 (1) (2010).
- [47] N. Ferro, S. Perotto, A. Cangiani, An anisotropic recovery-based error estimator for adaptive discontinuous Galerkin methods, *J. Sci. Comput.* 90 (1) (2022) 45.
- [48] 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.
- [49] L. Formaggia, S. Perotto, New anisotropic a priori error estimates, *Numer. Math.* 89 (4) (2001) 641–667.
- [50] S. Micheletti, S. Perotto, Reliability and efficiency of an anisotropic Zienkiewicz–Zhu error estimator, *Comput. Methods Appl. Mech. Eng.* 195 (9) (2006) 799–835.
- [51] S. Micheletti, S. Perotto, Anisotropic adaptation via a Zienkiewicz–Zhu error estimator for 2D elliptic problems, in: *Numerical Mathematics and Advanced Applications 2009: Proceedings of ENUMATH 2009, the 8th European Conference on Numerical Mathematics and Advanced Applications*, Uppsala, July 2009, Springer, 2010, pp. 645–653.
- [52] S. Micheletti, S. Perotto, P. Farrell, A recovery-based error estimator for anisotropic mesh adaptation in CFD, *SeMA J.* 50 (1) (2010) 115 – 137.

MOX Technical Reports, last issues

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

- 59/2024** Carbonaro, D.; Ferro, N.; Mezzadri, F.; Gallo, D.; Audenino, A.; Perotto, S.; Morbiducci, U.; Chiastra, C.
Easy-to-use formulations based on the homogenization theory for vascular stent design and mechanical characterization
- 60/2024** Temellini, E.; Ferro, N.; Stabile, G.; Delgado Avila, E.; Chacon Rebollo, T.; Perotto, S.
Space - time mesh adaptation for the VMS - Smagorinsky modeling of high Reynolds number flows
- 58/2024** Ciaramella, G.; Vanzan, T.
Variable reduction as a nonlinear preconditioning approach for optimization problems
- Speroni, G.; Ferro, N.
A novel metric-based mesh adaptation algorithm for 3D periodic domains
- Temellini, E.; Ferro, N.; Stabile, G.; Delgado Avila, E.; Chacon Rebollo, T.; Perotto, S.
Space-time mesh adaptation for the VMS-Smagorinsky modeling of high Reynolds number flows
- 56/2024** Parolini, N.; Covello, V.; Della Rocca, A.; Verani, M.,
Design of a checkerboard counterflow heat exchanger for industrial applications
- 57/2024** Negrini, G.; Parolini, N.; Verani, M.
An Immersed Boundary Method for Polymeric Continuous Mixing
- 55/2024** Artoni, A.; Ciaramella, G.; Gander, M.J.; Mazzieri, I.
Schwarz Waveform Relaxation and the Unmapped Tent-Pitching Method in 3D
- 54/2024** Antonietti, P.F.; Botti, M.; Cancrini, A.; Mazzieri, I.
A polytopal discontinuous Galerkin method for the pseudo-stress formulation of the unsteady Stokes problem
- 53/2024** Caldana, M.; Hesthaven, J. S.
Neural ordinary differential equations for model order reduction of stiff systems