Topology optimization of curved thick shells using level set method and non-conforming multi-patch isogeometric analysis

Fernando Hübner Scherer, Malek Zarroug, Hakim Naceur, Andrei Constantinescu

To cite this version:

Fernando Hübner Scherer, Malek Zarroug, Hakim Naceur, Andrei Constantinescu. Topology optimization of curved thick shells using level set method and non-conforming multi-patch isogeometric analysis. Computer Methods in Applied Mechanics and Engineering, 2024, 430, pp.117205. 10.1016/j.cma.2024.117205 . hal-04758603

HAL Id: hal-04758603
https://hal.science/hal-04758603v1

Submitted on 18 Nov 2024

Topology optimization of curved thick shells using level set method and non-conforming multi-patch isogeometric analysis

Fernando Hübner[a,b,c], Malek Zarroug[b], Hakim Naceur[c], Andrei Constantinescu[a,*]

[a]*Laboratoire de Mécanique des Solides, CNRS, École Polytechnique, 91128, Palaiseau, France*
[b]*Direction Scientifique et Technologies Futures, Groupe Stellantis, Route de Gisy, Velizy-Villacoublay, 78140, France*
[c]*INSA Hauts-de-France, Valenciennes, 59300, France*

Abstract

We present a novel framework for topological shape optimization of curved thick-shells subjected to external loads. Our method integrates the level set method (LSM) with a diffuse interface, a Hadamard shape derivative, and multi-patch isogeometric analysis (IGA) into a gradient descent algorithm to systematically capture the evolution of the shape. This integration enables us to directly manipulate CAD-compatible geometries and analysis techniques and to obtain the results as a CAD surface. The novelty lies in the utilization of multi-patch IGA models based on NURBS functions, which allows us to simultaneously maximize the stiffness and minimize the volume of the shell by searching for an optimal material distribution within its middle surface. The material is modeled under a small strain assumption in linear elasticity using a Reissner-Mindlin kinematic shell model in plane stress. The effectiveness of our approach is demonstrated on several curved conforming and non-conforming multi-patch geometries in 3D.

Keywords: Topology optimization, Isogeometric analysis, Level set method, Reissner-Mindlin shells, multi-patch

---

1. Introduction

The recent convergence of advanced shape optimization methods with innovative manufacturing technologies has opened up a new era for designing complex shapes that can simultaneously meet diverse mechanical constraints. However, despite this remarkable progress, there remain significant hurdles to the widespread adoption of these methods, namely, excessive computational time and incompatibilities between geometrical, computational, and design models.

Isogeometric analysis (IGA) is commonly used for defining a physical domain by means of parameterized curves called NURBS [1]. In optimization, however, it is unknown how the domain will evolve during the course of the iterations, which means that this explicit description of the physical domain will change as well. This could lead to

---

[*] Corresponding author

Email address: `andrei.constantinescu@polytechnique.edu` (Andrei Constantinescu)

Published in CMAME, 2024. DOI: https://doi.org/10.1016/j.cma.2024.117205

poor representation of the geometry, compromising the quality of the solution. To overcome this problem, this paper proposes to use the level set method (LSM) while still being able to benefit from key properties of IGA: parametric description of geometries and CAD compatibility. We will consider a fixed optimization domain, defined by the mid-surface (a NURBS surface) of a shell, and optimize the material distribution, which is defined implicitly by a level set function (LSF). The latter will also be numerically modeled as a NURBS, ensuring that the output of the method is a CAD model and that the boundaries of the domain are precisely defined. LSM has been vastly used for topology optimization, as known from some classic papers as [2, 3]. In the context of classic finite element method (FEM), some authors proposed frameworks to perform topology optimization under the shell model. For example, [4] used a density based method to minimize the compliance of a shell structure. Additionally, [5] proposed a comparison between the designs obtained from a compliance minimization and that of the maximum stress minimization in 2D plates. Similarly, an optimization using LSM was used by [6] to minimize the maximum stress, while taking into account the curvature of the geometry in three dimensional shells, also in the context of FEM. In [7], optimal microstructure for accounting simultaneously in-plane stiffness, out-of-plane stiffness and the extension–bending coupling effects in panels obtained by inverse homogenization, the Hadamard shape derivative and the level set method. In IGA, many authors have studied how to combine IGA and LSM for 2D linear elasticity, as in [8, 9, 10, 11, 12]. We believe, however, that a contribution for non-conforming multi-patch thick-shells and LSM is yet to be done. Out of the context of the LSM, [13] used IGA to optimize a spherical thin-shell using the phase-field method and [14] used trimmed IGA to perform parametric optimization on shells and used nucleation techniques to allow topology changes.

This paper is structured as follows: Section 2 defines the framework of a Reissner-Mindlin shell using IGA and explains how to express the strain and stress operators using the curvature of the geometry. Section 3 adapts the level set method to the setting of a parameterized non-conforming multi-patch geometry. Afterwards, Section 4 combines IGA and LSM approach for thick shells. Then, Section 5 presents the setting of the optimization problem. Finally, the numerical results are presented in Section 6.

2. Reissner-Mindlin shells with IGA

Let us consider the problem of the linear elastic thick shell under Reissner-Mindlin kinematic hypothesis solved using isogeometric analysis. This shell modeling approach will be employed to define an optimization domain $\mathcal{D}$, which will remain fixed throughout the optimization process and wherein the admissible shapes will be defined. It will be initially introduced as a single-patch geometry and subsequently extended to conforming and non-conforming multi-patch geometries. We refer to [15, 16] for further details on the development of an IGA solver for a linear elastic Reissner-Mindlin shell, and to [17], for the underlying principles of differential geometry and shell modeling.

Notation. Throughout the paper, Greek indices and exponents $\alpha, \beta, \mu, \lambda$ range in $\{1, 2\}$, whereas latin lowercase indices i, j, r, s vary in $\{1, 2, 3\}$, as well as the Einstein summation convention over repeated indices. Moreover, the symbol $(\hat{\ })$ is used to emphasize the parametric nature of a mathematical entity.

Let us now define mathematically the shell of mid-surface $\mathcal{D}$ embedded in $\mathbb{R}^3$ with constant thickness $\epsilon > 0$. A parameter domain $\hat{\mathcal{D}} \subset \mathbb{R}^2$ is introduced, over which the map $\mathbf{S} : \hat{\mathcal{D}} \to \mathbb{R}^3$ is defined through a NURBS parametrization. For simply connected domains, the parametric domain can be identified with the unit square in $\mathbb{R}^2$. The mid-surface $\mathcal{D}$ is then defined with $\mathcal{D} = \mathbf{S}(\hat{\mathcal{D}})$ and will also be referred to as *patch*.

In each point $(\xi_1, \xi_2) \in \hat{\mathcal{D}}$, we define a local covariant vector basis $\{\mathbf{a}_i\}$ with $\mathbf{a} = (\hat{\partial}_{\xi_1}\mathbf{S}, \hat{\partial}_{\xi_2}\mathbf{S}, \mathbf{n})^T$, where $\mathbf{n} = \dfrac{\hat{\partial}_{\xi_1}\mathbf{S} \times \hat{\partial}_{\xi_2}\mathbf{S}}{|\hat{\partial}_{\xi_1}\mathbf{S} \times \hat{\partial}_{\xi_2}\mathbf{S}|}$ and $|\cdot|$ designates the Euclidean norm. Moreover, the covariant basis of the mid-surface naturally defines a local orthonormal basis denoted as $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3)$ by normalizing the vectors $\{\mathbf{a}_i\}$, i.e.:

$$(\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3) = \left(\frac{\mathbf{a}_1}{|\mathbf{a}_1|}, \frac{\mathbf{a}_2}{|\mathbf{a}_2|}, \mathbf{n} \right) \tag{1}$$

Based on the covariant vector basis in each point of the mid-surface, and provided the thickness is sufficiently small with respect of the local curvature, one can construct the thick shell as the image of the *parametric volume map* $\boldsymbol{\mathcal{V}}$ given by:

$$\boldsymbol{\mathcal{V}} : \hat{V} \to \mathbb{R}^3 \qquad \boldsymbol{\mathcal{V}}((\xi_1, \xi_2), \xi_3) = \mathbf{S}(\xi_1, \xi_2) + \xi_3 \mathbf{n}(\xi_1, \xi_2), \tag{2}$$

where $\hat{V} := \hat{\mathcal{D}} \times \left[-\frac{\epsilon}{2}, \frac{\epsilon}{2}\right]$ designates the three-dimensional parameter domain.

Under the assumptions of C^3 regularity of the map $\mathbf{S}$ and a sufficiently small ϵ, the map $\boldsymbol{\mathcal{V}}$ is a 2-diffeomorphism between $\hat{V}$ and $V := \boldsymbol{\mathcal{V}}(\hat{V})$, as proven in [17]. Hence, the domains $\hat{V}$ and V are linked in a one-to-one fashion by the map $\boldsymbol{\mathcal{V}}$, meaning that a material point $\boldsymbol{x} = (x_i) \in V$ in the physical space has an unique representative in the parametric domain, i.e. $\exists! \boldsymbol{\xi} \in \hat{V}$ such that $\boldsymbol{x} = \boldsymbol{\mathcal{V}}(\boldsymbol{\xi})$. Moreover, to facilitate the readability of the equations, we will notate the derivatives with respect to the coordinates $\boldsymbol{\xi}$ by:

$$\hat{\partial}_\alpha := \frac{\partial}{\partial \xi_\alpha} \tag{3}$$

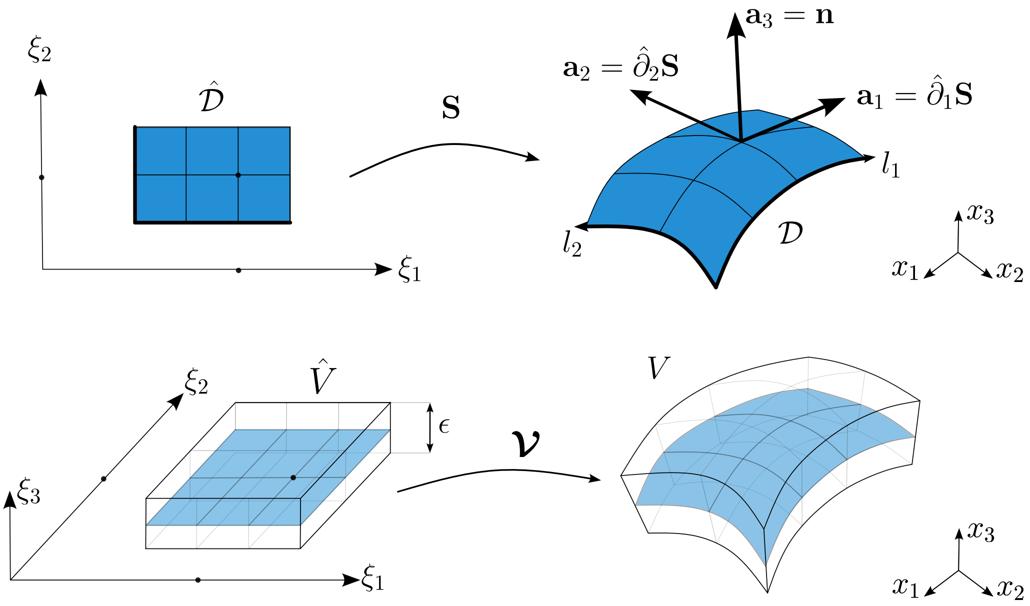

Figure 1: The construction of the thick shell: (top panel) The *mid-surface map* $\mathbf{S} : \hat{\mathcal{D}} \to \mathcal{D}$ transforms each point of the parametric domain $\hat{\mathcal{D}}$ into a point in the physical mid-surface $\mathcal{D}$. (bottom panel) The *parametric volume map* $\boldsymbol{\mathcal{V}} : \hat{V} \to V$ transforms each point of the parametric volume $\hat{V} = \hat{\mathcal{D}} \times \left[-\frac{\epsilon}{2}, \frac{\epsilon}{2}\right]$ into a point in the physical volume V.

The Jacobian matrix of the transformation $\mathbf{S}$ for points on the mid-surface is derived from the covariant basis vectors and is defined as the 3×3 matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$. Additionally, the 3×3 matrix $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3]$ represents a rotation matrix, which transforms a vector in the global physical coordinate system (x_1, x_2, x_3) into a vector aligned with the local physical coordinate system (l_1, l_2, l_3).

The covariant vector basis of the mid-surface $\{\mathbf{a}_i\}$ can be extended to a covariant vector basis in $\mathbf{g}_i$ in each point $\boldsymbol{\xi}$ of the volume $\hat{V}$ as follows:

$$\mathbf{g}_\alpha = \hat{\partial}_\alpha \boldsymbol{\mathcal{V}} = \mathbf{a}_\alpha + \xi_3 \hat{\partial}_\alpha \mathbf{a}_3 \quad \text{and} \quad \mathbf{g}_3 = \hat{\partial}_3 \boldsymbol{\mathcal{V}} = \mathbf{a}_3 = \mathbf{n} \tag{4}$$

Note that the two covariant vector basis coincide for points of the the mid-surface, i.e. $\mathbf{a}_i = \mathbf{g}_i$ when $\xi_3 = 0$.

Furthermore, the covariant basis $\mathbf{g}_i$ naturally defines the contravariant basis, or dual basis, by $\mathbf{g}^j \cdot \mathbf{g}_i = \delta_i^j$, where δ_i^j is the Kronecker symbol. These basis define the covariant and contravariant components of the metric tensor, which are computed in a similar fashion by $g_{ij} = \mathbf{g}_i \cdot \mathbf{g}_j$ and $g^{ij} = \mathbf{g}^i \cdot \mathbf{g}^j$, respectively.

Moreover, the set of vectors tangent to $\mathcal{D}$ at any given point $\boldsymbol{x} \in \mathcal{D}$ span the tangent space $T_{\boldsymbol{x}}\mathcal{D}$:

$$T_{\boldsymbol{x}}\mathcal{D} = \{\mathbf{v} \in \mathbb{R}^3 : \mathbf{v} \cdot \mathbf{a}_3 = 0 \text{ at } \boldsymbol{x} \in \mathcal{D}\} \tag{5}$$

The set of *all* vectors tangent to the manifold $\mathcal{D}$, define the *tangent bundle*:

$$T(\mathcal{D}) = \bigcup_{\boldsymbol{x} \in \mathcal{D}} T_{\boldsymbol{x}}\mathcal{D} \tag{6}$$

Mechanics: The prior geometric description of the thick shell allows for the introduction of vector and tensor fields representing the mechanical variables. For instance, the displacement vector field $\mathbf{u} : \hat{V} \to \mathbb{R}^3$ can be expressed using its covariant (u_i) or contravariant (u^i) components as:

$$\mathbf{u} = u_i \mathbf{g}^i = u^i \mathbf{g}_i. \tag{7}$$

The small strain tensor, defined by the symmetrical part of the displacement gradient with respect to the physical coordinate system, can now be rewritten in terms of the curvature of the shell as:

$$\varepsilon_{ij}(\mathbf{u}) = \frac{1}{2}\left(\mathbf{g}_i \cdot \hat{\partial}_j \mathbf{u} + \mathbf{g}_j \cdot \hat{\partial}_i \mathbf{u}\right) \tag{8}$$

Remark ***1.*** Note that if the shell was defined by the map $\mathbf{S} = \mathrm{id}$, then the covariant basis $\{\mathbf{g}_i\}$ is equal to the canonical basis of $\mathbb{R}^3$ and one regains the classic formula $\varepsilon_{ij} = \frac{1}{2}(\hat{\partial}_i \mathbf{u}_j + \hat{\partial}_j \mathbf{u}_i)$.

In a similar fashion, the balance equation of stresses, expressed using the divergence operator will equally be written taking into account the curvature of the geometry.

Let us also consider a classical problem of linear elasticity under the assumption of small strains with using the generalized Hooke's law:

$$\sigma^{ij} = A^{ijrs}\varepsilon_{rs}, \tag{9}$$

where A^{ijrs} is the constitutive tensor for the given elastic, isotropic and homogeneous material. An additional hypothesis of this model is the plane stress state, i.e. $\sigma_{33} = 0$.

Under these assumptions, let us consider the shell to be composed of a linear isotropic elastic material with Young's modulus E and Poisson's ratio ν. Then, the constitutive relation can be expressed [18] in terms of the geometry as:

$$\begin{cases} \sigma^{\alpha\beta} = H^{\alpha\beta\lambda\mu}\varepsilon_{\lambda\mu}, & H^{\alpha\beta\lambda\mu} = \dfrac{E}{2(1+\nu)}\left(g^{\alpha\lambda}g^{\beta\mu} + g^{\alpha\mu}g^{\beta\lambda} + \dfrac{2\nu}{1-\nu}g^{\alpha\beta}g^{\lambda\mu}\right) \\ \sigma^{\alpha 3} = \dfrac{1}{2}G^{\alpha\lambda}\varepsilon_{\lambda 3}, & G^{\alpha\lambda} = \dfrac{2E}{1+\nu}g^{\alpha\lambda}. \end{cases} \tag{10}$$

Consequently, the virtual internal work of stress, depending on the displacement field $\mathbf{u}$ and the virtual displacement $\mathbf{v}$, is transformed by the preceding constitutive relation in equation (10). It is then expressed as the following bilinear form:

$$\int_V \sigma(\mathbf{u}) : \varepsilon(\mathbf{v})\,\mathrm{d}V = \int_V H^{\alpha\beta\lambda\mu}\varepsilon_{\lambda\mu}(\mathbf{u})\varepsilon_{\alpha\beta}(\mathbf{v}) + G^{\alpha\beta}\varepsilon_{\alpha 3}(\mathbf{u})\varepsilon_{\beta 3}(\mathbf{v})\,\mathrm{d}V \tag{11}$$

Let us now introduce the kinematic assumption of the Reissner-Mindlin shell model, wherein the displacement field $\mathbf{u}$ within the thick shell is characterized by the displacement and rotation of a point in the mid-surface:

$$\mathbf{u}(\xi_1, \xi_2, \xi_3) = \mathbf{u}_{\mathcal{D}}(\xi_1, \xi_2) + \xi_3 \mathbf{r}(\xi_1, \xi_2) \times \mathbf{n}(\xi_1, \xi_2) \tag{12}$$

where $\mathbf{u}_{\mathcal{D}} : \hat{\mathcal{D}} \to \mathbb{R}^3$ is the displacements field and $\mathbf{r} : \hat{\mathcal{D}} \to \mathbb{R}^3$ is the field of rotations at the corresponding point along the normal of the mid-surface $\mathcal{D}$.

This decomposition induces the following functional space of the admissible displacement functions:

$$H^1_D(\mathcal{D}) = \{\mathbf{v} = (\mathbf{v}^U, \mathbf{v}^R) \in H^1(\mathcal{D}) \times H^1(\mathcal{D}) : \quad \mathbf{v}^U|_{\partial\mathcal{D}^U_D} = \mathbf{0}, \quad \mathbf{v}^R|_{\partial\mathcal{D}^R_D} = \mathbf{0}\}, \tag{13}$$

where $\partial\mathcal{D}^U_D$ and $\partial\mathcal{D}^R_D$ represent two covering subsets of the Dirichlet boundary $\partial\mathcal{D}_D$ where zero displacement and rotation boundary conditions are enforced, respectively.

The variational problem of an elastic Reissner-Mindlin thick-shell is stated as:

Find the displacement field $\mathbf{u} \in H^1_D(\mathcal{D})$ such that:

$$\int_V \sigma(\mathbf{u}) : \varepsilon(\mathbf{v})\,\mathrm{d}V = \int_V \mathbf{f}^V \cdot \mathbf{v}\,\mathrm{d}V + \int_{\mathcal{D}} \mathbf{f}^S \cdot \mathbf{v}\,\mathrm{d}\mathcal{D} + \int_{\partial\mathcal{D}_N} \left(\mathbf{v}^U \cdot \mathbf{g} + \mathbf{v}^R \cdot \mathbf{m}\right) ds \tag{14}$$

for any test function $\mathbf{v} \in H^1_D(\mathcal{D})$.

In the preceding problem, $\mathbf{f}^V$ denotes a volumetric force, $\mathbf{f}^S$ denotes a surface tension applied only on the mid-section $\mathcal{D}$, $\mathbf{g}$ denotes the lateral traction applied to $\partial\mathcal{D}_N$ and $\mathbf{m}$ is the moment applied to $\partial\mathcal{D}_N$. It is important to note that, for the sake of simplicity, we have introduced some abuse of notation regarding the variable $\mathbf{u}$. In this context, $\mathbf{u}$ serves as both a displacement field, as defined in equation (12), and as a tuple of H^1 functions when used as an element of $H^1_D(\mathcal{D})$.

Multi-patch in IGA. The definitions of the concepts and the construction of the equations are at this point only defined for a single parametric shell, called patch. We now aim to extend these definitions to multi-patch systems, where the entire domain is split into a set of distinct patches.

Let us consider a collection of K distinct and non-overlapping NURBS parameterizations $\{\mathbf{S}_k\}_{k=1}^K$. Each individual patch is characterized by its mid-surface denoted as $\mathcal{D}_k$, which is obtained from $\mathcal{D}_k = \mathbf{S}_k(\hat{\mathcal{D}}_k)$. This collection $\{\mathcal{D}_k\}_{k=1}^K$ defines a partition of the mid-surface $\mathcal{D}$ with:

$$\overline{\mathcal{D}} = \bigcup_{k=1}^{K} \overline{\mathcal{D}}_k \quad \text{where} \quad \mathcal{D}_k \cap \mathcal{D}_l = \emptyset \quad \text{if} \quad k \neq l \tag{15}$$

The interface γ_{kl} between two patches is defined as the closure of the intersection of their boundaries:

$$\gamma_{kl} = \overline{\partial\mathcal{D}_k \cap \partial\mathcal{D}_l} \quad \text{with} \quad 1 \leq k < l \leq K \tag{16}$$

Moreover, let A_{kl} be the edge of the patch k that interfaces the patch l, as depicted in Figure 2. Then, a interface is said to be *geometrically conforming* when $\gamma_{kl} = A_{kl} = A_{lk}$. Similarly, the term *matching discretization* designates interfaces where the elements overlap in a one-to-one fashion with elements of the neighbour patch.

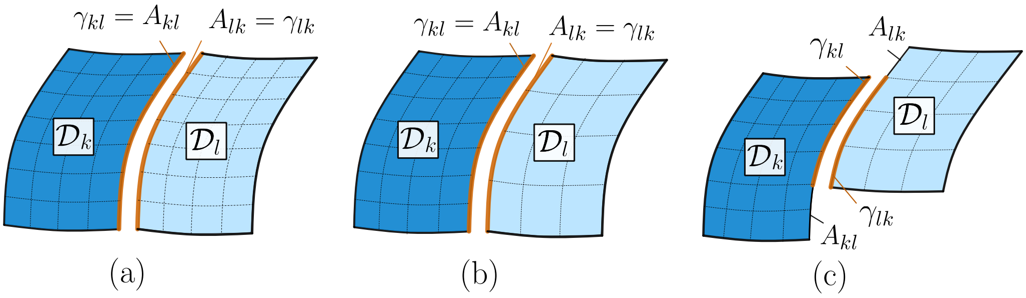

Figure 2: Illustration of three distinct simple multi-patch systems (for $K = 2$). In all cases, the patches k and l are images of the NURBS parameterization $\mathbf{S}_k$ and $\mathbf{S}_l$ and share a common interface $\gamma_{kl} = \overline{\partial\mathcal{D}_k \cap \partial\mathcal{D}_l}$. However, they are classified differently according their interface γ_{kl}: (a) geometrically conforming and matching discretization (b) geometrically conforming and non-matching discretization and (c) geometrically non-confomring and non-matching discretization.

By adding contributions of each individual patch, functions defined on patches can be naturally extended over the entire domain $\mathcal{D}$. For instance, consider a collection of functions $\{f_k\}_{k=1}^{K}$ where $f_k : \hat{\mathcal{D}}_k \to U$, with U representing a generic target space. In this context, it is possible to create a global function $f : \mathcal{D} \to U$ such that $f|_{\mathcal{D}_k} = f_k \circ \mathbf{S}_k^{-1}$ for $\forall k$. It is worth to observe that this principle applies to many of the mathematical entities defined earlier, such as $\mathbf{a}_i$, $\mathbf{g}_i$, $\mathbf{g}^i$, ε_{ij}, σ_{ij} and others. However, for the sake of clarity, we will refrain from using indices to reference patches in these quantities. Additionally, we will considered only continuous functions over the interfaces γ_{kl} between the patches.

3. Level set representation of the material domain

Modeling the shape within the shell using the level set method: Let us consider employing the level set technique to implicitly define a multi-patch shape $\Omega \subset \mathcal{D}$ using a level set function (LSF). For a specific patch k, let $\mathbf{S}_k$ denote its parameterization, resulting in a mid-surface $\mathcal{D}_k = \mathbf{S}_k(\hat{\mathcal{D}}_k)$. We introduce a level set function $\phi_k : \mathcal{D}_k \to \mathbb{R}$, which will be numerically modeled using NURBS. This representation enables precise delineation of shape boundaries and facilitates the generation of a NURBS output directly compatible with CAD software.

The *material phase*, i.e. the shape Ω_k, is as a subset of the mid-surface $\mathcal{D}_k$ defined as:

$$\Omega_k = \{\boldsymbol{x} \in \mathcal{D}_k : \phi_k(\boldsymbol{x}) < 0\} \tag{17}$$

Since the mid-surface $\mathcal{D}_k$ is parametric by construction, one can also define the *pull-back operator* of ϕ_k by $\mathbf{S}_k$ as the function $\hat{\phi}_k : \hat{\mathcal{D}}_k \to \mathbb{R}$ expressed by the following composition:

$$\hat{\phi}_k(\boldsymbol{\xi}) = \phi_k(\mathbf{S}_k(\boldsymbol{\xi})) \tag{18}$$

One can also introduce the *parametric material phase* $\hat{\Omega}_k$ as being the parametric counterpart of the shape Ω_k. By using the pullback operator, it is defined as:

$$\hat{\Omega}_k = \{\boldsymbol{\xi} \in \hat{\mathcal{D}}_k : \hat{\phi}_k(\boldsymbol{\xi}) < 0\} = \mathbf{S}_k^{-1}(\Omega_k) \tag{19}$$

Since the parameterization $\mathbf{S}_k$ is a diffeomorphism between $\hat{\mathcal{D}}_k$ and $\mathcal{D}_k$, it follows that for every $\boldsymbol{x} \in \mathcal{D}_k$:

$$\phi_k(\boldsymbol{x}) = \hat{\phi}_k(\mathbf{S}_k^{-1}(\boldsymbol{x})) \tag{20}$$

As a direct consequence, one can remark that the level set function ϕ_k is characterized solely by the pull-back operator $\hat{\phi}_k$. This concept is of utmost importance for the numerical modeling, as it highlights that a three-dimensional geometrical shape of the shell is completely defined by two parameters.

The general idea of the level set method is therefore to divide the physical domain into two subsets with distinct material properties. The *material phase* Ω_k has the material properties of the component that is being optimized, while the *void phase* $\mathcal{D}_k \setminus \Omega_k$ (also called *weak phase* or *ersatz material*) has weaker material properties, to simulate as if there is no material in $\mathcal{D}_k \setminus \Omega_k$. This whole setting can be summarized by the sketch on Figure 3. Moreover, this approach can equally be extended using n-NURBS surfaces to define 2^n material domains as in [19].

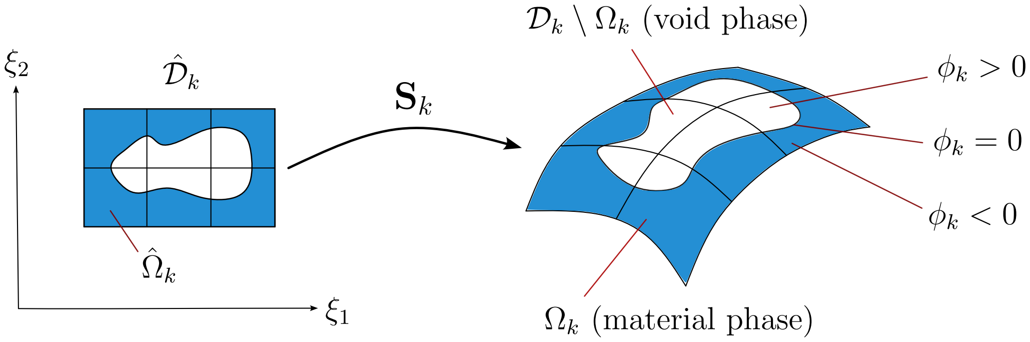

Figure 3: Schematic view of the level set representation in IGA. The material phase Ω_k (in blue) is defined as the set of points in the mid-surface $\mathcal{D}_k$ where the level set function ϕ_k is negative. The set $\mathcal{D}_k \setminus \Omega_k$ is called the *void phase* which, because of its weak material properties, acts as a void region. Similarly, the parametric material phase $\hat{\Omega}_k$ is the set of points in the parametric domain $\hat{\mathcal{D}}_k$ where $\hat{\phi}_k$ is negative.

During the optimization process, we consider that the shape Ω_k evolves as a function of pseudo-time $t > 0$ and can therefore be rewritten with a time dependency as $\phi_k(\boldsymbol{x}, t)$. A material point $\boldsymbol{x} \in \mathcal{D}$ will move according to a given velocity field $\boldsymbol{v}_k(\boldsymbol{x}, t)$, which will later be associated with the descent direction of a cost functional.

In this setting, the evolution of the boundaries of the domain Ω_k are described as:

$$\partial\Omega_k(t) = \{\boldsymbol{x} \in \mathcal{D}_k : \phi_k(\boldsymbol{x}(t), t) = 0\}. \tag{21}$$

Subsequently, for an arbitrary point $\boldsymbol{x}(t) \in \partial\Omega_k(t)$, the equation $\phi_k(\boldsymbol{x}(t), t) = 0$ holds. After differentiation with respect to the time t, one obtains the equation that governs the evolution of the shape Ω_k, namely the Hamilton-Jacobi equation:

$$\frac{\partial \phi_k}{\partial t} + \frac{\partial \boldsymbol{x}}{\partial t} \cdot \nabla \phi_k = \frac{\partial \phi_k}{\partial t} + \boldsymbol{v}_k \cdot \nabla \phi_k = 0. \tag{22}$$

It is worthwhile noting that these definitions are independent of the representation of the level set and are therefore seamlessly extended for multi-patch systems, as illustrated in Figure 4. By considering a collection of level set functions $\phi = \{\phi_k\}_{k=1}^K$, with $\phi_k : \mathcal{D}_k \to \mathbb{R}$, it follows that each ϕ_k defines a shape Ω_k on the patch k by equation 17. Consequently, the global shape Ω is straightforwardly realized as the union of these individual Ω_k:

$$\overline{\Omega} = \bigcup_{k=1}^{K} \overline{\Omega}_k \tag{23}$$

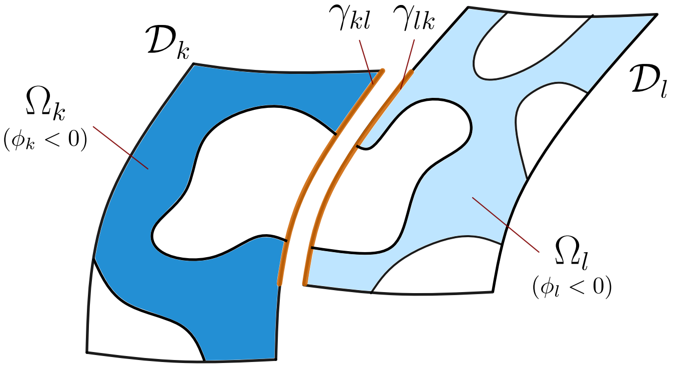

Figure 4: Multi-patch representation of a level set. By defining a set of level set functions $\{\phi_k\}_{k=1}^K$ one is able to define via equation (17) the set of shapes $\{\Omega_k\}_{k=1}^K$.

In numerical practice, an upwind scheme based on the Lax-Friedrichs flux [20, 21, 7] was employed to solve the Hamilton-Jacobi equation. The methodology involved solving the equation independently for each patch through the application of a finite difference scheme that incorporates the interfaces between the patches. The details of this approach are thoroughly elaborated in Appendix A.

3.1. Finding a CAD representation:

One notable advantage of using a topology optimization algorithm based on isogeometric analysis is the ability to extract domain boundaries as B-spline curves and optimal shapes as trimmed surfaces. This capability is advantageous for industrial applications, in particular for the automotive industry, as it enables the direct integration of optimized subsystem shapes with the global body-in-white assembly. In contrast, when employing classic topology optimization with FEM, an additional step is necessary to reconstruct the optimized CAD surfaces, as the output of the optimization process typically consists of a trimmed mesh.

The procedure to extract the domain B-splines starts by finding the set of parametric curves $\{(\hat{\Gamma}_k)_b\}_{b=1}^{B_k}$, also referred to as *p-curves*, that intercept the NURBS $\hat{\phi}_k$ with the plane $\xi_3 = 0$. Afterwards, we close the p-curves when both of their extremities intersect with the boundaries $\partial\hat{\mathcal{D}}_k$, as shown in Figure 5. Finally, the set $\{(\Gamma_k)_b\}_{b=1}^{B_k}$ of B-splines describing the boundary in the physical domain is obtained from the image of the parameterization $\mathbf{S}_k$ by the p-curves: $(\Gamma_k)_b = \mathbf{S}_k((\hat{\Gamma}_k)_b)$ for all b.

In numerical practice, the open-source C++ function `BRepAlgoAPI_Section` from OpenCascade [22] was used to compute the surface to plane intersection when computing the p-curves. Moreover, we remark that this set of closed curves precisely delineate the visible portion of the domain $\hat{\Omega}_k$, which can be used by a graphics display of a CAD system to render the trimmed surfaces, as elaborated in [23].

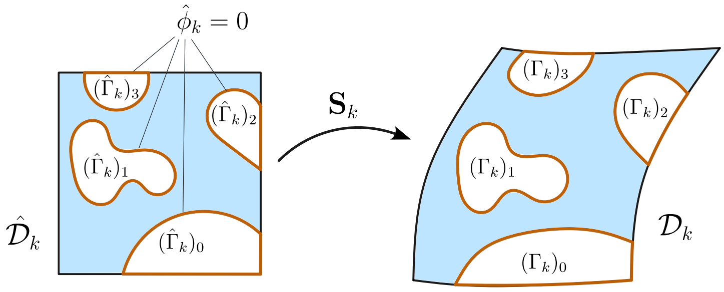

Figure 5: Extracting the domain boundaries as B-spline or trimmed surfaces. For a given patch k, we compute the set with $B_k \in \mathbb{N}$ closed parametric curves $\{(\hat{\Gamma}_k)_b\}_{b=1}^{B_k}$ where the level set $\hat{\phi}_k$ intersects the parametric plane $\xi_3 = 0$. We find then a corresponding B-spline representation in the physical domain, denoted $\{(\Gamma_k)_b\}_{b=1}^{B_k}$, by using the relation $(\Gamma_k)_b = \mathbf{S}_k((\hat{\Gamma}_k)_b)$ for all $b \in [\![1, B_k]\!]$.

4. IGA representation of the material level set

We are interested in solving the linear elasticity problem over a multi-patch shape Ω, described by a level set function $\phi = \{\phi_k\}_{k=1}^{K}$, using the Reissner-Mindlin shell of mid-surface $\mathcal{D}$. To achieve this, one has to model the material behavior of the two phases. Hence, let us consider a constitutive tensor C^{ijrs} for the strong phase and $\tilde{C}^{ijrs}$ for the weak phase. In each patch, the expected behavior of the material A^{ijrs} can be expressed in the domain $\mathcal{D}_k$ with a dependency on the level set function ϕ_k as:

$$A^{ijrs}(\phi_k) = \begin{cases} C^{ijrs} & \text{in } \Omega_k \qquad \textit{(strong phase, } \phi_k < 0\textit{)} \\ \tilde{C}^{ijrs} & \text{in } \mathcal{D}_k \setminus \Omega_k \qquad \textit{(weak phase, } \phi_k > 0\textit{)} \end{cases} \tag{24}$$

To avoid numerical complications, one can consider a smooth transition function to interpolate the properties of the material continuously from one phase to the other. Let $e > 0$ be a small parameter to represent the thickness of this interface between the two phases. One can introduce a smooth Heaviside function $\mathcal{H}_e \in C^\infty(\mathbb{R}, [0, 1])$, displayed on Figure 6, with the expression:

$$\mathcal{H}_e(\phi) := \frac{1}{2}\left(1 - \frac{\phi}{\sqrt{e^2 + \phi^2}}\right). \tag{25}$$

By defining an interpolation of constitutive tensors using the smooth transition in equation (25), one is able to express the stress tensor with a dependency on the level set function ϕ_k as such:

$$\sigma_{\phi_k}{}^{ij} = \underbrace{\left(\mathcal{H}_e(\phi_k)(C^{ijrs} - \tilde{C}^{ijrs}) + \tilde{C}^{ijrs}\right)}_{A^{ijrs}(\phi_k,\mathcal{H}_e)} \varepsilon_{rs}. \tag{26}$$

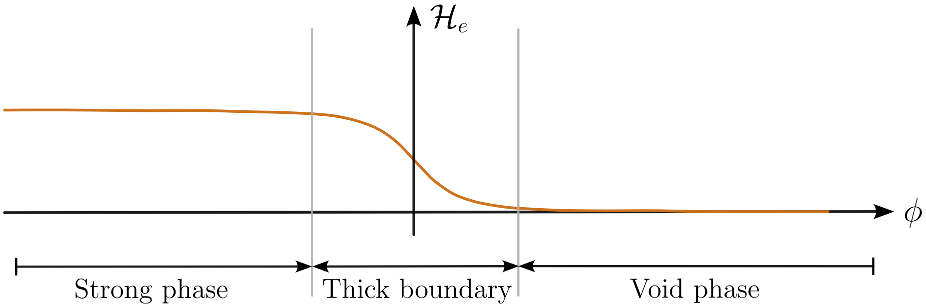

Figure 6: Smooth interface approach: plot of the heaviside function $\mathcal{H}_e$. This function provides a smooth continuous transition between the strong phase (negative values of ϕ) and the weak phase (positive values of ϕ). One can verify that equations (25) and (26) provide the desired effect by remarking that: $\lim_{e\to 0} A^{ijrs}(\phi,\mathcal{H}_e) = C^{ijrs}$ if $\phi < 0$ and $\tilde{C}^{ijrs}$ if $\phi > 0$.

Additionally, it is important to introduce the notation of the different regions in the boundaries. Let us consider the decomposition of the boundaries of $\mathcal{D}$ as $\partial\mathcal{D} = \partial\mathcal{D}_D \cup \partial\mathcal{D}_N \cup \partial\mathcal{D}_0$, where $\mathcal{D}_D$ and $\mathcal{D}_N$ are resp. the regions corresponding to Dirichlet and Neumann boundaries conditions and $\partial\mathcal{D}_0$ is a the traction-free region. Similarly, one can also decompose the boundaries $\partial\Omega$ of the shape Ω by using the same notation: $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N \cup \partial\Omega_0$.

With this setting, one can proceed towards the definition of the weak formulation of the linear elasticity. Let us consider the following Sobolev space, suited for the Reissner-Mindlin shell model:

$$H^1_D(\mathcal{D}_k) = \{\mathbf{v}_k = (\mathbf{v}^U_k, \mathbf{v}^R_k) \in H^1(\mathcal{D}_k) \times H^1(\mathcal{D}_k) : \mathbf{v}^U_k|_{\partial\mathcal{D}^U_D} = \mathbf{0}, \quad \mathbf{v}^R_k|_{\partial\mathcal{D}^R_D} = \mathbf{0}\}, \tag{27}$$

where $\partial\mathcal{D}^U_D$, $\partial\mathcal{D}^R_D$ are the subsets of the Dirichlet boundary $\partial\mathcal{D}_D$ wherein, respectively, zero displacements and rotations are imposed, and such that $\partial\mathcal{D}_D = \partial\mathcal{D}^U_D \cup \partial\mathcal{D}^R_D$.

Finally, one can mathematically recover the IGA shell model (see also [18]) now coupled with the level set formulation for the material distribution in the domain. Let us introduce the local (continuous and coercive) bilinear form $a_k : H^1_D(\mathcal{D}_k) \times H^1_D(\mathcal{D}_k) \to \mathbb{R}$ and a (continuous) linear form $L_k : H^1_D(\mathcal{D}_k) \to \mathbb{R}$ by:

$$a_k(\mathbf{u}_k, \mathbf{v}_k) = \int_{V_k} \sigma_{\phi_k}(\mathbf{u}_k) : \varepsilon(\mathbf{v}_k)\,\mathrm{d}V_k = \int_{[-\frac{\epsilon}{2},\frac{\epsilon}{2}]\times\hat{\mathcal{D}}_k} \sigma_{\phi_k}(\mathbf{u}_k) : \varepsilon(\mathbf{v}_k)|\det J_{\boldsymbol{\mathcal{V}}_k}|\,\mathrm{d}\hat{\mathcal{D}}_k\,\mathrm{d}\zeta \tag{28}$$

$$L_k(\mathbf{v}_k) = \int_{V_k} \left(\mathbf{f}^V_k \cdot \mathbf{v}_k\right) \mathbb{1}_{\Omega_k}\,\mathrm{d}V_k + \int_{\mathcal{D}_k} \left(\mathbf{f}^S_k \cdot \mathbf{v}_k\right) \mathbb{1}_{\Omega_k}\,\mathrm{d}\mathcal{D}_k + \int_{\partial\mathcal{D}_N} (\mathbf{v}^U_k \cdot \mathbf{g}_k + \mathbf{v}^R_k \cdot \mathbf{m}_k)\mathbb{1}_{\Omega_k} ds \tag{29}$$

where $\mathbf{f}_k^V$ is a volumetric force applied on the whole volume V_k, $\mathbf{f}_k^S$ is a surface force applied in the mid-section $\mathcal{D}_k$, $\mathbf{g}_k$ is the lateral forces applied to $\partial\mathcal{D}_N$, $\mathbf{m}_k$ is the moment applied to $\partial\mathcal{D}_N$, $J_{\boldsymbol{\mathcal{V}}_k}$ is the Jacobean matrix associated with transformation $\boldsymbol{\mathcal{V}}_k$ defined in equation (2) and $\mathbb{1}_B : \mathcal{D} \to \{0,1\}$ denotes the indicator function defined as $\mathbb{1}_B(x) = 1$ if $x \in B$ and $\mathbb{1}_B(x) = 0$ otherwise, for any set $B \subset \mathcal{D}_k$.

Consequently, to solve the linear elasticity in the context of multi-patch, we consider the penalty approach as formulated in [16]. The key concept of this approach is to introduce another a bilinear form $b_{kl} : H_D^1(\mathcal{D}_k) \times H_D^1(\mathcal{D}_l) \to \mathbb{R}$ that forces continuity between the interfaces of the patches k and l. One possible choice of for b_{kl} is the following:

$$b_{kl}(\mathbf{u}_k, \mathbf{v}_k) = \int_{\gamma_{kl}} (\mathrm{tr}_{kl}\mathbf{u}_k - \mathrm{tr}_{lk}\mathbf{u}_l)(\mathrm{tr}_{kl}\mathbf{v}_k - \mathrm{tr}_{lk}\mathbf{v}_l)\,\mathrm{d}\gamma_{kl} \tag{30}$$

where tr_{kl} denotes the boundary trace operator in the internal interface γ_{kl} and α_{kl} is a large penalty factor for the interface γ_{kl}.

The linear elasticity in multi-patch problem can then be solved by finding $\mathbf{u} = \{\mathbf{u}_k\} \in \prod_k H_D^1(\mathcal{D}_k)$:

$$\sum_k a_k(\mathbf{u}_k, \mathbf{v}_k) + \sum_{k>l} \alpha_{kl} b_{kl}(\mathbf{u}_k, \mathbf{v}_k) = \sum_k L_k(\mathbf{v}_k) \quad \forall \mathbf{v} = \{\mathbf{v}_k\} \in \prod_k H_D^1(\mathcal{D}_k) \tag{31}$$

As $a_k(\mathbf{u}_k, \mathbf{v}_k) = L_k(\mathbf{v}_k)$ is valid for all patches k, the term $\sum_{k>l} \alpha_{kl} b_{kl}(\mathbf{u}_k, \mathbf{v}_k)$, which represents the sum over all pairwise combinations of internal interfaces between the patches, has to be equal to zero. Furthermore, as a consequence of the definition of b_{kl}, the continuity of $\mathbf{u}_k$, $\mathbf{u}_l$ and $\mathbf{v}_k$, $\mathbf{v}_l$ at the interface γ_{kl} is guaranteed.

The computation of the boundary integral in b_{kl}, however trivial for matching discretizations, raises some technical difficulties that deserve close attention in the case of non-conforming interfaces. The issue lays on the fact that the boundary elements in one patch are not coincident to those of the neighbour patch, as depicted in Figure 7. To overcome this issue, a knot-to-segment approach was considered, as in [16], to build a common knot vector between the interfaces.

For that let us introduce a projection operator $P_{kl} : \hat{\gamma}_{kl} \to \hat{\gamma}_{lk}$ defined by:

$$P_{kl}(\boldsymbol{\xi}) = \underset{\boldsymbol{\zeta} \in \hat{\gamma}_{lk}}{\arg\min} \, |\mathbf{S}_k(\boldsymbol{\xi}) - \mathbf{S}_l(\boldsymbol{\zeta})| \tag{32}$$

where $\hat{\gamma}_{kl}$ and $\hat{\gamma}_{lk}$ are the parametric counterparts of the interface γ_{kl} in patches k and l, namely $\hat{\gamma}_{kl} = \mathbf{S}_k^{-1}(\gamma_{kl})$ and $\hat{\gamma}_{lk} = \mathbf{S}_l^{-1}(\gamma_{lk})$.

This projection operator allows us to find, for any given parametric coordinate $\boldsymbol{\xi}$ of patch k laying in the interface $\hat{\gamma}_{kl}$, the corresponding parametric position $\boldsymbol{\zeta}$ in patch l such that $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ are mapped to the same point in the boundary, i.e. $\mathbf{S}_l(\boldsymbol{\zeta})$ is coincident to $\mathbf{S}_k(\boldsymbol{\xi})$. Additionally, let $\Xi^{(kl)}$ represent the knot vector in patch k associated with interface γ_{kl}. One can construct a *common knot vector* $\Xi_C^{(kl)}$ for patch k by augmenting the original vector $\Xi^{(kl)}$ with the projection of the knot vector $\Xi^{(lk)}$ of the neighboring patch l onto patch k. Mathematically, this is expressed by:

$$\Xi_C^{(kl)} = \Xi^{(kl)} \cup P_{lk}\left(\Xi^{(lk)}\right) \tag{33}$$

It is worth noting that this construction provides a sequence of conforming elements that can be used for boundary integration similarly to the matching discretization case. This can be verified by the equality of the sets:

$$\mathbf{S}_k\left(\Xi_C^{(kl)}\right)=\mathbf{S}_l\left(\Xi_C^{(lk)}\right) \tag{34}$$

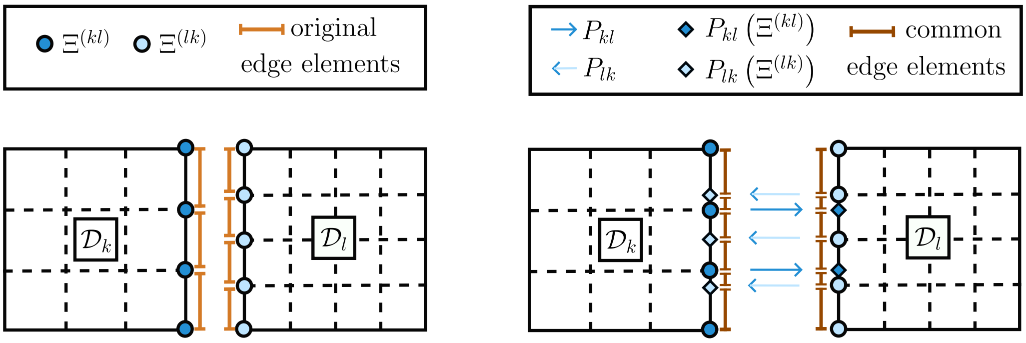

Figure 7: A basic example of a multi-patch system with non-matching discretization. On the left, one can observe that the elements at the non-matching interface do not overlap. On the right, the projection of the knots onto the neighboring patch enables the construction of a common knot vector, resulting in elements overlapping in a manner suited for numerical integration.

5. Shape optimization problem

We start by introducing the cost function $\mathcal{J}:\mathcal{P}(\mathcal{D})\to\mathbb{R}$ as in equation (35), where $\mathcal{P}(\mathcal{D})$ is the set of all subsets of $\mathcal{D}$. This function evaluates the performance of a certain multi-patch shape Ω, as a composition of the compliance and the volume. For such proposes, we consider a fixed Lagrange multiplier $\Lambda>0$ to penalize volume increases during the optimization. This is a classic technique, as used in [2, 24, 25, 21], to attempt to simultaneously increase the stiffness of the shape Ω and to reduce its volume. Therefore, the chosen value for the Lagrange multiplier Λ can be used to give more or less importance to one or other of these terms.

$$\mathcal{J}(\Omega)=\sum_k\underbrace{\int_{V_k}\sigma_{\phi_k}(\mathbf{u}_k):\varepsilon(\mathbf{u}_k)\,\mathrm{d}V_k}_{\text{compliance}}+\Lambda\sum_k\underbrace{\int_{V_k}\mathbb{1}_{\Omega_k}\,\mathrm{d}V_k}_{\text{volume}} \tag{35}$$

where $\mathbf{u}=\{\mathbf{u}_k\}\in\prod_k H^1_D(\mathcal{D}_k)$ is the solution of linear elasticity problem (31). Let us further remark that both the compliance term and the field $\mathbf{u}$ itself are dependent on the shape Ω because of the level set functions $\{\phi_k\}$.

Let us now consider set the of admissible shapes, defined as:

$$\mathcal{U}_{ad}=\{\Omega\subset\mathcal{D}:\partial\Omega_N=\partial\mathcal{D}_N\text{ and }\partial\Omega_D\subset\partial\mathcal{D}_D\} \tag{36}$$

Then, the optimization can then simply be summarized by finding the optimum shape $\Omega^*\in\mathcal{U}_{ad}$ such that:

$$\mathcal{J}(\Omega^*) = \inf_{\Omega \in \mathcal{U}_{ad}} \mathcal{J}(\Omega), \quad \text{subject to (31)} \tag{37}$$

As described in [2], the shape derivative of $\mathcal{J}$, we consider a transformation of a smooth initial shape $\Omega^{(0)}$ using a small perturbation field $\boldsymbol{\theta} \in W^{1,\infty}(\mathcal{D}, \mathbb{R}^d)$, as in:

$$T_{\boldsymbol{\theta}} = (\mathrm{id} + \boldsymbol{\theta}) \tag{38}$$

This transformation allows us to define the shape derivative as the Frechet derivative in $W^{1,\infty}(\mathcal{D}, \mathbb{R}^d)$ at 0 of the application $\boldsymbol{\theta} \to \mathcal{J}(T_{\boldsymbol{\theta}}(\Omega))$:

$$\mathcal{J}(T_{\boldsymbol{\theta}}(\Omega)) = \mathcal{J}(\Omega) + \mathcal{J}'(\Omega)(\boldsymbol{\theta}) + \mathcal{O}(\boldsymbol{\theta}) \quad \text{with } \lim_{\boldsymbol{\theta} \to \mathbf{0}} \frac{|\mathcal{O}(\boldsymbol{\theta})|}{\|\boldsymbol{\theta}\|} = 0 \tag{39}$$

Regularization field on the tangent bundle: since our goal is to find the optimal material distribution on the mid-surface $\mathcal{D}$, we propose to build a velocity field that has values on the tangent bundle of the manifold $\mathcal{D}$. If we transform a shape $\Omega^{(0)}$ with the transformation $T_{\boldsymbol{\theta}}$ (equation (38)), we would like to ensure that $T_{\boldsymbol{\theta}}(\Omega^{(0)})$ is still a subset of the optimization domain $\mathcal{D}$.

With this condition in mind, let us consider a H^1 Sobolev space that satisfy the following conditions:

$$\begin{aligned} \Theta(\mathcal{D}_k) = \Bigg\{ \boldsymbol{\vartheta}_k \in (H^1(\mathcal{D}_k))^3 : \quad & \boldsymbol{\vartheta}_k \cdot \mathbf{a}^3 = 0 \text{ in } \mathcal{D}_k \text{ and} \\ & \boldsymbol{\vartheta}_k \cdot \mathbf{t}_N = 0 \text{ on } \partial\mathcal{D}_k \setminus \bigcup_{\substack{l=1 \\ l \neq k}}^{K} \gamma_{kl} \Bigg\} \end{aligned} \tag{40}$$

where $\mathbf{t}_N \in T(\mathcal{D}_k)$ is the unit outward normal to $\partial\mathcal{D}_k$ in the tangent space and $\mathbf{a}^i$ denotes the contra-variant basis in the middle surfaces, obtained by $\mathbf{a}_i \cdot \mathbf{a}^j = \delta_i^j$. We remark that the field $\mathbf{t}_N$ can be identified with the contra-variant basis depending on the edges $\{A_1, A_2, A_3, A_4\}$ of the patch in the physical space, as illustrated in Figure 8.

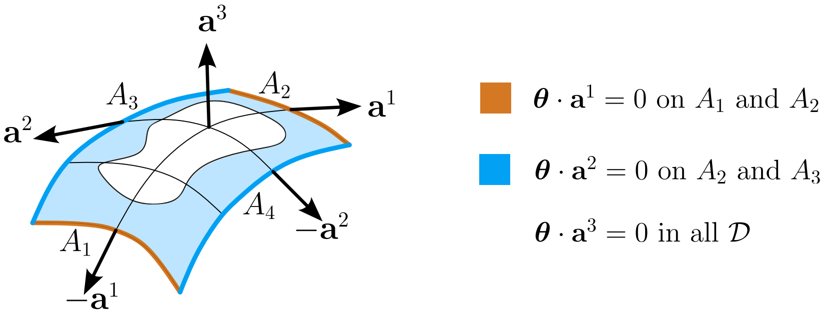

Figure 8: Example of a single patch shell illustrating how to build the function space $\Theta(\mathcal{D})$. One can also visualize how to determine the outward unit normal $\mathbf{t}_N$ tangent to the manifold $\mathcal{D}$ based on the contra-variant basis $\mathbf{a}^i$.

Furthermore, one could also observe that the function space satisfies the conditions that we desire. Because we aim to keep the thickness of the shell constant, the condition $\boldsymbol{\vartheta}_k \cdot \mathbf{a}^3 = 0$ imposes that a shape $\Omega_k^{(0)}$ transformed with $T_{\boldsymbol{\theta}_k}$ has no variation on the direction of the thickness, for any $\boldsymbol{\theta}_k \in \Theta(\mathcal{D})$. Similarly, one also observes that a material point $\boldsymbol{x} \in \partial\mathcal{D}_k$ will not move outwards from the domain because of the condition $\boldsymbol{\vartheta}_k \cdot \mathbf{t}_N = 0$.

Let us now introduce the shape derivative of the cost functional in equation (35). Typically, in the context of compliance, the shape derivative is expressed as a boundary integral, as demonstrated in works such as in [2, 24]. However, for our approach, we have chosen an alternative formulation, provided by [21], which offers a simpler implementation using a volume integral. In this study, we adapt this formulation to the context of multi-patch isogeometric analysis. Specifically, for $\boldsymbol{\vartheta} = \{\boldsymbol{\vartheta}_k\} \in \prod_k \Theta(\mathcal{D}_k)$, we consider the shape derivative expression as follows:

$$\mathcal{J}'(\Omega)(\boldsymbol{\vartheta}) = \sum_k \int_{\mathcal{D}_k} \big(2\nabla\mathbf{u}_k^T \sigma_{\phi_k}(\mathbf{u}_k) - \sigma_{\phi_k}(\mathbf{u}_k) : e(\mathbf{u}_k)\mathbf{I}\big) : \nabla\boldsymbol{\vartheta}_k + \Lambda \sum_k \int_{\Omega_k} \mathrm{div}\boldsymbol{\vartheta}_k, \quad (41)$$

where $\mathbf{I}$ is the second order identity tensor.

From the definition of the functional space $\Theta(\mathcal{D}_k)$ of the velocity fields, one can now propose its relation to the shape derivative of $\mathcal{J}$. This is achieved by defining a regularization problem in the form of an identification problem using a scalar product over $\Theta(\mathcal{D}_k)$. For clarity purposes, let us first enunciate this formulation for the single-patch case and then we genelarize it to the multi-patch approach.

In the **single-patch case**, the regularization problem could be enunciated by the following variational problem:

$$\textit{Find} \quad \boldsymbol{\theta} \in \Theta(\mathcal{D}) \quad \textit{s.t.} \quad \int_V \nabla_{\mathcal{D}}\boldsymbol{\theta} : \nabla_{\mathcal{D}}\boldsymbol{\vartheta} = -\mathcal{J}'(\Omega)(\boldsymbol{\vartheta}) \quad \forall \boldsymbol{\vartheta} \in \Theta(\mathcal{D}) \quad (42)$$

This approach can easily be generalized to a **multi-patch case**, similarly to what has been proposed for the linear elasticity case, by means of the same penalty approach. Therefore, the regularization problem aims to find the field $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k\} \in \prod_k \Theta(\mathcal{D}_k)$ such that

$$\sum_k \int_{V_k} \nabla_{\mathcal{D}}\boldsymbol{\theta}_k : \nabla_{\mathcal{D}}\boldsymbol{\vartheta}_k + \sum_{k>l} \beta_{kl} b_{kl}(\boldsymbol{\theta}_k, \boldsymbol{\vartheta}_k) = -\sum_k \mathcal{J}'(\Omega_k)(\boldsymbol{\vartheta}_k) \quad \forall \boldsymbol{\vartheta} = \{\boldsymbol{\vartheta}_k\} \in \prod_k \Theta(\mathcal{D}_k) \quad (43)$$

where β_{kl} is a penalty factor for the interface γ_{kl}.

We remark that for the multi-patch case, a given regularization field $\boldsymbol{\theta} \in \prod_k \Theta(\mathcal{D}_k)$ has the condition $\mathbf{t}_N \cdot \boldsymbol{\theta}_k = 0$ on the external boundary for $\forall k$. In Figure 9, we illustrate on a multi-patch system how to apply the boundary conditions on the boundaries of the patches according to their nature (external or internal interface).

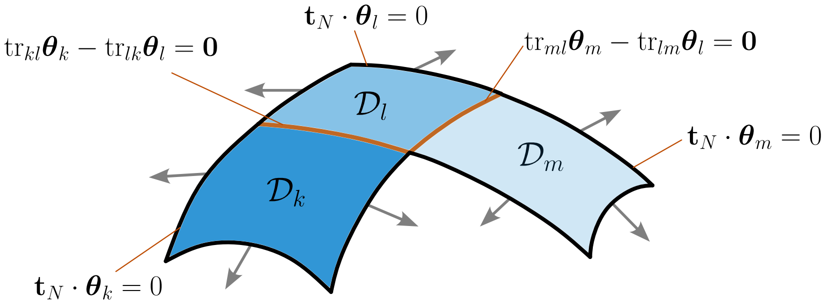

Figure 9: Illustration of the boundary conditions for the regularization problem in a simple multi-patch case. The external boundaries of $\mathcal{D}$ have the orthogonality condition $\boldsymbol{\theta}_k \cdot \mathbf{t}_N = 0$ for $\forall k$. The internal interfaces are handled with the penalty approach, where the aim is to guarantee the continuity of $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_l$ on the interface γ_{kl} for all pairs (k, l).

It is also important to remark that the left-hand side of the equation (43) does not change during the iterations. By denoting $\mathbf{L}$ the matrix associated with this term, it is possible to use the same factorization of $\mathbf{L}$ in all iteration, thereby considerably reducing computational cost.

As for the space discretion of the problem in (43), we consider a discretization of the space $\Theta(\mathcal{D}_k)$ by NURBS basis functions. This set $\Theta^h(\mathcal{D}_k) \subset \Theta(\mathcal{D}_k)$ can be defined as follows:

$$\Theta^h(\mathcal{D}_k) = \left\{ \boldsymbol{\vartheta} = \hat{\boldsymbol{\vartheta}} \circ \mathbf{S}_k^{-1} : \hat{\boldsymbol{\vartheta}} \in \bigotimes_{i \in [\![0,2]\!]} \mathcal{R}(p_{k_i}, \Xi_{k_i}) \right\} \tag{44}$$

where $\mathcal{R}(p_{k_i}, \Xi_{k_i})$ is the set of rational B-splines function defined on the patch k of degree p_{k_i} and with the knot vector Ξ_{k_i} in the direction $i \in \{0, 1, 2\}$.

The methodology detailed in this study can be conveniently encapsulated in the form of an algorithm, as presented in (1). The technical details on this implementation is discussed next in Section 6. Moreover, a guideline on how to solve the regularization problem in the tangent bundle using isogeometric analysis is explained in more depth in Appendix B. Finally, we also remark that this algorithm uses the notation introduced in Appendix A, where the numerical schemes for solving the Hamilton-Jacobi equation are presented. In this appendix, we introduce $[\phi_k]$ as the discrete counterpart of the level set function ϕ_k, representing the evaluation of the NURBS ϕ_k at the grid points $[\hat{G}_k]$, defined by a uniform discretization of the parameter domain.

Algorithm 1 Topology optimization with IGA shells and LSM

```
Input:
  Multi-patch geometry using a set of NURBS parametrizations $\{\mathbf{S}_k\}$
  Discretization grid $[\hat{G}_k]$ of the parametric domain $\hat{\mathcal{D}}_k$
  Initial level set discretizations $[\phi_k]^{(0)}$
  Fixed Lagrange multiplier $\Lambda > 0$

Compute:
Calculate and factorize the matrix $\mathbf{L}$ of the regularization problem (43)
for $m \geq 0$ iterate until convergence do
    Compute the NURBS $\{\phi_k\}^{(m)}$ by interpolating $[\phi_k]^{(m)}$ on the parametric grid $[\hat{G}_k]$
    Solve linear elasticity on the shell (31) using $\{\phi_k\}^{(m)}$
    Compute cost function $\mathcal{J}^{(m)}$
    if $\mathcal{J}^{(m)} < \mathcal{J}^{(m+1)}$ or $m == 0$ then
        Compute shape derivative $\mathcal{J}'$ using (41)
        Find $\{\boldsymbol{\theta}_k\}^{(m)}$ using $\mathbf{L}$ by updating right-hand term of (43) with $\mathcal{J}'$
    else
        Decrease $\Delta t$ of Hamilton-Jacobi
        Set $\{\boldsymbol{\theta}_k\}^{(m)} \leftarrow \{\boldsymbol{\theta}_k\}^{(m-1)}$
    end if
    Find $[\phi_k]^{(m+1)}$ by solving evolution equation (22) using $\{\boldsymbol{\theta}_k\}^{(m)}$, $\{\phi_k\}^{(m)}$ and $\Delta t$
end for
```

6. Numerical results

In this section, we present the results achieved through the application of the described optimization method. Numerically, the characteristics of the weaker phase were established by scaling the properties of the stronger phase by a small positive factor. Typically, the Young's modulus of the stronger phase can be multiplied by 10^{-3} while maintaining the same Poisson ratio. Throughout all the examples conducted in this study, we consider non-dimensionalized problems, with a Poisson ratio of $\nu = 0.3$, a Young's modulus of $E = 1$, penalty parameters set to $\alpha_{kl} = 10^3$ and $\beta_{kl} = 10^5$ and all concentrated forces have magnitude of one.

To fulfill the constraint $\partial\Omega_N = \partial\mathcal{D}_N$ specified in equation (36), an additional stiffness is introduced into the Laplace matrix $\mathbf{L}$, affecting the elements in a small neighbourhood of the Neumann boundary condition region. Typically, the original stiffness of these elements can be multiplied by a factor of 10^3. This technique effectively compels the velocity field $\boldsymbol{\theta}$ to approach zero within $\partial\Omega_N$, thereby forcing this region to remain unchanged while the shape Ω is evolving.

During the resolution of the IGA problems, the level set functions were modeled as NURBS. Following the resolution of the Hamilton-Jacobi equation for each patch k, which yields a matrix on the parametric grid (see Appendix A), the NURBS representation is subsequently updated through a process of bivariate interpolation [26] using the newly acquired values of ϕ on the grid. The degree of the NURBS surfaces was set to 3 across all problems. Additionally, the size of the parametric grid was chosen to be twice as refined as the knot vector employed for linear elasticity and the regularization problem.

Numerous resources were used to increase performance, given that the primary codebase is entirely implemented in Python. For accelerated spline evaluation, an implementation based on [27] was employed. Furthermore, for the construction of the stiffness matrix, a LIL matrix object was developed utilizing the *numba* package [28]. In addition, the package PyPardiso [29] played a pivotal role in the efficient resolution of linear systems and LU factorization processes.

6.1. Example 1: parabolic domain

The first problem considers two concentrated forces of direction $-\mathbf{e}_3$ applied on a single-patch domain obtained by the extrusion of a parabola (see left panel in Figure 10). The geometry was discretized with 140 elements along the $\mathbf{e}_2$ direction and with 414 elements in its perpendicular direction, ensuring a nearly uniform distribution of element sizes in the physical space. Additionally, the Lagrange multiplier was configured to $\Lambda = 6 \cdot 10^3$.

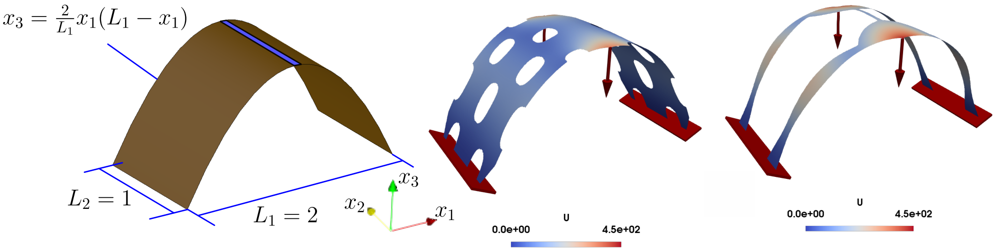

Figure 10: Example 1 - parabolic shell with two concentrated forces. From left to right: the parabolic optimization domain, the initial deformed shape with the loads applied and the final optimal deformed solution found after 186 iterations. The blue zone on the left geometry represents the stiffness region for the regularization problem. The colors on both deformed configurations are set using the same scale and correspond to the magnitude of the displacement field.

The evolution of the objective function, the compliance, the volume, as well as the shape at iterations 4, 18 and 33, are displayed in Figure 11. One can discern that, following 33 iterations, the shape begins to exhibit a keen resemblance to the final form achieved at iteration 186.

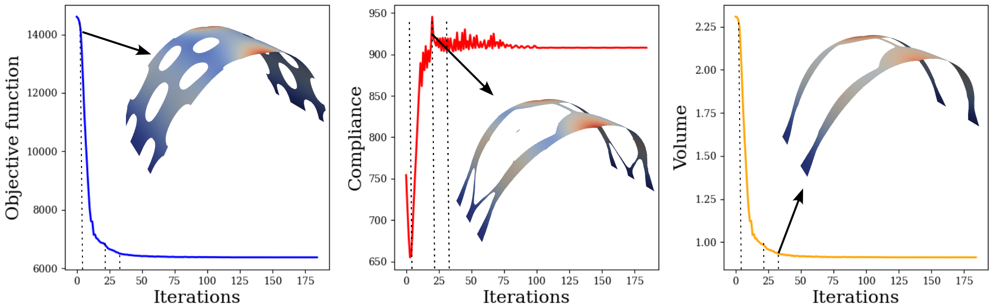

Figure 11: Example 1 - parabolic shell with two concentrated forces. The evolution of the objective function, the compliance and the volume with the increasing number of iterations shows the convergence of the optimization process. The evolution of the shape is displayed at iterations 4, 18 and 33.

6.2. *Example 2: conforming platform 3D* 1×1

The second example is dedicated to study a three dimensional platform consisting of three flat panels connected with two curved patches, as illustrated in Figure 12. As for the IGA discretization, the length L_1 was partitioned into 180 elements, whereas the element size in the perpendicular direction was chosen to maintain an uniform element size in physical space. This results in a total of 78120 elements for all patches. Furthermore, the thickness of all shells was fixed at $\epsilon = 0.05$ and the volume Lagrange multiplier was set to a constant value of $\Lambda = 8 \cdot 10^4$.

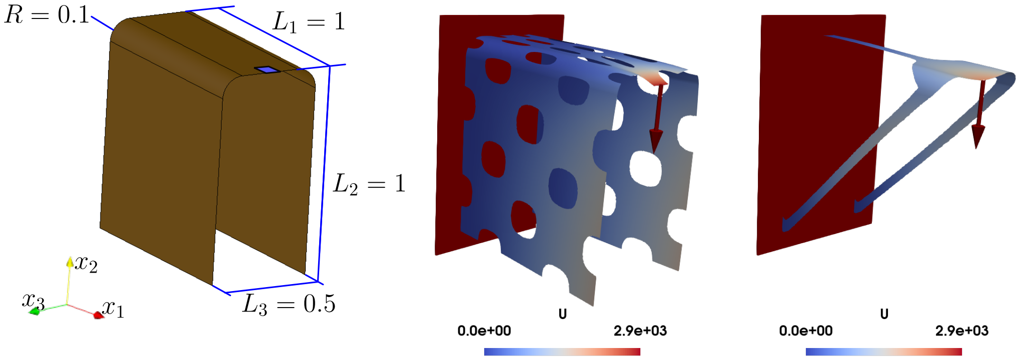

Figure 12: Example 2 - conforming platform 3D 1×1. From left to right: the optimization domain, the initial deformed shape with the applied load and the final optimal deformed solution found after 190 iterations. The blue zone on the left geometry represents the stiffness region for the regularization problem. The colors on both deformed configurations are set using the same scale and correspond to the magnitude of the displacement field.

It is possible to visualize the convergence of the objective function, the compliance and the volume in Figure 13. Moreover, the evolution of the shape is shown in three different stages of the optimization, at iterations 6, 17 and 89. The sudden peaks in compliance, triggering a peak in the cost function, occur because the volume shape derivative causes the shape Ω to be such that $\partial\Omega_N$ and $\partial\Omega_D$ are disconnected. This is a consequence of a modification in the descent algorithm. More precisely, we tolerate increases of the cost function after a maximum of three attempts to decrease the time step Δt. Despite this issue, the shape still manages to evolve into a connected domain. One can also observe the presence of disconnected regions on the shape Ω at iteration 17, that eventually disappear before iteration 89. Furthermore, one can remark that after iteration 89, the shape closely approximates the final state achieved by the conclusion of the optimization process.

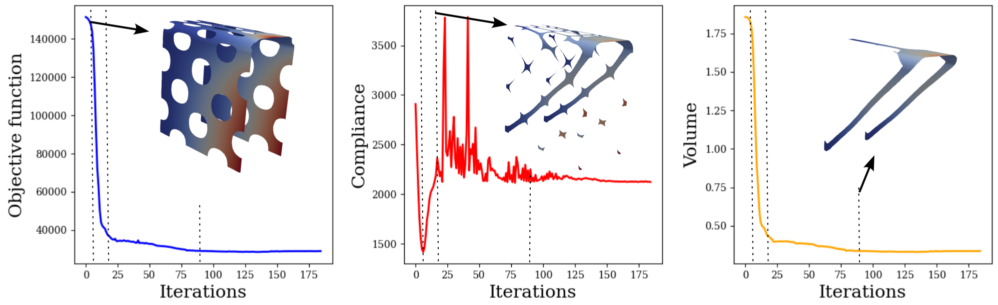

Figure 13: Example 2 - conforming platform 3D 1×1. The evolution of the objective function, the compliance and the volume with the increasing number of iterations shows the convergence of the optimization process. The evolution of the shape is displayed at iterations 6, 17 and 89.

6.3. *Example 3 - conforming platform 3D* 2×1

The following example, as depicted in Figure 14, explores a multi-patch optimization domain that is closely linked to the previous one. The primary deviation from the preceding geometry is the adjustment of the length L_1 to 2, along with an alteration in the thickness of all panels to $\epsilon = 0.1$. For the IGA discretization, the length L_1 was divide into 180 elements and the discretization in the orthogonal direction was selected to ensure uniform element sizes in physical space, which yielded a total of 39060 elements for all patches. Finally, a volume Lagrange multiplier of $\Lambda = 1.9 \cdot 10^3$ was incorporated in the analysis.

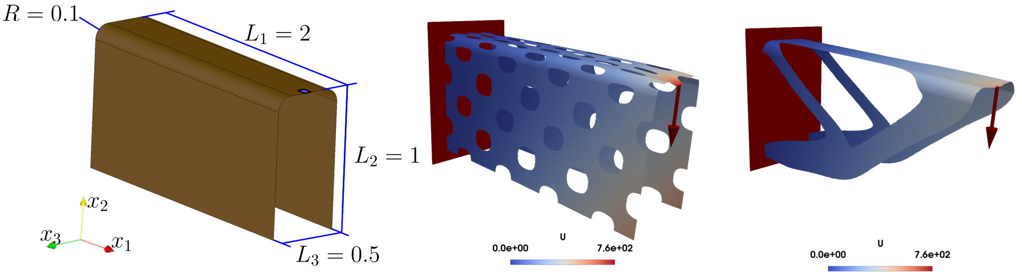

Figure 14: Example 3 - conforming platform 3D 2×1. From left to right: the optimization domain, the initial deformed shape with the applied load and the final optimal deformed solution found after 176 iterations. The blue zone on the left geometry represents the stiffness region for the regularization problem. The colors on both deformed configurations are set using the same scale and correspond to the magnitude of the displacement field.

The convergence of the objective function, compliance, and volume, along with the evolving shape at iterations 6, 17, and 54, is illustrated in Figure 15. Similar to the previous example, one can observe the existence of disconnected regions in iteration 17, but they vanish before reaching iteration 54. By iteration 54, the shape closely approximates the final optimal configuration.

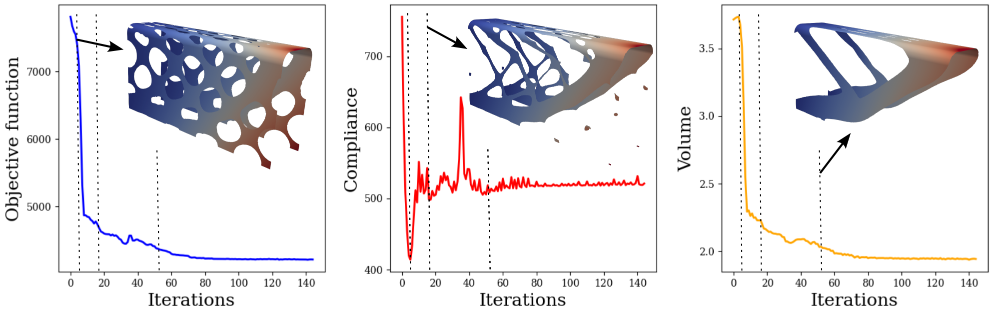

Figure 15: Example 3 - conforming platform 3D 2 × 1. The evolution of the objective function, the compliance and the volume with the increasing number of iterations shows the convergence of the optimization process. The evolution of the shape is displayed at iterations 6, 17 and 54.

6.4. Example 4: conforming cantilever 3D

In the subsequent problem, we address a multi-patch domain as depicted in Figure 16. This configuration comprises eight patches, each with a thickness of $\epsilon = 0.1$. For spatial discretization for the IGA problems, the element size in the parametric domain is chosen to ensure that elements in the physical domain are uniformly distributed. Under this criterion, the direction $\mathbf{e}_1$ is discretized with uniformity using 220 elements, resulting in a total of 69080 elements. Additionally, the volume Lagrange multiplier is set to $\Lambda = 2.5 \cdot 10^3$.

This particular example draws inspiration from a well-known study case in topology optimization, the cantilever beam. Typically applied in the context of 2D linear elasticity, the results for this example can be found in [2, 21]. Our method manages to replicate these results when applied to a rectangular Reissner-Mindlin plate.

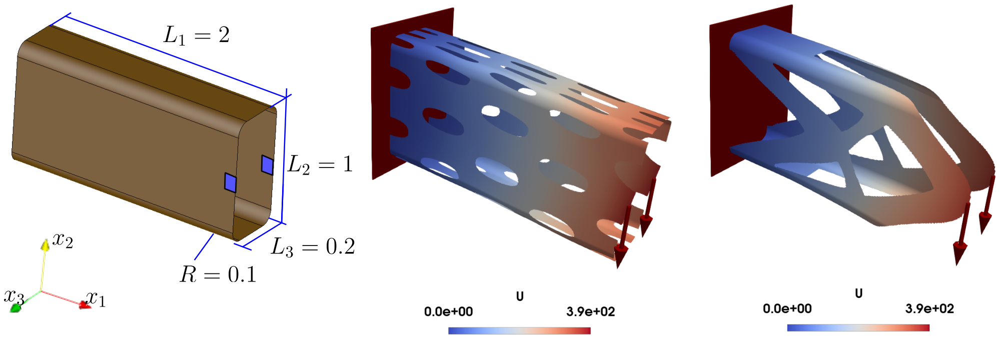

Figure 16: Example 4 - conforming cantilever 3D. From left to right: the optimization domain, the initial deformed shape with the applied loads and the final optimal deformed solution found after 255 iterations. The blue zones on the left geometry represent the stiffness region for the regularization problem. The colors on both deformed configurations are set using the same scale and correspond to the magnitude of the displacement field.

In Figure 17, we present the progressive evolution of the optimization process as the number of iterations increases. We provide visualizations of the metrics of our interest, namely the objective function, the compliance and the volume. Moreover, the transformation of the shape is depicted at three distinct stages of the optimization, specifically at iterations 7, 21 and 55. By the 55th iteration, it becomes clear that the shape closely resembles the converged configuration.

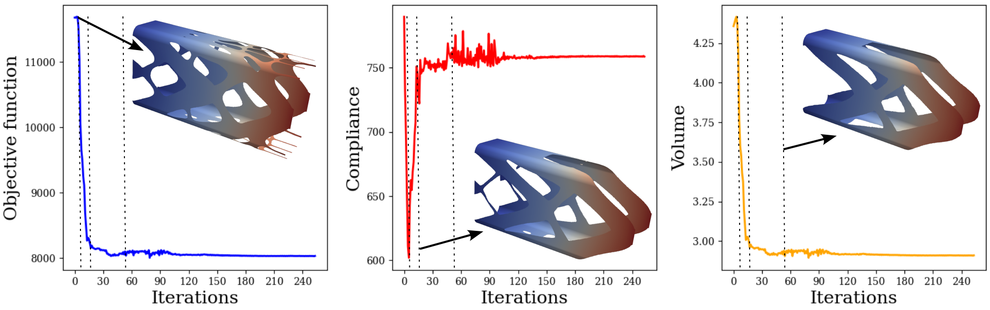

Figure 17: Example 4 - conforming cantilever 3D. The evolution of the objective function, the compliance and the volume with the increasing number of iterations shows the convergence of the optimization process. The evolution of the shape is displayed at iterations 7, 21 and 55.

6.5. Example 5: conforming arc 3D

The subsequent problem considers a multi-patch domain forming a three-dimensional arc structure (see left panel in Figure 18). The thickness is set to $\epsilon = 0.1$ for all patches. The element sizes in the parametric domain are chosen to ensure a near-uniform distribution of elements in the physical. For this purpose, the linear segment at the top of the arc, with size $R_e - R_i - 2R$, was discretize with 70 elements, which resulted in a total of 36080 elements for all patches. For this optimization process, a volume Lagrange multiplier of $\Lambda = 3.0 \cdot 10^3$ was employed.

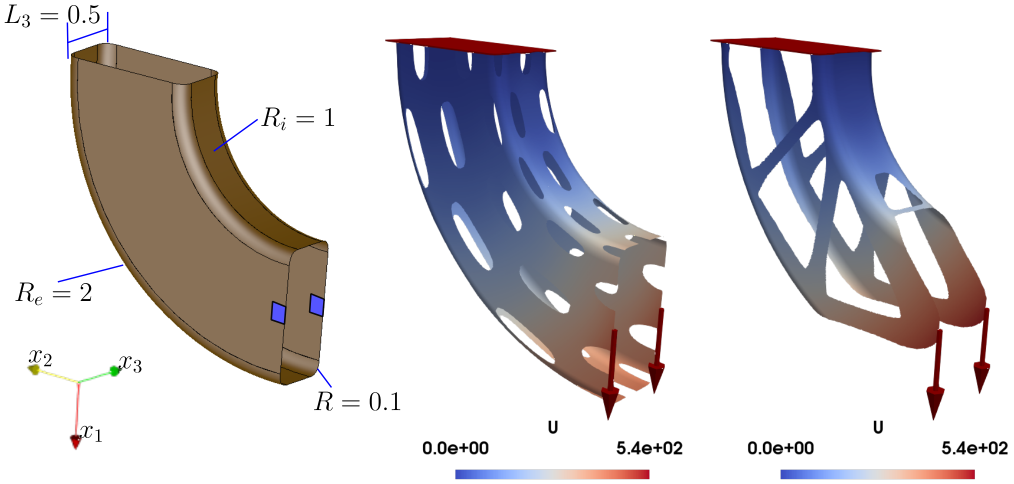

Figure 18: Example 5 - conforming arc 3D. From left to right: the optimization domain, the initial deformed shape with the applied loads and the final optimal deformed solution found after 135 iterations. The blue zones on the left geometry represent the stiffness region for the regularization problem. The colors on both deformed configurations are set using the same scale and correspond to the magnitude of the displacement field.

The progression of the optimization process is illustrated in Figure 18, where one can observe the convergence of the objective function, the compliance and the volume. Furthermore, three snapshots of the evolving shape are provided at iterations 5, 14, and 42, offering insights into the transformation of the shape over time.

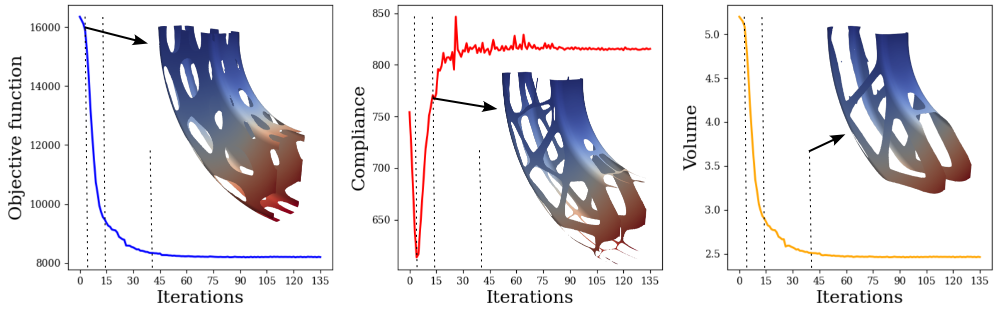

Figure 19: Example 5 - conforming arc 3D. The evolution of the objective function, the compliance and the volume with the increasing number of iterations shows the convergence of the optimization process. The evolution of the shape is displayed at iterations 5, 14 and 42.

6.6. Example 6: torsion of a conforming cylinder

The next problem under consideration involves a multi-patch representation of a cylindrical structure subjected to torsional loading. This representation utilizes four patches, as visually depicted in Figure 20. The geometry was discretized by partitioning the cylinder with 120 elements in the $\mathbf{e}_3$ direction. A commensurate refinement was employed in the perpendicular direction to guarantee uniform element sizing in the physical domain.

This discretization strategy yielded a total of 22560 elements across all patches. The thickness of the shells was uniformly set at $\epsilon = 0.1$ and the Lagrange multiplier was established at $\Lambda = 4.5 \cdot 10^3$

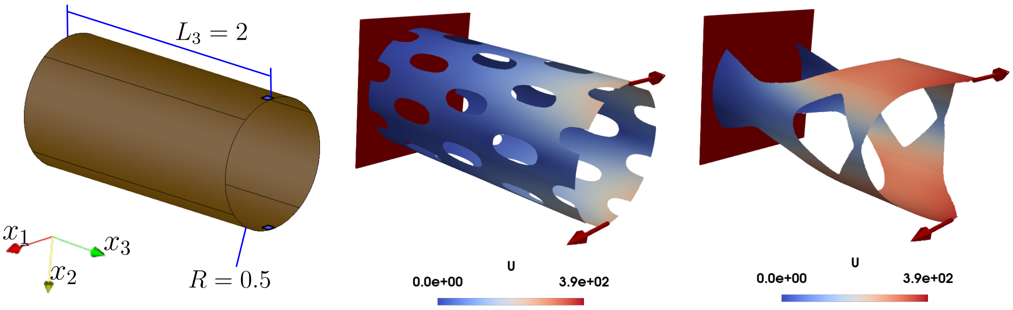

Figure 20: Example 6 - torsion of a conforming cylinder. From left to right: the optimization domain, the initial deformed shape with the applied loads and the final optimal deformed solution found after 187 iterations. The blue zones on the left geometry represent the stiffness region for the regularization problem. The colors on both deformed configurations are set using the same scale and correspond to the magnitude of the displacement field.

In Figure 21, one can visualize the evolution of the algorithm through the plots of the objective function, the compliance and the volume, as they vary with the increasing number of iterations. This figure highlights three pivotal stages of this evolution, observed at iterations 6, 22, and 71. One can remark that by iteration 71 one obtains a similar shape to that of the final optimum shape achieved at iteration 187.

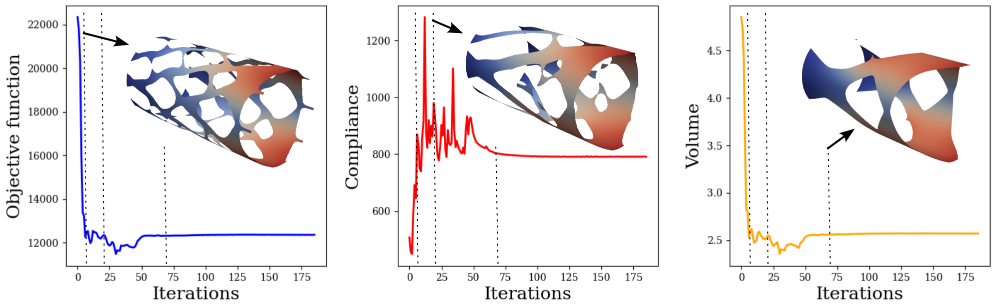

Figure 21: Example 6 - torsion of a conforming cylinder. The evolution of the objective function, the compliance and the volume with the increasing number of iterations shows the convergence of the optimization process. The evolution of the shape is displayed at iterations 6, 22 and 71

6.7. *Example 7: non-conforming platform 3D*

This following example aims to demonstrate the capability of this approach in managing geometrically non-conforming parameterizations and non-matching distributions. For that, we propose a structural arrangement akin to that of examples 2 and 3, while incorporating two non-conforming interfaces, as depicted in Figure 22. Regarding the discretization of the IGA problems, the length L_1 was divided into 130 elements, with the element size in the perpendicular direction uniformly distributed based on the discretization of L_1. This amounts to a total of 25610 elements for all patches. Moreover, the shell thickness was set at $\epsilon = 0.05$ across all patches and a value of $\Lambda = 2.0 \cdot 10^5$ was chosen for the volume Lagrange multiplier.

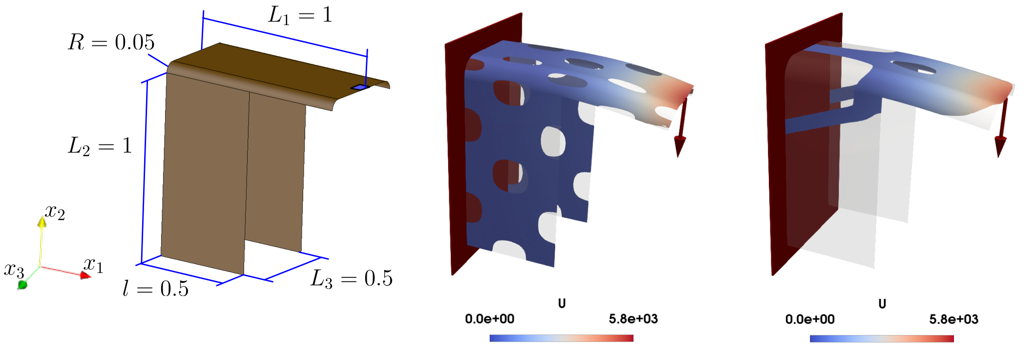

Figure 22: Example 7 - non-conforming platform 3D. From left to right: the optimization domain, the initial deformed shape with the applied loads and the final optimal deformed solution found after 101 iterations. The blue zone on the left geometry represents the stiffness region for the regularization problem. The colors on both deformed configurations are set using the same scale and correspond to the magnitude of the displacement field.

In Figure 23, the evolution of the shapes in the course of the iterations are shown by means of the plots of the objective function, the compliance and the volume. This illustration facilitates an observation of how the proposed method yields a continuous level set function at the interfaces of the patches, as evidenced at iterations 3, 9, and 45.

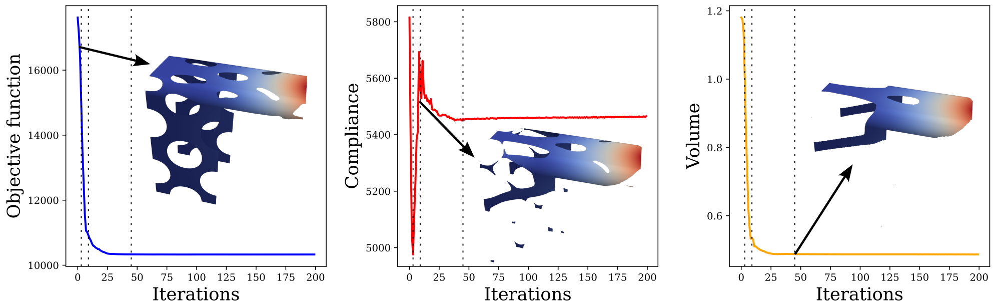

Figure 23: Example 7 - non-conforming platform 3D. The evolution of the objective function, the compliance and the volume with the increasing number of iterations shows the convergence of the optimization process. The evolution of the shape is displayed at iterations 3, 9 and 45

6.8. Example 8: torsion of a non-conforming cylinder

The final example under examination seeks to demonstrate the well-functioning of the approach in predominantly curved geometries with non-conforming and non-matching interfaces. Let us consider a multi-patch structure of a non-conforming cylindrical structure subjected to a torsion load, as depicted in Figure 24. The geometry underwent discretization by dividing the cylinder into 130 elements along the $\mathbf{e}_3$ direction. A corresponding refinement was applied perpendicular to ensure consistent element sizes in the physical domain. This discretization approach resulted in a combined total of 27040 elements for all patches. The shell thickness remained uniformly set at $\epsilon = 0.1$, while the Lagrange multiplier was fixed at $\Lambda = 4.5 \cdot 10^3$.

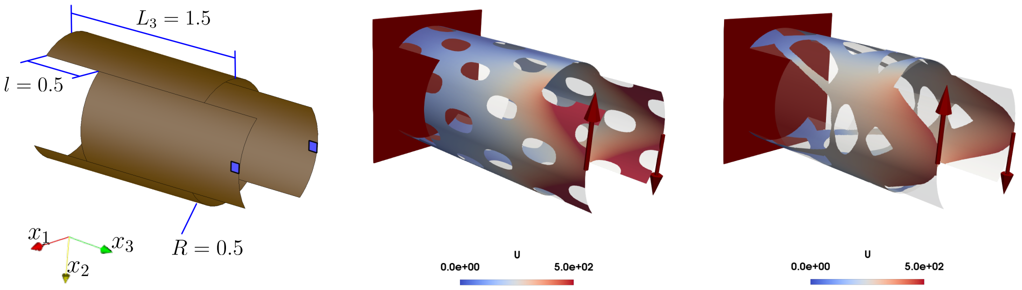

Figure 24: Example 8 - torsion of a non-conforming cylinder. From left to right: the optimization domain, the initial deformed shape with the applied loads and the final optimal deformed solution found after 181 iterations. The blue zones on the left geometry represent the stiffness region for the regularization problem. The colors on both deformed configurations are set using the same scale and correspond to the magnitude of the displacement field.

Insights into the progression of the algorithm via plots are provided by Figure 25, which illustrates the changes in the objective function, compliance, and volume over successive iterations. Similarly to the previous example, it becomes apparent how the proposed method maintains a continuous level set function at the non-conforming interfaces of the patches, exemplified at iterations 3, 22, and 154.

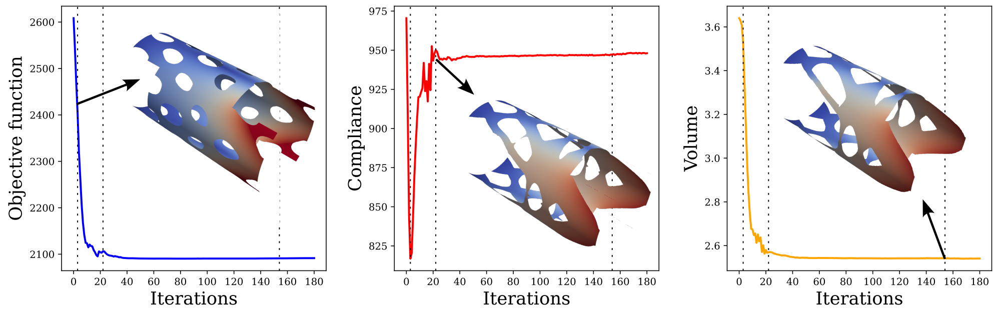

Figure 25: Example 8 - torsion of a non-conforming cylinder. The evolution of the objective function, the compliance and the volume with the increasing number of iterations shows the convergence of the optimization process. The evolution of the shape is displayed at iterations 3, 22 and 154

6.9. Extracting the solution as a CAD

As detailed in Section 3.1, the method subject of this study facilitates the extraction of domain boundaries in the form of CAD curves and trimmed surfaces. In Figure 26, we present an application of the procedure proposed for the structure in Example 8. One can visualize the level set function in two different patches, as well as the smooth B-spline domain lines that can be exported in regular CAD format such as IGES, STEP, and STL.

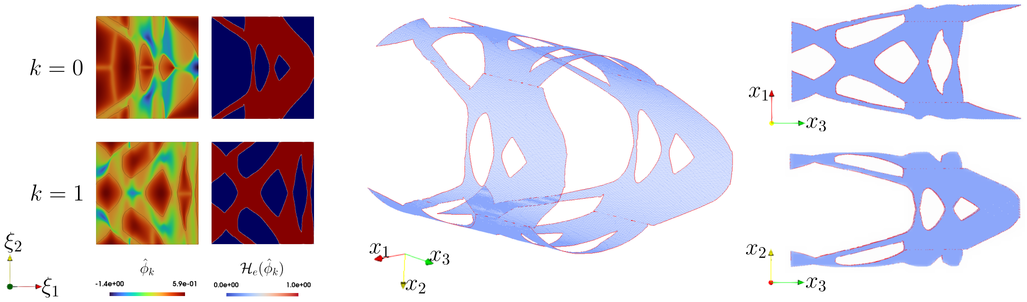

Figure 26: Process of extracting a smooth B-spline curve from the results of the optimization in Example 8 - Torsion of a non-conforming cylinder. On the left, one can visualize the pullback operator of the level set function $\hat{\phi}_k$ along with its composition with the Heaviside function $\mathcal{H}_e$. These functions are only plotted on the parameter domain of patches $k = 0, 1$, since the other two patches yield identical results, as the structure is symmetrical. On the right, the B-spline curves obtained from the CAD extraction procedure. It consists of computing the intersection of $\hat{\phi}_k$ with the parametric plane $\xi_3 = 0$ using the C++ function *BRepAlgoAPI_Section* from OpenCascade [22]. The resulting p-curves are mapped to the physical domain via the NURBS parameterization map $\mathbf{S}_k$.

7. Concluding remarks and discussion

This study has successfully implemented a topology optimization method that leverages the strengths of isogeometric analysis and Reissner-Mindlin thick shells while maintaining the inherent flexibility of the level set method. The use of NURBS-modeled level set functions enhances the precision of numerical integration. The resulting smooth boundary of the optimized shapes aligns perfectly with industrial requirements, as it can be exported as a CAD model.

One limitation of the proposed method is its assumption of G^1 continuity across the interfaces of the patches comprising the manifold $\mathcal{D}$. This assumption poses a challenge when dealing with structures that feature sharp corners. However, as our work demonstrates, this issue can be addressed in certain cases through the incorporation of fillets. Additionally, while the Hamilton-Jacobi equation was solved using finite differences, the full advantage of ϕ being modeled as a NURBS was not exploited. Exploring alternative schemes that directly incorporate the derivatives calculated from the NURBS could potentially improve computational efficiency and accuracy.

Looking ahead, we anticipate that this approach can be readily extended to encompass other industrially relevant cost functions. For instance, it could be applied to maximizing the smallest eigen-frequency, as shown in [11], target displacements/behavior, as in [2, 7], and minimization of the norm of the Von Mises stress, as investigated in [5].

Acknowledgements

The authors gratefully acknowledge the financial and technical collaboration with STELLANTIS, under ANRT CIFRE contract number 2021THESE0409/01636_22_00019.

Appendix A. Numerical scheme for Hamilton Jacobi equation in non conforming multi-patch

In this section, we present the numerical scheme to solve the Hamilton-Jacobi equation in a non-conforming multi-patch geometry. This scheme builds upon prior works [20, 21], and extends them to accommodate the complexities inherent in multi-patch parametric surfaces. To this end, let us assume that a velocity field $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k\}_{k=1}^K$ is known and that it is characterized by being a descent direction of a given cost functional $\mathcal{J}$. We recall that the establishment of this velocity field and its relation with the shape derivative of the cost-functional $\mathcal{J}$ are discussed in Section 5.

Let us revisit the Hamilton-Jacobi equation (22) on a given patch k, considering now $\{\boldsymbol{\theta}_k\}_{k=1}^K$ as the velocity field. As for the boundary conditions, the external interfaces are set with a zero flux condition (Neumann), whereas the internal interfaces between the patches must satisfy a continuity condition. Hence, the problem to be solve on each patch k is the following:

$$\begin{cases} \dfrac{\partial \phi_k}{\partial t} + \boldsymbol{\theta}_k \cdot \nabla_{\mathcal{D}} \phi_k = 0 & \text{in } \mathcal{D}_k \times \mathbb{R}_+ \\ \phi_k(\cdot, 0) = \phi_k^0 & \text{in } \mathcal{D}_k \\ \nabla_{\mathcal{D}} \phi_k \cdot \mathbf{t}_N = 0 & \text{on } \partial\mathcal{D}_k \setminus \cup_{l=1, l\neq k}^K \gamma_{kl} \\ \mathrm{tr}_{kl}\phi_k - \mathrm{tr}_{lk}\phi_l = 0 & \text{on } \cup_{l=1, l\neq k}^K \gamma_{kl} \end{cases} \tag{A.1}$$

where $(\nabla_{\mathcal{D}} \phi_k)_\alpha = \frac{\partial \phi_k}{\partial l_\alpha} \mathbf{t}_\alpha$ is the gradient operator written in the local coordinates system in the physical domain and ϕ_k^0 is a known NURBS initialization function of the level set at the patch k. As we operate within the tangent bundle, it is worth noting that the fields $\boldsymbol{\theta}_k$ have a vanishing normal component: $\boldsymbol{\theta}_k \cdot \mathbf{t}_3 = 0$.

The overall strategy to solve problem (A.1) consists of finding a discrete representation of the Hamilton-Jacobi equation, which involves the application of a finite differences scheme and subsequently solving it independently for each patch. The scheme incorporates information from neighboring patches, ensuring the continuity condition in equation (A.1) across the internal interfaces.

Initially, the parameter domain is discretized using a uniform grid $\hat{G}_k \subset \hat{\mathcal{D}}_k$, consisting of N_k^α points in each direction $\alpha \in \{1, 2\}$ and thus defining the steps $\Delta\xi_k^\alpha = \frac{1}{N_k^\alpha - 1}$. The components of the grid can then be computed as follows:

$$[\hat{G}_k]_{pq} = (p\Delta\xi_k^1, q\Delta\xi_k^2) \quad \forall (p, q) \in \prod_{\alpha=1}^{2} [\![0, N_k^\alpha - 1]\!]. \tag{A.2}$$

Note that this parametric grid induces a grid $[G_k]_{pq}$ in the physical space via the parametrization $\mathbf{S}_k$. Its components are computed for all $(p, q) \in \prod_{\alpha=1}^{2} [\![0, N_k^\alpha - 1]\!]$ with:

$$[G_k]_{pq} = \mathbf{S}_k([\hat{G}_k]_{pq}) \tag{A.3}$$

The operators in equation A.1 are also discretized uniformly in time by choosing a time step Δt that satisfies CFL conditions. To meet this condition, the Courant number C can be evaluated with $C = \Delta t \max_k (\sum_{\alpha=1}^{2} \max(|\boldsymbol{\theta}_k \cdot \mathbf{t}_\alpha|)/\Delta l_\alpha)$, where the steps Δl_α are the minimal distance between two consecutive points of the physical grid $[G_k]$ in the direction α. In numerical practice, no numerical instabilities were encountered when using the condition $C \leq C_{\max} = 1$.

Furthermore, we address the discretization of the level set function in both spatial and temporal domains. For this purpose, the evaluations of the function ϕ at the grid points will be denoted as follows:

$$[\phi_k]^n_{pq} = \hat{\phi}_k([\hat{G}_k]_{pq}, n\Delta t) \tag{A.4}$$

In sight of this considerations, we propose in equation A.5 a discrete representation of the Hamilton-Jacobi in equation A.1, which is based on the Lax-Friedrichs flux [20, 21]. By isolating the term $[\phi_k]^{n+1}_{pq}$, one obtains a recursive formula which can be straightforwardly solved for $n > 0$, starting from the initialization $[\phi_k]^0_{pq}$.

$$\begin{aligned} \frac{[\phi_k]^{n+1}_{pq} - [\phi_k]^n_{pq}}{\Delta t} + \sum_{\alpha=1}^{2} \Bigg(& \frac{[\boldsymbol{\theta}_k]_{pq} \cdot \mathbf{t}_\alpha}{2} \left(\mathrm{d}^+_{t_\alpha}[\phi_k]^n_{pq} + \mathrm{d}^-_{t_\alpha}[\phi_k]^n_{pq}\right) \\ & - \frac{|[\boldsymbol{\theta}_k]_{pq} \cdot \mathbf{t}_\alpha|}{2} \left(\mathrm{d}^+_{t_\alpha}[\phi_k]^n_{pq} - \mathrm{d}^-_{t_\alpha}[\phi_k]^n_{pq}\right) \Bigg) = 0 \end{aligned} \tag{A.5}$$

where $(p,q) \in \prod_{\alpha=1}^{2} [\![0, N^\alpha_k - 1]\!]$ are the indices of the spatial discretization, $n \in \mathbb{N}^*$ is the indice of the time discretization, $\alpha \in \{1,2\}$ is the indices of each direction in local coordinates, $[\phi_k]^n_{pq}$ is the evaluation of the level set function in the grid points, $[\boldsymbol{\theta}_k]_{pq} \cdot \mathbf{t}_\alpha$ is α-th component of the evaluation of the velocity field in the grid points, and $\mathrm{d}^\pm_{t_\alpha}$ are the forward (+) and backward (-) Euler finite difference scheme for computing the derivative in the local direction $\mathbf{t}_\alpha$.

The computation of the approximation of the derivatives with respect to the physical domain coordinates in equation (A.5) is achieved by a coordinate transformation of the finite differences computed with respect to the parametric domain:

$$\begin{bmatrix} \mathrm{d}^\pm_{t_1}[\phi_k]^n_{pq} \\ \mathrm{d}^\pm_{t_2}[\phi_k]^n_{pq} \\ 0 \end{bmatrix} = [\mathbf{T}_k]_{pq}[\mathbf{A}_k]_{pq} \begin{bmatrix} \mathrm{d}^\pm_{\xi_1}[\phi_k]^n_{pq} \\ \mathrm{d}^\pm_{\xi_2}[\phi_k]^n_{pq} \\ 0 \end{bmatrix} \tag{A.6}$$

where $\mathrm{d}^\pm_{\xi_m}$ are the forward (+) and backward (-) Euler finite difference scheme for computing the derivative with respect to the parametric coordinate ξ_α and the 3×3 matrices $[\mathbf{A}_k]_{pq}$, $[\mathbf{T}_k]_{pq}$ are the evaluation at the grid points of the change of basis matrices $\mathbf{A}, \mathbf{T}$ presented in Section 2.

Let us now introduce the computation of the finite differences $\mathrm{d}^\pm_{\xi_1}[\phi_k]^n_{pq}$ and $\mathrm{d}^\pm_{\xi_2}[\phi_k]^n_{pq}$ in equation (A.6). We remark firstly that, in order to simplify the notation of the schemes, we assume that the local reference system of the adjacent patches is aligned with that of patch k, which can easily be achieved by a simple transposition operation.

Then, the **backward Euler schemes** for the derivatives with respect to the parametric coordinates (ξ_1, ξ_2) are computed with:

$$\mathrm{d}^-_{\xi_1}[\phi_k]^n_{pq} = \begin{cases} \dfrac{[\phi_k]^n_{pq} - [\phi_k]^n_{p-1q}}{\Delta\xi^1_k} & \text{if } p \in [\![1, N^1_k - 1]\!], \\[2ex] \dfrac{[\phi_k]^n_{0q} - \hat{\phi}_l\left(P_{kl}\left([\hat{G}_k]_{0q}\right) - \begin{bmatrix}\Delta\xi^1_l \\ 0\end{bmatrix}\right)}{\Delta\xi^1_l} & \text{if } p = 0 \text{ and } \exists l \neq k \\ & \text{such that } [G_k]_{0q} \in \partial\mathcal{D}_l, \\[2ex] \dfrac{[\phi_k]^n_{1q} - [\phi_k]^n_{0q}}{\Delta\xi^1_k} & \text{otherwise.} \end{cases} \tag{A.7}$$

$$\mathrm{d}^-_{\xi_2}[\phi_k]^n_{pq} = \begin{cases} \dfrac{[\phi_k]^n_{pq} - [\phi_k]^n_{pq-1}}{\Delta\xi^2_k} & \text{if } q \in [\![1, N^2_k - 1]\!] \\[2ex] \dfrac{[\phi_k]^n_{p0} - \hat{\phi}_l\left(P_{kl}\left([\hat{G}_k]_{p0}\right) - \begin{bmatrix}0 \\ \Delta\xi^2_l\end{bmatrix}\right)}{\Delta\xi^2_l} & \text{if } q = 0 \text{ and } \exists l \neq k \\ & \text{such that } [G_k]_{p0} \in \partial\mathcal{D}_l, \\[2ex] \dfrac{[\phi_k]^n_{p1} - [\phi_k]^n_{p0}}{\Delta\xi^2_k} & \text{otherwise.} \end{cases} \tag{A.8}$$

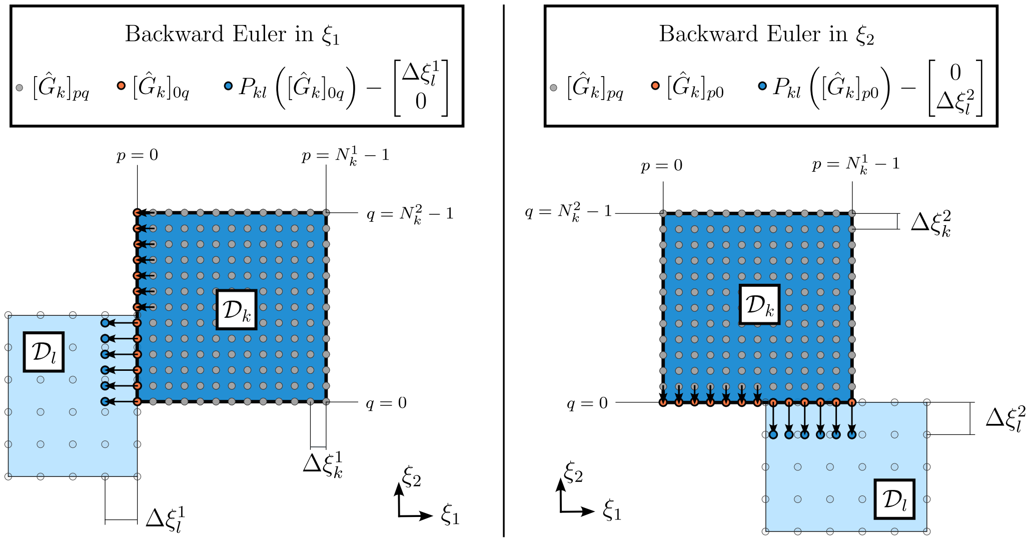

Figure A.27: Illustration on how to compute the backward Euler finite differences for the derivatives with respect to the parametric coordinates (ξ_1, ξ_2). As the scheme is trivial in the interior of the domain, we depict only how to compute the last two cases in equations A.7 and A.8.

Similarly, the **forward Euler schemes** for the derivatives with respect to the parametric coordinates (ξ_1, ξ_2) are obtained with the following:

$$\mathrm{d}^+_{\xi_1}[\phi_k]^n_{pq} = \begin{cases} \dfrac{[\phi_k]^n_{p+1q} - [\phi_k]^n_{pq}}{\Delta\xi_k^1} & \text{if } p \in [\![0, N_k^1 - 2]\!] \\[2ex] \dfrac{\hat{\phi}_l\left(P_{kl}\left([\hat{G}_k]_{N_k^1-1q}\right) + \begin{bmatrix}\Delta\xi_l^1 \\ 0\end{bmatrix}\right) - [\phi_k]^n_{N_k^1-1q}}{\Delta\xi_l^1} & \text{if } p = N_k^1 - 1 \text{ and } \exists l \neq k \\ & \text{s.t. } [\hat{G}_k]_{N_k^1-1q} \in \partial\mathcal{D}_l, \\[2ex] \dfrac{[\phi_k]^n_{N_k^1-1q} - [\phi_k]^n_{N_k^1-2q}}{\Delta\xi_k^1} & \text{otherwise.} \end{cases} \tag{A.9}$$

$$\mathrm{d}^+_{\xi_2}[\phi_k]^n_{pq} = \begin{cases} \dfrac{[\phi_k]^n_{pq+1} - [\phi_k]^n_{pq}}{\Delta\xi_k^2} & \text{if } q \in [\![0, N_k^2 - 2]\!] \\[2ex] \dfrac{\hat{\phi}_l\left(P_{kl}\left([\hat{G}_k]_{pN_k^2-1}\right) + \begin{bmatrix}0 \\ \Delta\xi_l^2\end{bmatrix}\right) - [\phi_k]^n_{pN_k^2-1}}{\Delta\xi_l^2} & \text{if } q = N_k^2 - 1 \text{ and } \exists l \neq k \\ & \text{s.t. } [G_k]_{pN_k^2-1} \in \partial\mathcal{D}_l, \\[2ex] \dfrac{[\phi_k]^n_{pN_k^2-1} - [\phi_k]^n_{pN_k^2-2}}{\Delta\xi_k^2} & \text{otherwise.} \end{cases} \tag{A.10}$$

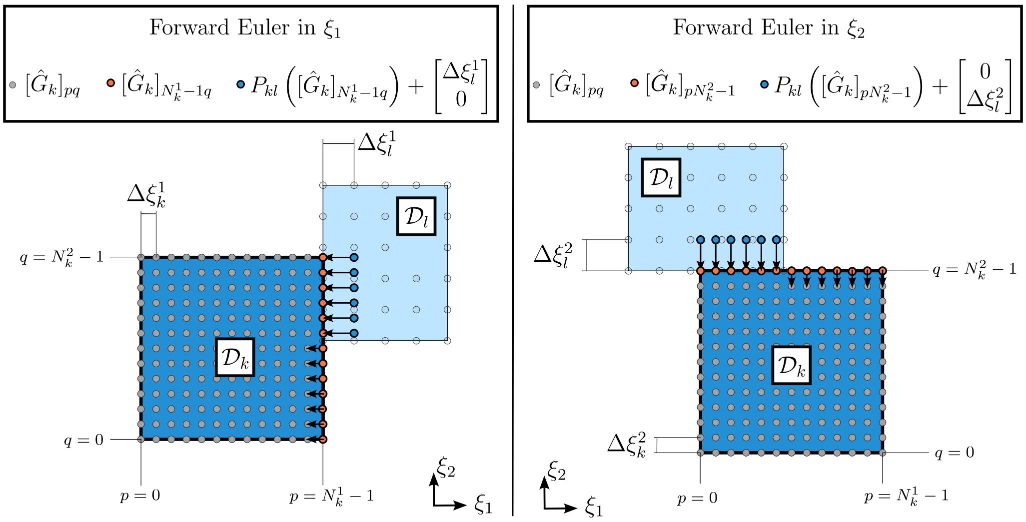

Figure A.28: Illustration on how to compute the forward Euler finite differences for the derivatives with respect to the parametric coordinates (ξ_1, ξ_2). As the scheme is trivial in the interior of the domain, we depict only how to compute the last two cases in equations A.9 and A.10.

We remark the distinction of the three different cases in each one the schemes in equations A.7 to A.10. The first case handles the points in the interior of the domain, in which the Euler schemes are computed trivially. The second case addresses points in the internal interfaces between patches, so as to satisfy the continuity condition in equation A.1. Let us further notice that this term requires the evaluation of $\hat{\phi}_l$ in an extension of the grid of patch k into patch l. Since this extension point generally does not coincide with another point of grid the $[\hat{G}_l]$, the evaluation of $\hat{\phi}_l$ is computed with a linear interpolation of the adjacent points. And finally, the third case is related to the points with external boundary, where the Neumann boundary condition is applied.

Appendix B. Regularization problem in tangent bundle

Let us now discuss how to numerically implement the regularization problem in the tangent bundle. We start by recalling the definition of the space discretion of problem (43), which is notated by $\Theta^h(\mathcal{D}_k) \subset \Theta(\mathcal{D}_k)$ and is given by the following set:

$$\Theta^h(\mathcal{D}_k) = \left\{ \boldsymbol{\vartheta} = \hat{\boldsymbol{\vartheta}} \circ \mathbf{S}_k^{-1} : \hat{\boldsymbol{\vartheta}} \in \bigotimes_{i \in [\![0,2]\!]} \mathcal{R}(p_{k_i}, \Xi_{k_i}) \right\} \tag{B.1}$$

where $\mathcal{R}(p_{k_i}, \Xi_{k_i})$ is the set of rational B-splines function defined on the patch k of degree p_{k_i} and with the knot vector Ξ_{k_i} in the direction $i \in \{0, 1, 2\}$.

The solutions $\boldsymbol{\theta}^h$ of problem (42) will therefore be obtain as a linear combination of function of $\Theta^h(\mathcal{D}_k)$, as i.e

$$\boldsymbol{\theta}^h(\boldsymbol{\xi}) = \sum_{A=1}^{N} R_A(\boldsymbol{\xi}) \boldsymbol{w}_A \tag{B.2}$$

One can classically show that the regularization problem (42), which is nothing but a Laplace equation, can be rewritten using this discretization as a linear system of the form:

$$\mathbf{L}\boldsymbol{w} = \mathbf{j} \tag{B.3}$$

where $\mathbf{L} \in \mathcal{M}_{3N \times 3N}(\mathbb{R})$ is the global stiffness matrix related to the bilinear form in the left of equation (42) and $\mathbf{j} \in \mathbb{R}^{3N}$ is a vector related to the source term in the right-hand side in the same equation.

In order to solve problem (42) in the tangent bundle, we transform this discrete representation into a local coordinates system, where the conditions in (40) can naturally be expressed as Dirichlet boundary conditions. To achieve this, we can construct a global rotation matrix $\mathbf{R}_g \in \mathcal{M}_{3N \times 3N}(\mathbb{R})$. For such purposes, we will use an approximation of the orthonormal basis $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3)$ by using the Greville abscissae [30], which can be defined by:

$$\mathcal{G}_i = \left\{ \tilde{\xi}_A = \frac{1}{p_i} \sum_{j=1}^{p_i} \xi_{A+j}, \quad \forall A \in [\![1, N_i]\!] \right\} \quad \forall i \in [\![0, 2]\!] \tag{B.4}$$

Let us view the global rotation matrix $\mathbf{R}_g$ as an embedded matrix in $\mathcal{M}_{N \times N}(\mathcal{M}_{3\times 3}(\mathbb{R}))$. In this setting, we define $\mathbf{R}_g$ as:

$$(\mathbf{R}_g)_{ij} = \begin{cases} \tilde{\mathbf{T}}_i, \text{ if } i = j \\ \mathbf{0}_{\mathcal{M}_{3\times3}} \quad \text{otherwise} \end{cases} \qquad \forall (i,j) \in [\![0, N]\!]^2 \tag{B.5}$$

where $\tilde{\mathbf{T}}_i$ is an approximation of the local basis rotation matrix $\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3)$, defined via the use of an approximation of the covariant basis as $\tilde{\mathbf{a}}_1^i = \hat{\partial}_1 \mathbf{S}(\tilde{\boldsymbol{\xi}}_i)$ and $\tilde{\mathbf{a}}_2^i = \hat{\partial}_2 \mathbf{S}(\tilde{\boldsymbol{\xi}}_i)$ using the Greville abscissae:

$$\tilde{\mathbf{T}}_i = \left(\frac{\tilde{\mathbf{a}}_1^i}{|\tilde{\mathbf{a}}_1^i|}, \frac{\tilde{\mathbf{a}}_2^i}{|\tilde{\mathbf{a}}_2^i|}, \frac{\tilde{\mathbf{a}}_1^i \times \tilde{\mathbf{a}}_2^i}{|\tilde{\mathbf{a}}_1^i \times \tilde{\mathbf{a}}_2^i|} \right) \tag{B.6}$$

Finally, one can use the global rotation matrix $\mathbf{R}_g$ to transform the matrix linear system $\mathbf{L}\boldsymbol{w} = \mathbf{j}$ into its description in local coordinates:

$$\mathbf{R}_g^T \mathbf{L} \mathbf{R}_g \boldsymbol{w} = \mathbf{R}_g \mathbf{j} \tag{B.7}$$

References

[1] Y. B. T.J.R. Hughes, J.A. Cottrell, NURBS as a Pre-Analysis Tool: Geometric Design and Mesh Generation, John Wiley & Sons, Ltd, 2009, Ch. 2, pp. 19–68. `arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470749081.ch2`, `doi:https://doi.org/10.1002/9780470749081.ch2`.

[2] G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method* 1, Journal of Computational Physics 194 (2004) 363–393. `doi:10.1016/j.jcp.2003.09.032`.

[3] M. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, Computer Methods in Applied Mechanics and Engineering 192 (2003) 227–246. `doi:10.1016/S0045-7825(02)00559-5`.

[4] R. Ansola, J. Canales, J. Tárrago, J. Rasmussen, An integrated approach for shape and topology optimization of shell structures, Computers & Structures - COMPUT STRUCT 80 (2002) 449–458. `doi:10.1016/S0045-7949(02)00019-6`.

[5] S. Goo, S. Wang, J. Hyun, J. Jung, Topology optimization of thin plate structures with bending stress constraints, Computers & Structures 175 (10 2016). `doi:10.1016/j.compstruc.2016.07.006`.

[6] T. Ho-Nguyen-Tan, H.-G. Kim, Level set-based topology optimization for compliance and stress minimization of shell structures using trimmed quadrilateral shell meshes, Computers & Structures 259 (2021) 106695. `doi:10.1016/j.compstruc.2021.106695`.

[7] F. Agnelli, G. Nika, A. Constantinescu, Design of thin micro-architectured panels with extension–bending coupling effects using topology optimization, Computer Methods in Applied Mechanics and Engineering 391 (2022) 114496. `doi:10.1016/j.cma.2021.114496`.

[8] J. Gao, X. Wu, M. Xiao, V. P. Nguyen, L. Gao, T. Rabczuk, Multi-patch isogeometric topology optimization for cellular structures with flexible designs using nitsche's method, Computer Methods in Applied Mechanics and Engineering 410 (2023) 116036. `doi:https://doi.org/10.1016/j.cma.2023.116036`.
URL `https://www.sciencedirect.com/science/article/pii/S0045782523001603`

[9] E. Shakour, O. Amir, Topology optimization with precise evolving boundaries based on iga and untrimming techniques, Computer Methods in Applied Mechanics and Engineering 374 (2021) 113564. `doi:10.1016/j.cma.2020.113564`.

[10] W. Wall, M. Frenzel, C. Cyron, Isogeometric structural shape optimization. comput methods appl mech eng, Computer Methods in Applied Mechanics and Engineering - COMPUT METHOD APPL MECH ENG 197 (2008) 2976–2988. `doi:10.1016/j.cma.2008.01.025`.

[11] M. Xu, S. Wang, X. Xie, Level set-based isogeometric topology optimization for maximizing fundamental eigenfrequency, Frontiers of Mechanical Engineering 14 (02 2019). `doi:10.1007/s11465-019-0534-1`.

[12] Y. Wang, D. Benson, Isogeometric analysis for parameterized lsm-based structural topology optimization, Computational Mechanics 57 (01 2016). `doi:10.1007/s00466-015-1219-1`.

[13] L. Dede, M. Borden, T. Hughes, Isogeometric analysis for topology optimization with a phase field model, Archives of Computational Methods in Engineering 19 (09 2012). `doi:10.1007/s11831-012-9075-z`.

[14] P. Kang, S.-K. Youn, Isogeometric topology optimization of shell structures using trimmed nurbs surfaces, Finite Elements in Analysis and Design 120 (2016) 18–40. `doi:10.1016/j.finel.2016.06.003`.

[15] D. Benson, Y. Bazilevs, M. Hsu, T. Hughes, Isogeometric shell analysis: The reissner–mindlin shell, Computer Methods in Applied Mechanics and Engineering 199 (2010) 276–289. `doi:10.1016/j.cma.2009.05.011`.

[16] N. Adam, Méthodes isogéométrique multipatch pour des coques épaisses non linéaires avec contact, Theses, Institut Polytechnique de Paris (Sep. 2020).
URL `https://theses.hal.science/tel-02982163`

[17] P. G. Ciarlet, C. Mardare, An Introduction to Shell Theory, in: Differential Geometry: Theory and Applications, Vol. 9, CO-PUBLISHED WITH HIGHER EDUCATION PRESS, 2008, pp. 94–184. `doi:10.1142/9789812771476\_0002`.
URL `https://hal.sorbonne-universite.fr/hal-01077558`

[18] N. Adam, P. Tallec, M. Zarroug, Multipatch isogeometric mortar methods for thick shells, Computer Methods in Applied Mechanics and Engineering 372 (2020) 113403. `doi:10.1016/j.cma.2020.113403`.

[19] G. Nika, A. Constantinescu, Design of multi-layer materials using inverse homogenization and a level set method, Computer Methods in Applied Mechanics and Engineering 346 (2019) 388–409. `doi:10.1016/j.cma.2018.11.029`.

[20] S. Osher, C.-W. Shu, High-order essentially nonoscillatory schemes for hamilton–jacobi equations, Siam Journal on Numerical Analysis - SIAM J NUMER ANAL 28 (08 1991). `doi:10.1137/0728049`.

[21] A. Laurain, A level set-based structural optimization code using fenics, Structural and Multidisciplinary Optimization 58 (05 2017). `doi:10.1007/s00158-018-1950-2`.

[22] OpenCASCADE Development Team, BRepAlgoAPI Section: Boolean Operations on B-Rep Models, OpenCASCADE Documentation, accessed on February 23, 2024 (February 2022).
URL `https://dev.opencascade.org/doc/refman/html/class_b_rep_algo_a_p_i___section.html`

[23] B. Marussig, T. Hughes, A review of trimming in isogeometric analysis: Challenges, data exchange and simulation aspects, Archives of Computational Methods in Engineering 25 (2017) 1–69. `doi:10.1007/s11831-017-9220-9`.

[24] G. Allaire, Conception Optimale de Structures, Springer Berlin Heidelberg, 2007. `doi:10.1007/978-3-540-36856-4`.

[25] O. Sigmund, Sigmund, o.: A 99 line topology optimization code written in matlab. structural and multidisciplinary optimization 21, 120-127, Structural and Multidisciplinary Optimization 21 (2001) 120–127. `doi:10.1007/s001580050176`.

[26] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. `doi:10.1038/s41592-019-0686-2`.

[27] P. Dierckx, Curve and surface fitting with splines, in: Monographs on numerical analysis, 1994, pp. 249–266.

[28] S. K. Lam, A. Pitrou, S. Seibert, Numba: A llvm-based python jit compiler, in: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, 2015, pp. 1–6.

[29] O. Schenk, K. Gärtner, W. Fichtner, A. Stricker, Pardiso: A high-performance serial and parallel sparse linear solver in semiconductor device simulation, Future Generation Computer Systems 18 (2000) 69–78. `doi:10.1016/S0167-739X(00)00076-5`.

[30] G. Farin, Curves and Surfaces for Computer-Aided Geometric Design. A Practical Guide, Fourth Edition, Academic Press, San Diego, 1997.