Isogeometric multipatch surface fitting in tomographic images: application to lattice structures

Dorian Bichet, Jean-Charles Passieux, Jean-Noël Périé, Robin Bouclier

Isogeometric multipatch surface fitting in tomographic images: application to lattice structures

D. Bichet[a,b], J.C. Passieux[b], J.N. Périé[b] and R. Bouclier[b,a,c,*]

[a]Institut de Mathematiques de Toulouse (IMT), Univ Toulouse, UPS-INSA-CNRS UMR5219, 135 Avenue de Rangueil, Toulouse, F31077, France
[b]Institut Clement Ader (ICA), Univ Toulouse, INSA-ISAE-Mines Albi-UPS-CNRS, UMR 5312, 3 Rue Caroline Aigle, Toulouse, F-31400, France
[c]Institut Universitaire de France (IUF), France

ARTICLE INFO

Keywords:
Image-based modeling
Digital twin
IsoGeometric Analysis
Active contours
Virtual Image Correlation
Multipatch B-spline

Abstract

Additive manufacturing has enabled the production of cellular architected (lattice) structures known for their exceptional mechanical performances. However, the printed components often exhibit geometric defects on a scale close to that of lattice struts, leading to significant deviations in mechanical behavior when comparing simulations based on the as-designed (defect-free) geometry with experimental tests on the as-manufactured (imperfect) geometry. In this work, we develop a method to extract an analysis-suitable CAD-based geometry from 3D scan data. We start by building a multipatch B-spline surface model of the as-designed lattice boundary, and then deform it to match its as-manufactured counterpart observed in a volumetric image. To achieve this, key contributions include a data fitting metric based on the Virtual Image Correlation approach, combined with an image learning component; the integration of the membrane strain energy of the surface for regularization; the enforcement of higher continuity between patches where appropriate; an automatic estimation of the pose of the CAD object in the image; and the computation of a distance indicator map between the aligned CAD model and the as-manufactured surface. These elements enable comprehensive, accurate, and efficient measurement of geometric defects in lattice structures. Validated through various experiments, including those on a BCC lattice structure, this method achieves sub-voxel accuracy. Ultimately, it provides a compact and explicit representation of the as-manufactured geometry, maintaining the same CAD-based discretization as the initial design, and thereby facilitating quantitative defect assessment.

1. Introduction

Today, with the advent of additive manufacturing, it becomes possible to produce structures with almost unlimited geometric forms [8]. A notable example is the development of cellular architected structures, also commonly referred to as *lattice structures*, which are currently experiencing significant growth in industry, particularly due to metal additive manufacturing [50, 12, 36]. These structures are designed to mimic natural architectures, such as bones and wood, allowing for exceptional specific performance, starting with unprecedented stiffness-to-weight ratios. In practice, lattice structures typically consist of periodically repeating a well-designed unit-cell, often composed of struts (or beams) interconnected in various ways via nodes. Amongst others, these multiscale heterogeneous structures can be engineered to be highly stretchable, auxetic, and multifunctional. Therefore, in addition to their appeal for lightweight components, they hold crucial interest for applications such as energy absorption, thermal management, and the design of medical implants [53, 6, 63].

The remarkable design freedom offered by metal additive manufacturing comes at the cost of process complexity, which results, in particular, to geometric strut-level defects compared to the initial Computer-Aided Design (CAD) geometry developed during the design stage. In this respect, strut thickness variations, strut waviness, displacements of lattice nodes, and excess material accumulation in the nodes can be observed [40, 37, 20, 25]. Given the unique geometry of lattice structures, which are often composed of slender beams and exhibit rich symmetries, both of which can be easily altered by imperfections, their mechanical behavior is influenced not only by cell topology and shape but also by these geometric defects. Consequently, significant deviations in mechanical behavior are observed when comparing simulations based on the idealized, defect-free geometry – commonly referred to as the *as-designed*

*Corresponding author
bouclier@insa-toulouse.fr (R. Bouclier)
ORCID(s):

geometry – with experimental tests conducted on the real, imperfect geometry – also known as the *as-manufactured* geometry. To characterize the mechanical behavior of such materials, it is thus crucial to create a digital mechanical twin, i.e., a geometric model that (i) incorporates the real, specific geometry of the tested sample, and (ii) can be easily used for numerical simulation (in the latter case, it is referred to as an analysis-suitable model). This process typically begins with X-ray micro-computed tomography (μ-CT) to reveal the internal architecture of the structure [2, 18]. The reconstructed volume images are then considered to develop specimen-specific image-based models that account for the as-manufactured geometric architecture of the structure [40, 61, 24, 37, 38].

Different approaches can be conducted to extract an analysis-suitable geometry from scan data. One of the earliest approaches involves constructing a Finite Element (FE) model using the well-known marching cubes method [41, 44]. More precisely, after selecting a threshold graylevel to distinguish between the two phases of the material (solid and void in case of lattice structures), the marching cubes algorithm allows for the construction of a triangular surface mesh (at the voxel scale) of the boundary of the object. Subsequently, a volumetric mesh is created from this surface mesh, and finally, the FE method is applied for numerical simulation. This approach is the most widespread [40, 2, 24, 52] and there are numerous tools available to implement it (such as the Avizo software, the scikit-image Python library [58], etc.). However, the drawback of this approach is that it often leads to numerically expensive models. As a result, a recent trend that seems to be emerging is to approximate such FE meshes in post-processing using less-demanding models, such as those based on non-uniform rational B-spline (NURBS) functions from CAD [47]. That being said, another drawback of the method is that it does not bridge the gap with the original, defect-free model. The FE model is constructed independently of the knowledge of the as-designed geometry. Consequently, achieving a flexible and comprehensive description of all geometrical variations remains challenging [40].

Alternatively, a second approach relies on immersed methods, where the geometry is directly represented by voxels (using either simple segmentation or a level-set method), while a simple Cartesian mesh (utilizing FE or B-spline functions from CAD) encompassing the imaged structure is used to represent the mechanical fields. This option simplifies mesh generation and offers models with a contained number of degrees-of-freedom (DOF). These methods have been applied in various fields, including the analysis of trabecular bones [59, 17], foams [19, 51], composite materials [60], and lattice structures [37, 38, 33]. Nevertheless, it is important to note that (i) their proper execution can be complex, particularly concerning the treatment of ill-conditioning issues and numerical integration over cut elements [54, 15], and (ii) they only provide an implicit representation of the geometry, thereby breaking the link with the initial as-designed CAD, as with the FE marching cubes approach.

Eventually, a third option is to consider template-fitting techniques. This class of methods begins with the construction of a parameterized reference geometry, known as the template geometry, which captures the essential features of a specimen. This geometry is then (iteratively) deformed to match the scan data. This approach was first used to deform FE meshes [4]. Later, with the advent of IsoGeometric Analysis (IGA) [29, 13], it gained significant interest for constructing patient-specific models in the biomedical field using B-spline and NURBS functions [65, 57, 62]. This resulted in light, regular, CAD-compatible models that can offer increased per-DOF accuracy in numerical simulations. Very recently, research closely related to this approach has emerged, specifically focusing on lattice structures. Notably, isogeometric interconnected beam models of 2D cellular materials have been constructed from image data [46], and defect measurement in lattice structures has been performed using a modal decomposition of beam-type kinematics with variable cross-section [14]. Furthermore, FE models of lattice structures have been deformed using radial basis functions to align with defects observed in tomographic projections [11]. Overall, these techniques require starting from a description whose topology is compatible with that of the scanned object. This requirement is relatively easy to meet for lattice structures since the as-designed model is available. In turn, these methods offer the advantage of providing the geometric transformation from the as-designed to the as-manufactured component, which, in our opinion, is highly beneficial for explicitly describing defects.

Building on this state-of-the-art, our objective in this work is to extract an analysis-suitable CAD-based geometry from scan data, allowing for a flexible, comprehensive, and accurate measurement of the geometric defects encountered in lattice structures. To this end, our method begins with the construction of a B-spline surface model that finely represents the boundary (the skin) of the considered lattice structure. Given the geometric complexity of this type of structure, a multipatch B-spline model is derived, with particular care taken to ensure its analysis-suitability, i.e. creating patches that are conformally and smoothly connected (where appropriate), from which direct simulation via IGA could be performed [26, 42]. Then, we develop a template fitting technique that involves repositioning the control

points of the as-designed B-spline surface to obtain the as-manufactured surface, as observed through the graylevels in the volumetric image obtained from X-ray CT. To achieve this, we adopt the framework of active contour methods developed in computer vision [32, 1, 30]. Within the latter, we propose using a data fitting metric based on the Virtual Image Correlation (VIC) method, which has been developed over the years, mainly in 2D in mechanics [23, 3, 22, 46] and, to a lesser extent, in 3D for simple geometries [31]. The principle is to create a virtual image representing the boundary to be measured and to seek the optimal geometric mapping that allows interrogation of the real image in a region that best matches the virtual image. This approach is particularly interesting in our context as it can reach subvoxel accuracy and it has a local character, as only a region near the boundary is evaluated in the real image, making it an efficient method. Furthermore, and this represents a significant contribution of our work compared to the state-of-the-art in general surface fitting methods (in both the computer vision and mechanics communities), we propose using the membrane strain energy of the surface model in a Tikhonov-like scheme to regularize the inverse problem. This term appeared necessary in 3D to avoid spurious in-plane modes while allowing the surface to deform in the normal direction to fit the fine geometric defects. Finally, we introduce an additional term in the functional aimed at ensuring C^1-continuity between the patches, where appropriate with respect to the considered structures. With all the aforementioned components, we end up with a method capable of obtaining an explicit representation of the as-manufactured structure using a CAD-based model that shares the same support as the initial as-designed geometry.

The paper is organized as follows: Section 2 introduces the geometric analysis-suitable B-spline multipatch surface model considered in this study for fitting tomographic image data. Then, in Section 3, we present our surface fitting method to obtain the digital twin of the as-manufactured geometry, thoroughly detailing all its components, such as the three metrics used in the minimization process. Finally, in Section 4, we demonstrate the performance of the developed CAD-based fitting approach through various examples, including synthetic and real images of lattice structure components, up to an entire Body Centered Cubic (BCC) cell, before formulating concluding remarks in Section 5.

2. Geometric model

This section introduces the context of the study from a geometric point of view. To this end, we provide the necessary elements regarding (i) the B-spline modeling technique, (ii) the parametric representation of 3D surfaces, and (iii) the multipatch modeling approach related to the analysis-suitable nature of the model.

2.1. B-spline description

B-spline technology, along with its NURBS extension, has become the standard over the years for geometric modeling in CAD and computer graphics [5, 16, 39]. These techniques are now also well-known in the scientific computing community, particularly with the advent of IGA. Therefore, only the fundamentals are provided here. For further details, the reader may consult, e.g., the books [48, 13, 9] and the references therein. In this work, we exclusively focus on B-spline geometries (thus excluding NURBS) and move the positions of their control points to modify their shape. This approach already allows to accurately approximate a wide range of geometries (including conic sections), provided a sufficient level of refinement [56, 10]. Additionally, while NURBS also enable shape variation through modification of the control point weights [49, 43], it has been observed that it is generally sufficient to rely solely on the control point coordinates for free-form surfaces [35, 27, 62]. In the following, we restrict ourselves to single-patch geometries, the extension to multipatch modeling being addressed later in Section 2.3.

A B-spline entity of topological dimension $\mathrm{n_{param}}$ is defined by a specific mapping from a parametric space, usually $\Omega_{\mathrm{param}} = [0,1]^{\mathrm{n_{param}}}$, to a physical space $\mathbb{R}^{\mathrm{n_{phy}}}$ (where $\mathrm{n_{phy}}$ is the spatial dimension of the problem). This mapping is a linear combination of B-spline basis functions with coefficients corresponding to control points that reside in the physical space. Two examples of B-spline geometries are given in Fig. 1: the first corresponds to the case $\mathrm{n_{param}} = 1$ and $\mathrm{n_{phy}} = 2$ (see Figs. 1a and 1b), while the second involves $\mathrm{n_{param}} = 2$ and $\mathrm{n_{phy}} = 3$ (see Figs. 1c and 1d). In what follows, we provide the necessary mathematical details behind these figures.

Univariate B-spline basis functions are piecewise polynomials defined by their polynomial degree p and a set of non-decreasing parametric coordinates $\xi_i \in \mathbb{R}$, covering the interval $[0,1]$, collected into a knot-vector Ξ. The coordinates ξ_i, referred to as knots, divide the parametric space into elements, and the entire interval $[0,1]$ constitutes the isogeometric patch in the parametric space. An example of univariate quadratic B-spline basis functions defined

over a mesh composed of 4 elements is given in Fig. 1a. The extension to multivariate geometries, i.e., surfaces and volumes, is achieved using a tensor-product construction of univariate B-spline functions. Let us denote the associated $\mathrm{n_{bf}}$ B-spline basis functions as follows:

$$\mathbf{N}: \begin{array}{ccc} \Omega_{\text{param}} & \to & \mathbb{R}^{\mathrm{n_{bf}}} \\ \boldsymbol{\xi} & \mapsto & \begin{pmatrix} \mathrm{N}_1(\boldsymbol{\xi}) \\ \vdots \\ \mathrm{N}_{\mathrm{n_{bf}}}(\boldsymbol{\xi}) \end{pmatrix}. \end{array} \tag{1}$$

Perhaps the most interesting feature of B-spline functions is their potential for high regularity. If m is the multiplicity of a given knot, the functions are C^{p-m}-continuous at that location. This property is particularly appealing from a geometric representation and shape fitting perspective, as it allows for the definition of smooth free-form shapes and shape updates using few DOFs. Returning to Fig. 1a, it can be observed that the regularity of the B-spline functions at the interior knots is C^1, leading to a smooth spline curve and correspondingly smooth modifications of the latter when associated control points are moved (see Fig. 1b). Another attractive property of B-spline basis functions is that they have compact support, meaning their influence is limited to a small interval, so that moving a control point of the associated geometric entity results in a local shape modification (see again Fig. 1b).

From a mathematical point of view, considering the case where $\mathrm{n_{phy}} = 3$ and denoting:

$$\mathbf{X} = (\mathrm{X}_{x,1}, ..., \mathrm{X}_{x,\mathrm{n_{bf}}}, \mathrm{X}_{y,1}, ..., \mathrm{X}_{y,\mathrm{n_{bf}}}, \mathrm{X}_{z,1}, ..., \mathrm{X}_{z,\mathrm{n_{bf}}})^T, \tag{2}$$

the vector that gathers all the spatial coordinates $(\mathrm{X}_{x,i}, \mathrm{X}_{y,i}, \mathrm{X}_{z,i})_{i\in[\![1,\mathrm{n_{bf}}]\!]}$ of the $\mathrm{n_{bf}}$ control points associated with the $\mathrm{n_{bf}}$ B-spline basis functions, the B-spline mapping can be expressed as:

$$\mathbf{x}: \begin{array}{ccc} \Omega_{\text{param}} & \to & \mathbb{R}^{\mathrm{n_{phy}}} \\ \boldsymbol{\xi} & \mapsto & \mathbf{N}^{3\mathrm{D}}(\boldsymbol{\xi})\,\mathbf{X}, \quad \text{with} \quad \mathbf{N}^{3\mathrm{D}}(\boldsymbol{\xi}) := \begin{pmatrix} \mathbf{N}(\boldsymbol{\xi})^T & 0 & 0 \\ 0 & \mathbf{N}(\boldsymbol{\xi})^T & 0 \\ 0 & 0 & \mathbf{N}(\boldsymbol{\xi})^T \end{pmatrix}. \end{array} \tag{3}$$

The corresponding B-spline geometric entity or initial shape is then given by:

$$\Omega = \left\{ \mathbf{x}(\boldsymbol{\xi}), \forall\, \boldsymbol{\xi} \in \Omega_{\text{param}} \right\}, \tag{4}$$

resulting in a (possibly curved) line if $\mathrm{n_{param}} = 1$ (see Fig 1b), a (possibly non-planar) surface if $\mathrm{n_{param}} = 2$ (see Fig 1d), or a volume if $\mathrm{n_{param}} = 3$. At this point, it is important to note that a single patch can only model elementary geometries, due to the tensor-product structure of the parametric space. Consequently, to generate more complex shapes, we will employ multipatch models, see Section 2.3. With this in mind, the positions of the control points can be adjusted to create the intricate shapes encountered in engineering. In particular, even though we use only B-splines, it is possible to accurately approximate conic sections. This is illustrated in Fig.1b which shows the approximation of a quarter circle using a four-element quadratic B-spline curve (see [56, 10] for further details).

Finally, considering the case where $\mathrm{n_{phy}} = 3$ and denoting (similarly to Eq. (2)):

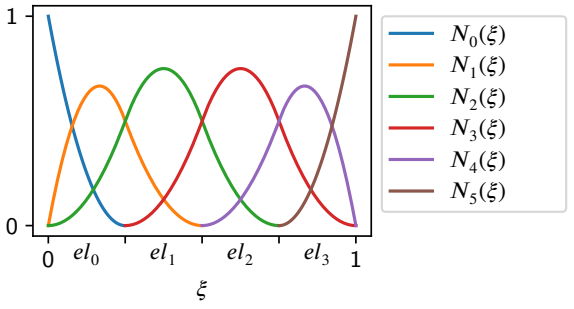
$$\mathbf{U} = (\mathrm{U}_{x,1}, ..., \mathrm{U}_{x,\mathrm{n_{bf}}}, \mathrm{U}_{y,1}, ..., \mathrm{U}_{y,\mathrm{n_{bf}}}, \mathrm{U}_{z,1}, ..., \mathrm{U}_{z,\mathrm{n_{bf}}})^T, \tag{5}$$

the vector that collects all the displacement components $(\mathrm{U}_{x,i}, \mathrm{U}_{y,i}, \mathrm{U}_{z,i})_{i\in[\![1,\mathrm{n_{bf}}]\!]}$ of the $\mathrm{n_{bf}}$ control points, the shape modification can be expressed through the following displacement field:

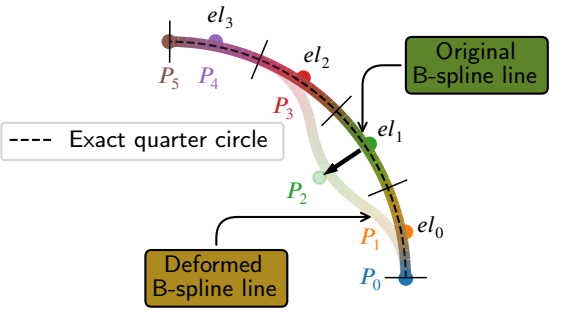
$$\mathbf{u}: \begin{array}{ccc} \Omega_{\text{param}} \times \mathbb{R}^{\mathrm{n_{bf}} \times \mathrm{n_{phy}}} & \to & \mathbb{R}^{\mathrm{n_{phy}}} \\ (\boldsymbol{\xi}; \mathbf{U}) & \mapsto & \mathbf{N}^{3\mathrm{D}}(\boldsymbol{\xi})\,\mathbf{U} \end{array} \tag{6}$$

It is defined in an isoparametric element manner from Eq. (3), which is also consistent with the isogeometric concept [29, 13]. Additionally, note that the dependence on $\mathbf{U}$ is explicitly shown in the mapping (6) since $\mathbf{U}$ represents the model parameters to optimize during our fitting procedure. More precisely, starting with an as-designed B-spline geometric model, the objective is to determine the appropriate control point displacements, yielding a deformed configuration:

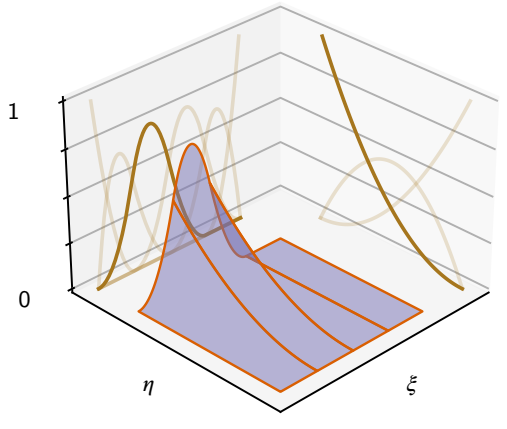
$$\Omega_{\text{moved}}(\mathbf{U}) = \left\{ \mathbf{x}(\boldsymbol{\xi}) + \mathbf{u}(\boldsymbol{\xi}; \mathbf{U}), \forall\, \boldsymbol{\xi} \in \Omega_{\text{param}} \right\}, \tag{7}$$



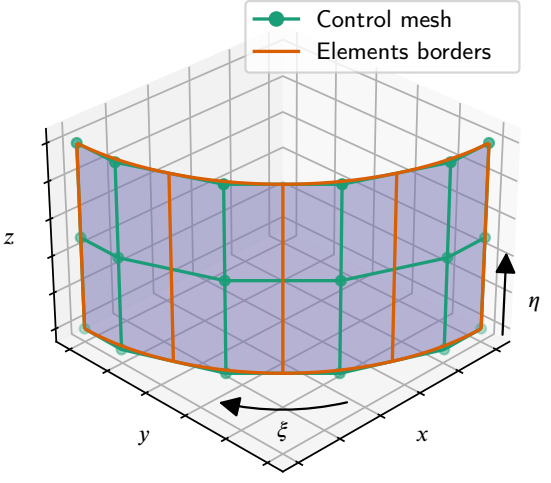
(a) Univariate B-spline basis functions of polynomial degree 2 for a mesh composed of 4 elements (plotted in the parametric space $[0, 1]$). The corresponding knot-vector is $\Xi = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$.



(b) An example of B-spline geometry and shape modification using the space generated by the B-spline functions shown in (a). The initial shape accurately approximates a quarter circle. This geometry is then modified by radially moving control point P_2, resulting in a local smooth shape variation. The local influence of the control points can also be observed through the color gradient of the geometric entity, which corresponds to the color of the different basis functions in (a).



(c) Bivariate B-spline basis functions created by performing a tensor-product between the univariate basis functions shown in (a) (which depend on ξ in this example) and the univariate basis functions associated with a single B-spline element of polynomial degree 2 (the latter depending on η in this case). The surface plotted in blue is an example of bivariate basis function.



(d) Example of a surface B-spline geometry using the space generated by the B-spline functions in (c). The control points of the surface are chosen such that they create an extrusion of (b) (into the z-direction). The affine interpolation of the control points is called the control mesh, which is shown in green here.

Figure 1: Examples of B-spline geometrical descriptions: (a)-(b) case of a curved line; (c)-(d) case of a non-planar surface.

that matches the tomographic image data.

Again, the use of a B-spline description emerges as an optimal solution in our context. Due to its potential for increased smoothness, it offers a concise and compact representation of the geometry and, even more importantly, of the shape modification, which also results in a natural regularization of the inverse fitting problem.

2.2. 3D surface description

In this work, we focus on the deformation of a non-planar B-spline surface in $\mathbb{R}^3$. From here on, we thus consider only the case where $\mathrm{n}_{\mathrm{param}} = 2$ and $\mathrm{n}_{\mathrm{phy}} = 3$. The B-spline formalism offers a natural way to represent geometries in parametric form, which allows for the immediate definition of many geometrical surface descriptors. This explains, in part, why shell-type formulations have seen a substantial increase in development with the advent of IGA. In this section, we describe some of the geometrical surface descriptors that will be used in our fitting approach and refer the interested reader to the literature on shell theory in IGA, e.g. [34, 21, 9], for further details.

To start with, let us specify that the coordinates in the parametric space $\boldsymbol{\xi}$ are taken here as $\boldsymbol{\xi} = (\xi, \eta)$. Using $\times$ to denote the vector product, a local basis, known as the covariant basis, can be defined on the initial configuration Ω as follows:

$$\boldsymbol{A}_\xi := \frac{\partial \mathbf{x}}{\partial \xi} = \frac{\partial \mathbf{N}^{3\mathrm{D}}}{\partial \xi}\mathbf{X}\,; \tag{8}$$

$$\boldsymbol{A}_\eta := \frac{\partial \mathbf{x}}{\partial \eta} = \frac{\partial \mathbf{N}^{3\mathrm{D}}}{\partial \eta}\mathbf{X}\,; \tag{9}$$

$$\boldsymbol{A}_\gamma := \frac{\boldsymbol{A}_\xi \times \boldsymbol{A}_\eta}{\boldsymbol{S_e}}\,; \tag{10}$$

with:

$$\boldsymbol{S_e} = \left\| \boldsymbol{A}_\xi \times \boldsymbol{A}_\eta \right\|. \tag{11}$$

A schematic illustration is provided in Fig. 2(left). $\boldsymbol{A}_\xi$ and $\boldsymbol{A}_\eta$ are the tangent vectors of the basis. They are not normalized and are not necessarily orthogonal to each other. In contrast, $\boldsymbol{A}_\gamma$ is the normal vector. It is normalized and orthogonal to both $\boldsymbol{A}_\xi$ and $\boldsymbol{A}_\eta$. Making use of $\boldsymbol{A}_\gamma$, we introduce a volume entity called surface neighborhood, defined as:

$$\mathbf{x}^{\mathrm{nn}}(\xi, \eta, \gamma) = \mathbf{x}(\xi, \eta) + \gamma\, \boldsymbol{A}_\gamma(\xi, \eta), \quad \forall (\xi, \eta) \in \Omega_{\mathrm{param}}, \quad \forall \gamma \in [-h, h]. \tag{12}$$

This combines the B-spline surface mapping with an identity mapping along the normal direction (γ directly represents a length in the physical space). The resulting volume is normal to the surface, extending both above and below it, with a thickness of $2h$. This surface neighborhood will be crucial in our fitting strategy, as it will allow to explore the image within a thin region surrounding the boundary of the object to be measured.

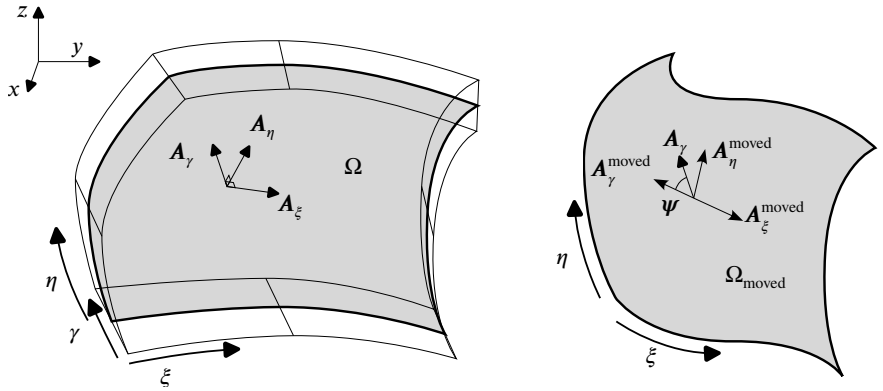


Figure 2: The initial B-spline surface, its local covariant basis, and the surface neighborhood are depicted on the left. On the right, the surface is shown after deformation, which, in particular, involves rotating its normal vector by an angle $\boldsymbol{\psi}$.

In the same way, a local, covariant basis can be defined in the deformed configuration $\Omega_{\mathrm{moved}}(\mathbf{U})$ (see Fig. 2(right)):

$$\boldsymbol{A}_\xi^{\mathrm{moved}} := \frac{\partial(\mathbf{x}+\mathbf{u})}{\partial \xi} = \frac{\partial \mathbf{N}^{3\mathrm{D}}}{\partial \xi}(\mathbf{X}+\mathbf{U})\,; \tag{13}$$

$$\boldsymbol{A}_{\eta}^{\text{moved}} := \frac{\partial(\mathbf{x}+\mathbf{u})}{\partial\eta} = \frac{\partial\,\mathbf{N}^{3\text{D}}}{\partial\eta}(\mathbf{X}+\mathbf{U})\,; \tag{14}$$

$$\boldsymbol{A}_{\gamma}^{\text{moved}} := \frac{\boldsymbol{A}_{\xi}^{\text{moved}} \times \boldsymbol{A}_{\eta}^{\text{moved}}}{\boldsymbol{S_e}^{\text{moved}}}\,; \tag{15}$$

with:

$$\boldsymbol{S_e}^{\text{moved}} = \left\| \boldsymbol{A}_{\xi}^{\text{moved}} \times \boldsymbol{A}_{\eta}^{\text{moved}} \right\|, \tag{16}$$

and the corresponding deformed surface neighborhood can be expressed as:

$$\mathbf{x}_{\text{moved}}^{\text{nn}}(\xi,\eta,\gamma;\mathbf{U}) = \mathbf{x}(\xi,\eta) + \mathbf{u}(\xi,\eta;\mathbf{U}) + \gamma\, \boldsymbol{A}_{\gamma}^{\text{moved}}(\xi,\eta;\mathbf{U}), \quad \forall(\xi,\eta)\in\Omega_{\text{param}}, \quad \forall\gamma\in[-h,h]. \tag{17}$$

At this stage, a volume displacement field representing the transformation of the surface neighborhood can be defined as:

$$\begin{aligned}\mathbf{u}^{\text{nn}}(\xi,\eta,\gamma;\mathbf{U}) &= \mathbf{x}_{\text{moved}}^{\text{nn}}(\xi,\eta,\gamma;\mathbf{U}) - \mathbf{x}^{\text{nn}}(\xi,\eta,\gamma)\\ &= \mathbf{u}(\xi,\eta;\mathbf{U}) + \gamma\left[\boldsymbol{A}_{\gamma}^{\text{moved}}(\xi,\eta;\mathbf{U}) - \boldsymbol{A}_{\gamma}(\xi,\eta)\right].\end{aligned} \tag{18}$$

The update of the surface neighborhood at each iteration of our optimization algorithm will be based on this field (see Section 3.2). To achieve this, we need to compute the derivative of $\mathbf{u}^{\text{nn}}$ with respect to $\mathbf{U}$, which mathematically corresponds to the Jacobian matrix $\boldsymbol{J}_{\mathbf{U}}\mathbf{u}^{\text{nn}}(\xi,\eta,\gamma;\mathbf{U})$. This operator can be computed analytically. It results in the sum of two terms: the derivative of the surface displacement $\mathbf{u}(\xi,\eta;\mathbf{U})$, which are simply the B-spline basis functions since the surface displacement is expressed as a linear combination of the latter with $\mathbf{U}$ (see Eq. (6)), and the derivative of the variation of the surface normal $\boldsymbol{A}_{\gamma}$. The first term is constant with respect to $\mathbf{U}$. For simplicity and to save some matrix products in the procedure, the second term is also approximated as a constant in this work, leading to a fully constant Jacobian matrix across the iterations, denoted $\boldsymbol{J}_{\mathbf{U}}\mathbf{u}_{\text{lin}}^{\text{nn}}(\xi,\eta,\gamma)$. To realize this approximation, we apply the following trick: by assuming that the rotation of $\boldsymbol{A}_{\gamma}$ is small during the shape modification (see Fig. 2(right) for illustration), we can express the variation $\boldsymbol{A}_{\gamma}^{\text{moved}} - \boldsymbol{A}_{\gamma}$ as a cross product between the initial normal vector and a "rotation vector" $\boldsymbol{\psi}\times\boldsymbol{A}_{\gamma}$. Consequently, Eq. (18) can be modified as follows (see, e.g., [34, 21, 9]):

$$\mathbf{u}_{\text{lin}}^{\text{nn}}(\xi,\eta,\gamma;\mathbf{U}) = \mathbf{u}(\xi,\eta;\mathbf{U}) + \gamma\left[\boldsymbol{\psi}(\xi,\eta;\mathbf{U})\times\boldsymbol{A}_{\gamma}(\xi,\eta)\right], \tag{19}$$

where, using $\cdot$ to denote the scalar product:

$$\boldsymbol{\psi} = \left[\frac{1}{\boldsymbol{S_e}}\frac{\partial\,\mathbf{u}}{\partial\eta}\cdot\boldsymbol{A}_{\gamma}\right]\boldsymbol{A}_{\xi} + \left[\frac{1}{\boldsymbol{S_e}}\frac{\partial\,\mathbf{u}}{\partial\xi}\cdot\boldsymbol{A}_{\gamma}\right]\boldsymbol{A}_{\eta}\,. \tag{20}$$

With this treatment, $\mathbf{u}_{\text{lin}}^{\text{nn}}$ becomes linear with respect to $\mathbf{U}$, thereby making the Jacobian matrix $\boldsymbol{J}_{\mathbf{U}}\mathbf{u}_{\text{lin}}^{\text{nn}}(\xi,\eta,\gamma)$ constant with respect to $\mathbf{U}$, which was the objective. More precisely, this constant Jacobian matrix is given by:

$$\boldsymbol{J}_{\mathbf{U}}\mathbf{u}_{\text{lin}}^{\text{nn}} = \mathbf{N}^{3\text{D}} - \frac{\gamma[\boldsymbol{A}_{\gamma}]_{\times}}{\boldsymbol{S_e}}\left[\boldsymbol{A}_{\xi}\left(\boldsymbol{A}_{\gamma}{}^{T}\,\frac{\partial\,\mathbf{N}^{3\text{D}}}{\partial\eta}\right) + \boldsymbol{A}_{\eta}\left(\boldsymbol{A}_{\gamma}{}^{T}\,\frac{\partial\,\mathbf{N}^{3\text{D}}}{\partial\xi}\right)\right], \tag{21}$$

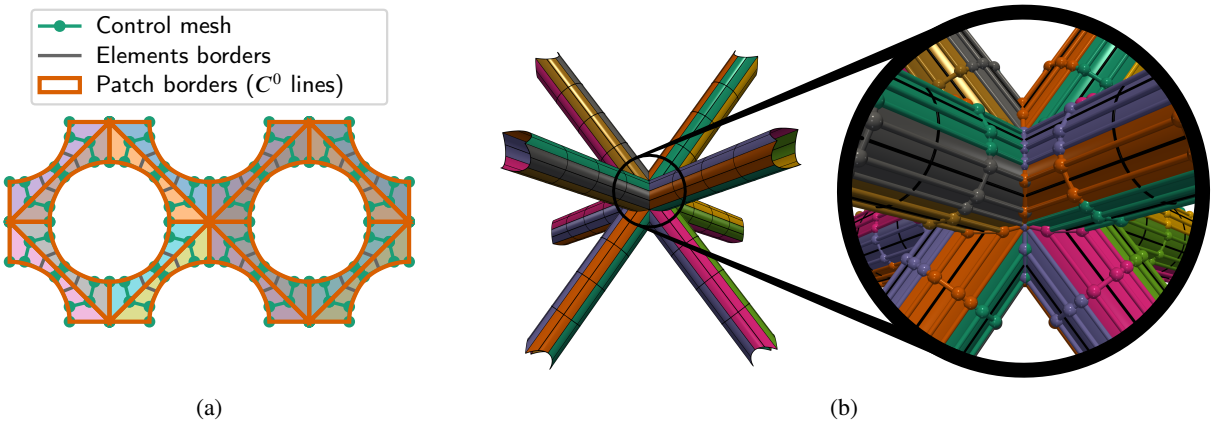
where $[\boldsymbol{A}_{\gamma}]_{\times}$ is the skew-symmetric matrix representing the cross product with $\boldsymbol{A}_{\gamma}$:

$$[\boldsymbol{A}_{\gamma}]_{\times} = \begin{pmatrix} 0 & -\mathrm{A}_{\gamma_z} & \mathrm{A}_{\gamma_y} \\ \mathrm{A}_{\gamma_z} & 0 & -\mathrm{A}_{\gamma_x} \\ -\mathrm{A}_{\gamma_y} & \mathrm{A}_{\gamma_x} & 0 \end{pmatrix}. \tag{22}$$

Remark 2.1. *Let us emphasize that it is not necessary to compute the exact Jacobian matrix $\boldsymbol{J}_{\mathbf{U}}\mathbf{u}^{\text{nn}}(\xi,\eta,\gamma;\mathbf{U})$ to achieve convergence in our procedure described in Section 3. Indeed, our strategy employs a Gauss-Newton minimization algorithm, where the Hessian and gradient are approximated using $\boldsymbol{J}_{\mathbf{U}}\mathbf{u}_{\text{lin}}^{\text{nn}}(\xi,\eta,\gamma)$ instead of $\boldsymbol{J}_{\mathbf{U}}\mathbf{u}^{\text{nn}}(\xi,\eta,\gamma;\mathbf{U})$. As long as the Gauss-Newton method generates a descent direction, convergence of the algorithm is ensured [45], which has been numerically observed in all the tests (potentially involving large local rotations) conducted in this work.*

2.3. Multipatch representation

Due to the tensor-product structure of multivariate B-spline mappings, a single-patch B-spline surface geometry is topologically equivalent to a rectangle. Therefore, for complex geometries, it is necessary to use multipatch B-spline models, i.e., models constructed by combining several patches. In this work, we consider multipatch models that are analysis-suitable, meaning they can be directly used by an isogeometric analysis code with standard functionalities. This requires that the multiple patches in the model are conformally juxtaposed along their edges, i.e., the discretizations on the common boundaries of the patches match perfectly. Illustrations of conforming multipatch geometries are provided in Fig.3. Note that the geometry depicted in Fig.3b will be utilized in the examples Section 4. In addition, to fully exploit the higher regularity of B-spline functions in solving our inverse problem, we aim to minimize the number of C^0-continuity lines in the multipatch decomposition, and we consider the different patches as the regions enclosed by these C^0 lines. In other words, our analysis-suitable B-spline multipatch surface models consist of a non-overlapping union of patches with superior internal regularity.



(a) (b)

Figure 3: Example of analysis-suitable B-spline multipatch surface models. The different patches are displayed in various colors. (a) A multipatch representation of two connected lattice cells in 2D, each with the same multipatch parameterization (composed of 8 patches). (b) A multipatch surface representation corresponding to a BCC lattice cell in 3D. The model is composed of different struts, with the surface of each strut further subdivided into six patches. Once again, the lines between the patches exhibit C^0-continuity. The control points can be seen around the node of the cell in the zoomed window on the right.

Let us consider $\mathrm{n}_{\mathrm{patches}}$ patches in our model. Each patch maps points from a parametric space, $\Omega^p_{\mathrm{param}}$, to form a shape in the physical space (following Eqs. (3) and (4)):

$$\Omega^p = \left\{ \mathbf{x}^p(\boldsymbol{\xi}), \forall\, \boldsymbol{\xi} \in \Omega^p_{\mathrm{param}} \right\}. \tag{23}$$

The associated multipatch B-spline shape is then created as:

$$\hat{\Omega} = \bigcup_{p=1}^{\mathrm{n}_{\mathrm{patches}}} \Omega^p. \tag{24}$$

Henceforth, any quantity denoted with the superscript p refers to the corresponding quantity defined for the patch Ω^p. For instance, each patch has a corresponding set of control points $\mathbf{X}^p$, a local basis $(\boldsymbol{A}_\xi{}^p, \boldsymbol{A}_\eta{}^p, \boldsymbol{A}_\gamma{}^p)$, a displacement field $\mathbf{u}^p$, etc.

During the deformation of a B-spline multipatch shape, it is essential to maintain (at least) C^0-continuity between the patches. From a practical point of view, since the different patches are conforming along their common edges in this work, C^0-continuity is imposed by using a connectivity matrix, denoted by $\mathbf{C}$. This sparse matrix embodies the transition from a vector of unique control points $\bar{\mathbf{X}} \in \mathbb{R}^{\mathrm{n}_{\mathrm{bf}} \times \mathrm{n}_{\mathrm{phy}}}$ to a vector containing duplicated control points at the common edges, $\hat{\mathbf{X}} \in \mathbb{R}^{\mathrm{n}_{\mathrm{bf}} \times \mathrm{n}_{\mathrm{phy}}}$:

$$\hat{\mathbf{X}} = \begin{pmatrix} \mathbf{X}^1 \\ \vdots \\ \mathbf{X}^{\mathrm{n}_{\mathrm{patches}}} \end{pmatrix} = \mathbf{C}\,\bar{\mathbf{X}} = \begin{pmatrix} \mathbf{C}^1 \\ \vdots \\ \mathbf{C}^{\mathrm{n}_{\mathrm{patches}}} \end{pmatrix} \bar{\mathbf{X}} \tag{25}$$

More precisely, $\mathbf{C}$ is a vertical stack of $\mathrm{n}_{\mathrm{patches}}$ local connectivity matrices $(\mathbf{C}^p)_{p\in[\![1,\mathrm{n}_{\mathrm{patches}}]\!]}$. Each $\mathbf{C}^p$ extracts the control points $\mathbf{X}^p$ for the p-th B-spline patch from the vector of all unique control points $\bar{\mathbf{X}}$ in the multipatch B-spline model. With this in hand, C^0-continuity during the deformation of the multipatch B-spline shape is easily maintained by applying the same operation to the displacement field $\mathbf{u}$:

$$\hat{\mathbf{U}} = \begin{pmatrix} \mathbf{U}^1 \\ \vdots \\ \mathbf{U}^{\mathrm{n}_{\mathrm{patches}}} \end{pmatrix} = \mathbf{C}\,\bar{\mathbf{U}} = \begin{pmatrix} \mathbf{C}^1 \\ \vdots \\ \mathbf{C}^{\mathrm{n}_{\mathrm{patches}}} \end{pmatrix} \bar{\mathbf{U}}, \tag{26}$$

where $\hat{\mathbf{U}}$ and $\bar{\mathbf{U}}$ are the displacement counterparts of $\hat{\mathbf{X}}$ and $\bar{\mathbf{X}}$, respectively.

3. Surface fitting method

Considering the geometric surface model presented earlier as the shape to update, we now provide a detailed description of the proposed isogeometric fitting approach. This method is designed to reposition the control points of the analysis-suitable B-spline multipatch surface to accurately fit tomographic image data. To aid understanding, we first outline the overall scheme of the method, which involves minimizing the sum of several carefully chosen functionals. Subsequently, we delve into the details of the approach, providing a mathematical description of each functional and explaining the minimization process.

3.1. An active contour-inspired method

The proposed method is inspired by the active contours method developed in the computer vision community (see, e.g., [32] for the origin on 2D images, and more recently [1, 64, 30] for applications to complex cases). It is based on the minimization of a sum of cost functions, also referred to as energies in related works, including the so-called image energy and internal energy. In our adaptation of the approach to measure shape defects in lattice structures from 3D CT scans, we employ specific and novel variants of these two energies. In particular, the internal energy is divided into intra-patch and inter-patch components. Additionally, to achieve better accuracy and solution convergence, we propose learning the graylevel transition profile in a small region surrounding the boundary of the object (similar idea as in [46]). To this end, a new set of design variables, denoted by $\boldsymbol{\rho} \in \mathbb{R}^{\mathrm{n}_\rho}$, is introduced alongside the surface shape control variables $\bar{\mathbf{U}}$ (see Eq. (26)). From a global point of view, the resulting optimization problem to be solved is formulated as follows:

$$(\bar{\mathbf{U}}^\star, \boldsymbol{\rho}^\star) = \underset{(\bar{\mathbf{U}},\boldsymbol{\rho})\in\mathbb{R}^{\mathrm{n_{bf}}\times\mathrm{n_{phy}}}\times\mathbb{R}^{\mathrm{n}_\rho}}{\operatorname{argmin}} \mathrm{E}_{\mathrm{image}}(\bar{\mathbf{U}}, \boldsymbol{\rho}) + \alpha\,\mathrm{E}_{\mathrm{intra}}(\bar{\mathbf{U}}) + \beta\,\mathrm{E}_{\mathrm{inter}}(\bar{\mathbf{U}}), \tag{27}$$

where α and β ($\in \mathbb{R}_+^*$) are weighting parameters that allow for adjusting the relative importance of $\mathrm{E}_{\mathrm{intra}}$ and $\mathrm{E}_{\mathrm{inter}}$ with respect to $\mathrm{E}_{\mathrm{image}}$.

- $\mathrm{E}_{\mathrm{image}}$: This energy is used to attract the geometry to the correct location in the image, i.e., in this work, to the surface boundary of the considered volumetric object. To achieve this accurately and efficiently, we propose building on the Virtual Image Correlation (VIC) functional, whose performance has been improved over the years for many applications in mechanics (see, e.g., [23, 31, 3, 22, 46]). In addition, let us note that the design variables gathered in $\boldsymbol{\rho}$, which control the graylevel transition profile, are learned through the minimization of this image functional.

- $\mathrm{E}_{\mathrm{intra}}$: This energy is used to regularize the problem within a B-spline patch by constraining some deformations of the geometry that are not affected by the previous image energy. In fact, the intrinsic regularization of B-spline functions may allow VIC to perform well in simple situations (e.g., 2D cases) and with (very) coarse meshes. The issue here is that, to accurately capture the fine shape defects we aim to measure in 3D, it is necessary to sufficiently refine the spline mesh. This refinement results in a design space of higher dimension, which can lead to the appearance of spurious tangential deformation modes on the surface (that preserve its geometry), thereby significantly degrading the conditioning of the inverse problem. Therefore, additional regularization is needed.

  In this respect, gradient or laplacian-based regularizations, which are commonly used in the active contour methods literature [1], appear to degrade the results in our case by penalizing out-of-plane displacements in

addition to in-plane displacements (see Section 4 for numerical illustrations). As a remedy, we propose using a regularization term that is more meaningful from a mechanical perspective: the membrane strain energy. This energy increases when the surface is subjected to tangential strains, such as in-plane shear and stretching, thereby regularizing in-plane deformations while leaving out-of-plane modes unaffected.

- $\mathrm{E}_{\mathrm{inter}}$: This final energy further regularizes the problem by ensuring a smooth transition of the updated shape at the interfaces between the patches, when appropriate with respect to the considered structure. It takes the form of a penalization term that ensures $\mathcal{C}^1$-continuity at specific interfaces between the patches.

For the developments that follow, let us introduce vector $\boldsymbol{D}$, which collects all the design variables of the optimization problem (27):

$$\boldsymbol{D} = \begin{pmatrix} \bar{\mathbf{U}} \\ \boldsymbol{\rho} \end{pmatrix}. \tag{28}$$

In practice, Problem (27) will result in a non-linear least-squares problem due to the image energy term (see Section 3.2 below). Consequently, it will be solved using a Gauss-Newton-type algorithm, in the sense that the Hessian matrix corresponding to the minimization of the image energy will be approximated using first-order derivatives of the image. The other energy terms will be quadratic with respect to $\bar{\mathbf{U}}$ (see Sections 3.3 and 3.4), so the exact (constant) Hessian matrix will be used for their minimization. Starting with initial guess $\boldsymbol{D}^{(0)}$, at iteration $n+1$, given the estimate $\boldsymbol{D}^{(n)}$, we seek $\boldsymbol{D}^{(n+1)} = \boldsymbol{D}^{(n)} + \delta\boldsymbol{D}^{(n+1)}$ by solving the following linear system:

$$\mathrm{H}^{\approx}_{\boldsymbol{D}}\mathrm{E}(\boldsymbol{D}^{(n)})\,\delta\boldsymbol{D}^{(n+1)} = -\nabla_{\boldsymbol{D}}\mathrm{E}(\boldsymbol{D}^{(n)}), \tag{29}$$

where E is the complete cost function:

$$\mathrm{E}(\boldsymbol{D}) = \mathrm{E}(\bar{\mathbf{U}}, \boldsymbol{\rho}) = \mathrm{E}_{\mathrm{image}}(\bar{\mathbf{U}}, \boldsymbol{\rho}) + \alpha\,\mathrm{E}_{\mathrm{intra}}(\bar{\mathbf{U}}) + \beta\,\mathrm{E}_{\mathrm{inter}}(\bar{\mathbf{U}}), \tag{30}$$

and $\nabla_{\boldsymbol{D}}\mathrm{E}(\boldsymbol{D}^{(n)})$ and $\mathrm{H}^{\approx}_{\boldsymbol{D}}\mathrm{E}(\boldsymbol{D}^{(n)})$ are, respectively, the gradient vector and approximated Hessian matrix of this complete cost function:

$$\nabla_{\boldsymbol{D}}\mathrm{E}(\boldsymbol{D}) = \nabla_{\boldsymbol{D}}\mathrm{E}(\bar{\mathbf{U}}, \boldsymbol{\rho}) = \nabla_{\boldsymbol{D}}\mathrm{E}_{\mathrm{image}}(\bar{\mathbf{U}}, \boldsymbol{\rho}) + \begin{pmatrix} \alpha\,\nabla_{\bar{\mathbf{U}}}\mathrm{E}_{\mathrm{intra}}(\bar{\mathbf{U}}) + \beta\,\nabla_{\bar{\mathbf{U}}}\mathrm{E}_{\mathrm{inter}}(\bar{\mathbf{U}}) \\ 0 \end{pmatrix}; \tag{31}$$

$$\mathrm{H}^{\approx}_{\boldsymbol{D}}\mathrm{E}(\boldsymbol{D}) = \mathrm{H}^{\approx}_{\boldsymbol{D}}\mathrm{E}(\bar{\mathbf{U}}, \boldsymbol{\rho}) = \mathrm{H}^{\approx}_{\boldsymbol{D}}\mathrm{E}_{\mathrm{image}}(\bar{\mathbf{U}}, \boldsymbol{\rho}) + \begin{pmatrix} \alpha\,\mathrm{H}_{\bar{\mathbf{U}}}\mathrm{E}_{\mathrm{intra}} + \beta\,\mathrm{H}_{\bar{\mathbf{U}}}\mathrm{E}_{\mathrm{inter}} & 0 \\ 0 & 0 \end{pmatrix}. \tag{32}$$

The different terms specified in Eqs (31)-(32) are detailed in the following sections.

3.2. Virtual Image Correlation cost function and minimization

As stated previously, we decided to rely on VIC for the image energy component of the approach. From a global point of view, the VIC method carried out in this work can be summarized as follows: the idea is to construct a virtual image, in the parametric space of the B-spline functions, that mimics the real image at the boundary of the object to be measured. Then, we seek the surface B-spline mapping that allows to interrogate the real image where it best matches the virtual image, i.e., in a thin region surrounding the actual boundary to be measured. An overview of the approach is depicted in Fig. 4. In what follows, we specify the strategy by elaborating on the theory behind this figure.

Step 1: creation of a virtual image.

The boundary to be measured in our setting corresponds to a piecewise constant implicit description of a surface in a volume. More precisely, moving along a normal to the surface in the real image, one encounters a black-to-white (or white-to-black) graylevel variation within a narrow voxel band (see the background in Fig. 4(Step 3)). Therefore, we propose to construct the virtual image using the profile shown in Fig. 5. This profile presents a smooth variation between a background color, labeled bg, and a foreground color, labeled fg. In practice, bg and fg are taken, respectively, as the mean background and foreground graylevels of the real image. The target transition profile is used to generate the virtual image along the normal direction to the surface, i.e., as a function of γ, and is furthermore considered invariant at every point on the surface, i.e., $\forall(\xi,\eta)$. In other words, the virtual image is defined as follows:

$$\mathrm{g} : \begin{array}{ccc} ([0,1]\times[0,1]\times[-h,h])\times]0,h] & \rightarrow & [\mathrm{bg},\mathrm{fg}] \\ (\xi,\eta,\gamma;\rho) & \mapsto & \mathrm{g}(\gamma;\rho) \end{array}. \tag{33}$$

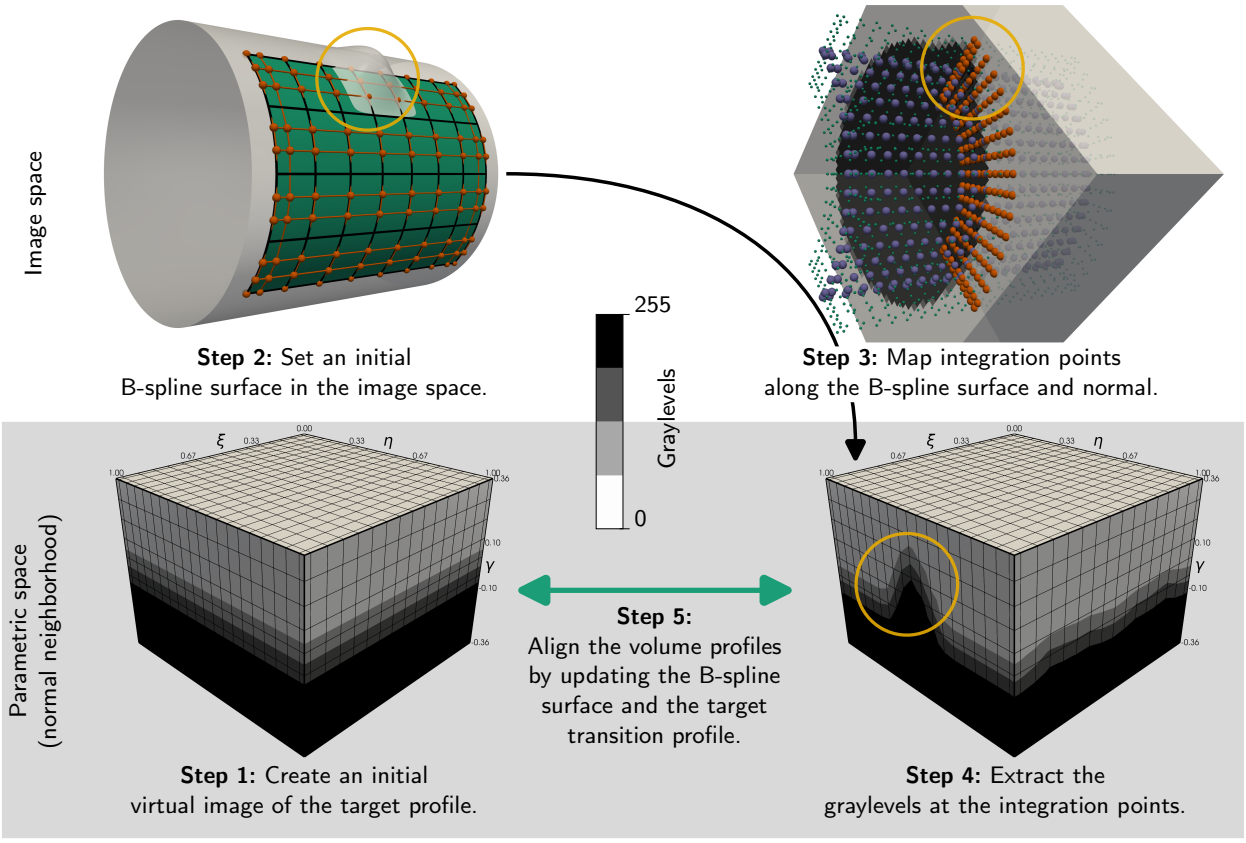


Figure 4: Overview of the VIC method carried out in this work for surface fitting. (Step 1) A virtual volumetric image is created to mimic the real image around the boundary to be captured. (Step 2) A B-spline surface mapping is defined to position and update the surface within the image space. (Step 3) Points are distributed in the surface neighborhood of the B-spline surface. (Step 4) These points are used to sample the real image in a thin region near the boundary to be measured, producing a representation of the boundary region of the real image in the parametric space. (Step 5) The objective is to adjust the spline mapping (Step 2) so that the surface neighborhood interacts with the image (Step 3) at points where the real (Step 4) and virtual (Step 1) profiles align. The orange-circled bump, identified as the primary source of error on this example, can be tracked throughout the steps of the process.

The parameters that control the shape of the virtual image are reduced here to a single value, $\rho \in]0, h]$. The function g is constant on the intervals $[-h, -\rho]$ and $[\rho, h]$, taking the values bg and fg, respectively. In the middle interval $[-\rho, \rho]$, the virtual image is a third-degree polynomial, chosen so that the entire function is C^1 on $[-h, h]$. This results in an odd function, centered on the point $(0, (\mathrm{fg} + \mathrm{bg})/2)$, with respect to γ, which is a desired property in VIC to ensure convergence of the algorithm [23]. Additionally, h must be sufficiently large to intersect the true boundary in the real image, which is necessary for the approach to converge. Finally, ρ is learned during the process in addition to the surface update, which ultimately yields a virtual image that better resembles the real image, thereby improving both the convergence of the algorithm and the accuracy of the obtained solution.

Remark 3.1. *For even more efficiency, h could be updated during the algorithm, meaning that the thickness of the surface neighborhood would be reduced as the method approaches the boundary to be measured. This is not implemented here, as the computational cost is already relatively low compared to the large amount of data being processed.*

Step 2: define and update a B-spline surface in the image space.

Of course, the analysis-suitable B-spline multipatch surface model along with its shape update, as detailed in Section 2, is used in Step 2 of Fig. 4. Since this model is defined in the physical coordinate system, denoted here $\mathcal{W}_{\mathrm{phy}}$, it now needs to be mapped into the image coordinate system, denoted $\mathcal{W}_{\mathrm{image}}$. This transformation is achieved using a

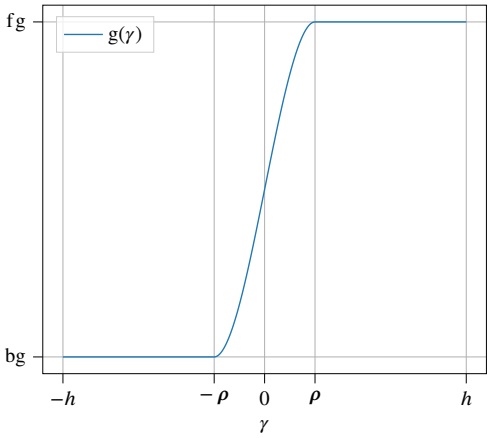


Figure 5: The virtual image function g embodies the target graylevel transition profile shown in the figure. Here, the smooth transition is learned by optimizing ρ to better fit the real profile, improving both the algorithm convergence and the accuracy of the obtained solution. Additionally, the constant regions at the periphery allow sufficient exploration of the real image, ensuring the algorithm convergence.

camera model. The camera model employed here is an affine mapping that accounts for a scaling factor $\mathrm{s_P}$, a translation vector $\boldsymbol{t_P}$, and three angles $(\theta_1, \theta_2, \theta_3)$ that define a rotation matrix $\boldsymbol{R_P}(\theta_1, \theta_2, \theta_3)$. More precisely, this camera model, labeled $\boldsymbol{P}$, is such that:

$$\boldsymbol{P} : \begin{array}{ccc} \mathcal{W}_{\mathrm{phy}} & \to & \mathcal{W}_{\mathrm{image}} \\ (x,y,z) & \mapsto & \boldsymbol{t_P} + \mathrm{s_P}\, \boldsymbol{R_P} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \end{array}. \tag{34}$$

The calibration of this camera model will be specified later in Section 4. Additionally, in our numerical experiments, the initial surface will correspond to that of the as-designed structure (i.e., the initially designed, perfect structure), which is assumed to be reasonably close to the as-manufactured part (i.e., including defects).

Step 3: map sampling points within the surface neighborhood in the image.

This step consists in distributing points to evaluate the real image in the surface neighborhood (see Eq. (17)). First, the points are sampled on the B-spline surface, and then they are extruded along the normal direction within the thickness interval $[-h, h]$. For the sake of simplicity and efficiency, the approximate mapping in Eq. (19) is used to move these points during the surface update in the algorithm. Of course, the camera model is again employed to transition from the physical to image coordinate system.

Step 4: extract the graylevels of the real image at the sampling points.

At this step, a subpixel interpolator is required to evaluate the graylevels at the (generally) non-integer voxel positions of the sampling points. For this purpose, a piecewise trilinear function that depends on the graylevels of the eight closest voxels is defined. It is denoted by:

$$\mathrm{f} : \begin{array}{ccc} \mathcal{W}_{\mathrm{image}} & \to & [0,255] \\ (x,y,z) & \mapsto & \mathrm{f}(x,y,z) \end{array}. \tag{35}$$

In the end, the evaluation of the real image in the surface neighborhood associated with patch p during the surface update is mathematically expressed as follows:

$$\mathrm{f} \circ \boldsymbol{P} \circ \mathbf{x}_{\mathrm{moved}}^{\mathrm{nn}\,p}(\xi,\eta,\gamma;\mathbf{U}^p), \quad \text{with} \quad \mathbf{x}_{\mathrm{moved}}^{\mathrm{nn}\,p}(\xi,\eta,\gamma;\mathbf{U}^p) \approx \mathbf{x}^{\mathrm{nn}\,p}(\xi,\eta,\gamma) + \mathbf{u}_{\mathrm{lin}}^{\mathrm{nn}\,p}(\xi,\eta,\gamma;\mathbf{U}^p). \tag{36}$$

In the above equation, $\mathbf{x}^{\mathrm{nn}\,p}$ is taken from Eq. (12), $\mathbf{u}_{\mathrm{lin}}^{\mathrm{nn}\,p}$ from Eq. (19), and $\circ$ refers to the composition operator between two applications.

Step 5: correlate the real image with the virtual one.

With all the previously introduced ingredients, the goal now is to seek the shape update of the B-spline surface (and of the virtual image profile) that best correlates the real image (evaluated in the surface neighborhood) with the virtual image. To this end, let us define the following VIC residual:

$$r^p(\xi,\eta,\gamma;\bar{\mathbf{U}},\boldsymbol{\rho}) = \mathrm{g}(\gamma;\boldsymbol{\rho}) - \mathrm{f}\circ\boldsymbol{P}\circ{\mathbf{x}^{\mathrm{nn}}_{\mathrm{moved}}}^p(\xi,\eta,\gamma;\mathbf{C}^p\,\bar{\mathbf{U}}), \quad \forall p\in\left\{1,2,\dots,\mathrm{n_{patches}}\right\}. \tag{37}$$

For the minimization, an L_2-norm of this graylevel residual is employed. Additionally, in this work, to prevent the discretization of the B-spline surface from affecting the results, it is proposed to integrate in the physical space (rather than in the parametric space, as it can be performed in VIC [31, 3, 46]). The VIC-type cost function used as the image energy in our approach is then expressed as follows:

$$\mathrm{E_{image}}(\bar{\mathbf{U}},\boldsymbol{\rho}) = \frac{1}{2}\sum_{p=1}^{\mathrm{n_{patches}}}\int_{\Omega^p}\frac{1}{2h}\int_{-h}^{h} {r^p}^2(\xi,\eta,\gamma;\bar{\mathbf{U}},\boldsymbol{\rho})d\gamma\,d\Omega^p \tag{38}$$

$$= \frac{1}{2}\sum_{p=1}^{\mathrm{n_{patches}}}\int_{\Omega^p_{\mathrm{param}}}\frac{1}{2h}\int_{-h}^{h} {r^p}^2(\xi,\eta,\gamma;\bar{\mathbf{U}},\boldsymbol{\rho})d\gamma\;{\boldsymbol{S_e}}^p(\xi,\eta)d\Omega^p_{\mathrm{param}}, \tag{39}$$

where ${\boldsymbol{S_e}}^p$, taken from Eq. (11), is the determinant of the Jacobian associated to the mapping from the parametric space (ξ,η) to the physical surface $\boldsymbol{\Omega}^p$ (see Eq. (23)).

In practice, the sampling points in Step 3 of the approach (see again Fig. 4) serve to integrate the VIC residual (using a simple rectangle quadrature rule) within the surface neighborhood. Consequently, we aim to place the sampling points on the surface so that there is approximately one per voxel, and we pay attention to take more points within the thickness (two or even three per voxel) to mimic the spatial quadrature that results from the formation of a voxel graylevel. Indeed, it is recognized that the accuracy of the integration rule in the thickness has a great impact on the method precision [23].

Then, to apply a Gauss-Newton scheme to minimize $\mathrm{E_{image}}(\bar{\mathbf{U}},\boldsymbol{\rho})$ from Eq. (39), it is required to compute the gradient of residual (37). The gradient is given by:

$$\nabla_{\boldsymbol{D}} r^p(\xi,\eta,\gamma;\bar{\mathbf{U}},\boldsymbol{\rho}) = \begin{pmatrix} -\left[\boldsymbol{J}_{\bar{\mathbf{U}}}\left(\mathrm{f}\circ\boldsymbol{P}\circ{\mathbf{x}^{\mathrm{nn}}_{\mathrm{moved}}}^p(\xi,\eta,\gamma;\mathbf{C}^p\,\bar{\mathbf{U}})\right)\right]^T \\ \nabla_{\boldsymbol{\rho}}\mathrm{g}(\gamma;\boldsymbol{\rho}) \end{pmatrix}, \tag{40}$$

where $\nabla_{\boldsymbol{\rho}}\mathrm{g}$ is the gradient of g with respect to ρ, and $\boldsymbol{J}_{\bar{\mathbf{U}}}\left(\mathrm{f}\circ\boldsymbol{P}\circ{\mathbf{x}^{\mathrm{nn}}_{\mathrm{moved}}}^p\right)$ is the Jacobian of $\mathrm{f}\circ\boldsymbol{P}\circ{\mathbf{x}^{\mathrm{nn}}_{\mathrm{moved}}}^p$ with respect to $\bar{\mathbf{U}}$. The latter reads:

$$\boldsymbol{J}_{\bar{\mathbf{U}}}\left(\mathrm{f}\circ\boldsymbol{P}\circ{\mathbf{x}^{\mathrm{nn}}_{\mathrm{moved}}}^p(\xi,\eta,\gamma;\mathbf{C}^p\,\bar{\mathbf{U}})\right) = \boldsymbol{J}\mathrm{f}(\boldsymbol{P}({\mathbf{x}^{\mathrm{nn}}_{\mathrm{moved}}}^p(\xi,\eta,\gamma;\mathbf{C}^p\,\bar{\mathbf{U}})))\left[\mathrm{s_P}\,\boldsymbol{R_P}\right]\boldsymbol{J}_{\mathbf{U}}{\mathbf{u}^{\mathrm{nn}\,p}_{\mathrm{lin}}}(\xi,\eta)\,\mathbf{C}^p, \tag{41}$$

where $\boldsymbol{J}\mathrm{f}$ is the Jacobian of f, taken from Eq. (35). Finally, with Eq. (40) in hand, the gradient of the VIC-type functional can be computed as follows:

$$\nabla_{\boldsymbol{D}}\mathrm{E_{image}}(\bar{\mathbf{U}},\boldsymbol{\rho}) = \sum_{p=1}^{\mathrm{n_{patches}}}\int_{\Omega^p_{\mathrm{param}}}\frac{1}{2h}\int_{-h}^{h} r^p(\xi,\eta,\gamma;\mathbf{C}^p\,.\,\bar{\mathbf{U}},\boldsymbol{\rho})\left[\nabla_{\boldsymbol{D}} r^p(\xi,\eta,\gamma;\mathbf{C}^p\,.\,\bar{\mathbf{U}},\boldsymbol{\rho})\right]d\gamma\;{\boldsymbol{S_e}}^p(\xi,\eta)d\Omega^p_{\mathrm{param}}, \tag{42}$$

and the corresponding approximate Hessian matrix for the Gauss-Newton algorithm is given by:

$$\mathrm{H}^{\approx}_{\boldsymbol{D}}\mathrm{E_{image}}(\bar{\mathbf{U}},\boldsymbol{\rho}) = \sum_{p=1}^{\mathrm{n_{patches}}}\int_{\Omega^p_{\mathrm{param}}}\frac{1}{2h}\int_{-h}^{h}\left[\nabla_{\boldsymbol{D}} r^p(\xi,\eta,\gamma;\mathbf{C}^p\,.\,\bar{\mathbf{U}},\boldsymbol{\rho})\right].\left[\nabla_{\boldsymbol{D}} r^p(\xi,\eta,\gamma;\mathbf{C}^p\,.\,\bar{\mathbf{U}},\boldsymbol{\rho})\right]^T d\gamma\;{\boldsymbol{S_e}}^p(\xi,\eta)d\Omega^p_{\mathrm{param}}. \tag{43}$$

Overall, the method proposed here for the image energy part may appear cheap compared to the large amount of data being processed. Indeed, only a surface problem (rather than a volumetric one) is solved in terms of the number of design variables, and the computations are restricted to a thin layer near the boundary, rather than being carried out over the entire volumetric image. Additionally, according to the literature, such a VIC-type functional should enable achieving accuracy below the voxel level [23, 22].

3.3. Intra-patch regularization with the membrane strain energy

As explained in Section 3.1, the above VIC-type problem becomes ill-posed when a sufficiently refined B-spline mesh is employed to capture local shape variations. As a remedy, we implement a weak (Tikhonov-like) regularization scheme, which involves complementing the image energy functional with, firstly, a regularization energy that acts within each patch of our complete model. To achieve this robustly and efficiently in 3D, we propose using the membrane strain energy of the surface, which constrains tangential deformation modes while allowing out-of-plane displacements, enabling the surface to accommodate defects in the 3D CT scan. This mechanical energy is computed here by attaching a linear elastic isogeometric shell model to the surface and considering only the membrane contribution. In the following, we present the main ingredients for computing this energy. For more information, interested readers are encouraged to consult the literature on isogeometric shells, for instance [34, 21, 9].

From a global point of view, the membrane strain energy (in the context of small deformations) associated with patch p of the initial surface formally reads:

$$\mathscr{E}^p_{\text{intra}}(\mathbf{U}^p) = \frac{1}{2}\int_{\Omega^p} t\ \boldsymbol{e}^p(\xi,\eta;\mathbf{U}^p)^T\ \ \boldsymbol{H}^p(\xi,\eta)\ \ \boldsymbol{e}^p(\xi,\eta;\mathbf{U}^p)\ d\Omega^p, \tag{44}$$

where t is the shell thickness, $\boldsymbol{e}^p$ the membrane strain tensor and $\boldsymbol{H}^p$ the Hooke's tensor. Voight notation is used in the above equation, i.e., $\boldsymbol{e}^p$ is a vector and $\boldsymbol{H}^p$ a matrix. These two operators are typically defined in the contravariant basis of the surface, which is not detailed here for the sake of conciseness (again, see, e.g., [34, 21, 9] for more details). More precisely, the membrane strain tensor is given by:

$$\boldsymbol{e}^p(\xi,\eta;\mathbf{U}^p) = \begin{pmatrix} \frac{\partial \mathbf{u}^p}{\partial \xi}(\xi,\eta;\mathbf{U}^p).\ \boldsymbol{A}^p_\xi(\xi,\eta) \\ \frac{\partial \mathbf{u}^p}{\partial \eta}(\xi,\eta;\mathbf{U}^p).\ \boldsymbol{A}^p_\eta(\xi,\eta) \\ \frac{1}{2}\left[\frac{\partial \mathbf{u}^p}{\partial \xi}(\xi,\eta;\mathbf{U}^p).\ \boldsymbol{A}^p_\eta(\xi,\eta) + \frac{\partial \mathbf{u}^p}{\partial \eta}(\xi,\eta;\mathbf{U}^p).\ \boldsymbol{A}^p_\xi(\xi,\eta)\right] \end{pmatrix}, \tag{45}$$

where $\mathbf{u}^p$ represents the surface displacement field, taken from Eq. (6), and $\boldsymbol{A}^p_\xi$ and $\boldsymbol{A}^p_\eta$ are the tangent vectors of the covariant basis, see Eqs. (8) and (9). Since we are in the context of small deformations, the membrane strain depends linearly on the design variables $\mathbf{U}^p$, and can therefore be simply expressed using the (constant) Jacobian of $\boldsymbol{e}^p$ with respect to $\mathbf{U}^p$:

$$\boldsymbol{e}^p(\xi,\eta;\mathbf{U}^p) = \boldsymbol{J}_{\mathbf{U}^p}\boldsymbol{e}^p(\xi,\eta)\ \ \mathbf{U}^p, \tag{46}$$

where the Jacobian matrix is given by:

$$\boldsymbol{J}_{\mathbf{U}^p}\boldsymbol{e}^p(\xi,\eta) = \begin{pmatrix} \boldsymbol{A}^{pT}_\xi \frac{\partial \mathbf{N}^{3Dp}}{\partial \xi} \\ \boldsymbol{A}^{pT}_\eta \frac{\partial \mathbf{N}^{3Dp}}{\partial \eta} \\ \frac{1}{2}\left[\boldsymbol{A}^{pT}_\xi \frac{\partial \mathbf{N}^{3Dp}}{\partial \xi} + \boldsymbol{A}^{pT}_\eta \frac{\partial \mathbf{N}^{3Dp}}{\partial \eta}\right] \end{pmatrix}. \tag{47}$$

Next, the Hooke's tensor considered here corresponds to a material with a Poisson's ratio of 0, as the Poisson effect is not relevant to our surface fitting problem. Additionally, we assume a Young's modulus of 1 and a shell thickness $t = 1$, so that the scaling factor of the overall stiffness will be fully represented by the weighting factor α (see Eq. (27)). As a result, the following symmetric matrix for the Hooke's tensor is obtained:

$$\boldsymbol{H}^p(\xi,\eta) = \begin{pmatrix} \mathrm{A}^{\xi\xi p}(\xi,\eta)\mathrm{A}^{\xi\xi p}(\xi,\eta) & \mathrm{A}^{\xi\eta p}(\xi,\eta)\mathrm{A}^{\xi\eta p}(\xi,\eta) & \mathrm{A}^{\xi\xi p}(\xi,\eta)\mathrm{A}^{\xi\eta p}(\xi,\eta) \\ \star & \mathrm{A}^{\eta\eta p}(\xi,\eta)\mathrm{A}^{\eta\eta p}(\xi,\eta) & \mathrm{A}^{\eta\eta p}(\xi,\eta)\mathrm{A}^{\xi\eta p}(\xi,\eta) \\ \star & \star & \frac{1}{2}\mathrm{A}^{\xi\xi p}(\xi,\eta)\mathrm{A}^{\eta\eta p}(\xi,\eta) + \frac{1}{2}\mathrm{A}^{\xi\eta p}(\xi,\eta)\mathrm{A}^{\xi\eta p}(\xi,\eta) \end{pmatrix}, \tag{48}$$

where $\mathrm{A}^{\xi\xi p}$, $\mathrm{A}^{\eta\eta p}$, and $\mathrm{A}^{\xi\eta p}$ are related to the contravariant basis. These coefficients are derived from the covariant basis by computing the symmetric matrix $\boldsymbol{A}^p$ such that:

$$\boldsymbol{A}^p = \begin{pmatrix} \mathrm{A}^{\xi\xi p} & \mathrm{A}^{\xi\eta p} & \mathrm{A}^{\xi\gamma p} \\ \star & \mathrm{A}^{\eta\eta p} & \mathrm{A}^{\eta\gamma p} \\ \star & \star & \mathrm{A}^{\gamma\gamma p} \end{pmatrix} = \begin{pmatrix} \| \boldsymbol{A}^p_\xi \|^2 & \boldsymbol{A}^p_\xi . \boldsymbol{A}^p_\eta & 0 \\ \star & \| \boldsymbol{A}^p_\eta \|^2 & 0 \\ \star & \star & 1 \end{pmatrix}^{-1}. \tag{49}$$

Now, substituting the terms (45)-(49) into Eq. (44), the overall intra-patch regularization energy can be computed as follows:

$$\mathrm{E}_{\mathrm{intra}}(\bar{\mathbf{U}}) = \frac{1}{2}\,\bar{\mathbf{U}}^T\,\boldsymbol{K}\,\bar{\mathbf{U}}, \tag{50}$$

where the stiffness matrix $\boldsymbol{K}$ is given by:

$$\boldsymbol{K} = \sum_{p=1}^{\mathrm{n_{patches}}} \mathbf{C}^{pT} \left[\int_{\Omega^p} \boldsymbol{J}_{\mathbf{U}^p} \boldsymbol{e}^p(\xi,\eta)^T \;\; \boldsymbol{H}^p(\xi,\eta)\; \boldsymbol{J}_{\mathbf{U}^p} \boldsymbol{e}^p(\xi,\eta) d\Omega^p \right] \mathbf{C}^p \tag{51}$$

$$= \sum_{p=1}^{\mathrm{n_{patches}}} \mathbf{C}^{pT} \left[\int_{\Omega^p_{\mathrm{param}}} \boldsymbol{J}_{\mathbf{U}^p} \boldsymbol{e}^p(\xi,\eta)^T \;\; \boldsymbol{H}^p(\xi,\eta)\; \boldsymbol{J}_{\mathbf{U}^p} \boldsymbol{e}^p(\xi,\eta)\;\; \boldsymbol{S_e}^p(\xi,\eta) d\,\Omega^p_{\mathrm{param}} \right] \mathbf{C}^p\,. \tag{52}$$

To compute these integrals, a standard Gauss quadrature scheme with $(p+1)^2$ points per surface spline element of degree p is employed, as is classical in IGA [29, 13]. Finally, the derivatives of the intra-patch energy are straightforward, leading to the following expressions for the gradient and Hessian:

$$\nabla_{\bar{\mathbf{U}}}\,\mathrm{E}_{\mathrm{intra}}(\bar{\mathbf{U}}) = \boldsymbol{K}\,\bar{\mathbf{U}}\;; \tag{53}$$

$$\mathrm{H}_{\bar{\mathbf{U}}}\,\mathrm{E}_{\mathrm{intra}} \quad = \boldsymbol{K}. \tag{54}$$

Remark 3.2. *The use of the membrane strain energy in the context of small deformations has proven effective throughout our numerical experiments, which aim to deform the initial, as-designed surface to fit shape defects in lattice structures. Indeed, the levels of membrane strain in these cases are limited, making it possible to find a range for the weighting factor α that allows to regularize the problem while ensuring good solution accuracy. However, in the opposite case where the membrane strain would become excessive (due to very large deformations), it would still be possible to update the surface Ω^p in the algorithm at a certain point, and recompute all the operators, thus including the proposed intra-patch regularization energy, for this new configuration.*

3.4. Inter-patch regularization with C^1 penalization at patch interfaces

At this step, it remains to regularize the surface fitting problem at the interfaces between the patches. Indeed, up to this point, the transitions between the patches of our B-spline surface are only C^0-continuous, as explained in Section 2.3. As a result, kinks may appear between patches when modifying the shape of the multipatch surface, even though there is often no specific reason for them to occur precisely at those inter-patch locations.

To address this issue, we introduce an additional penalization term through $\mathrm{E}_{\mathrm{inter}}$ in Eq. (27) to ensure that the transitions between certain patches, which are initially of C^1-continuity, remain so during the shape modification process. Leveraging the properties of B-spline functions, this can be achieved by ensuring that the boundary control points, along with their two neighboring points within each patch, are aligned and equally spaced (see Fig. 6a for illustration). The term $\mathrm{E}_{\mathrm{inter}}$ can then be expressed as:

$$\mathrm{E}_{\mathrm{inter}}(\bar{\mathbf{U}}) = \frac{1}{2} \sum_{(a,b,c)} \sum_{s\in\{x,y,z\}} \left(P_a^s - 2P_b^s + P_c^s \right)^2, \tag{55}$$

where (P_a, P_b, P_c) represent the triplets of control points between the patches for which we want to ensure that the transition remains C^1-continuous during the shape modification (see again Fig. 6a for the notations). From a practical point of view, $\mathrm{E}_{\mathrm{inter}}$ can be computed as follows:

$$\mathrm{E}_{\mathrm{inter}}(\bar{\mathbf{U}}) = \frac{1}{2} \left(\bar{\mathbf{X}} + \bar{\mathbf{U}} \right)^T \boldsymbol{M} \left(\bar{\mathbf{X}} + \bar{\mathbf{U}} \right), \tag{56}$$

where the (constant) matrix $\boldsymbol{M}$ is given by:

$$[\boldsymbol{M}]_{ij} = \sum_{(a,b,c)} \sum_{s\in\{x,y,z\}} \left(\delta_{a_s,i} - 2\delta_{b_s,i} + \delta_{c_s,i} \right) \left(\delta_{a_s,j} - 2\delta_{b_s,j} + \delta_{c_s,j} \right), \tag{57}$$

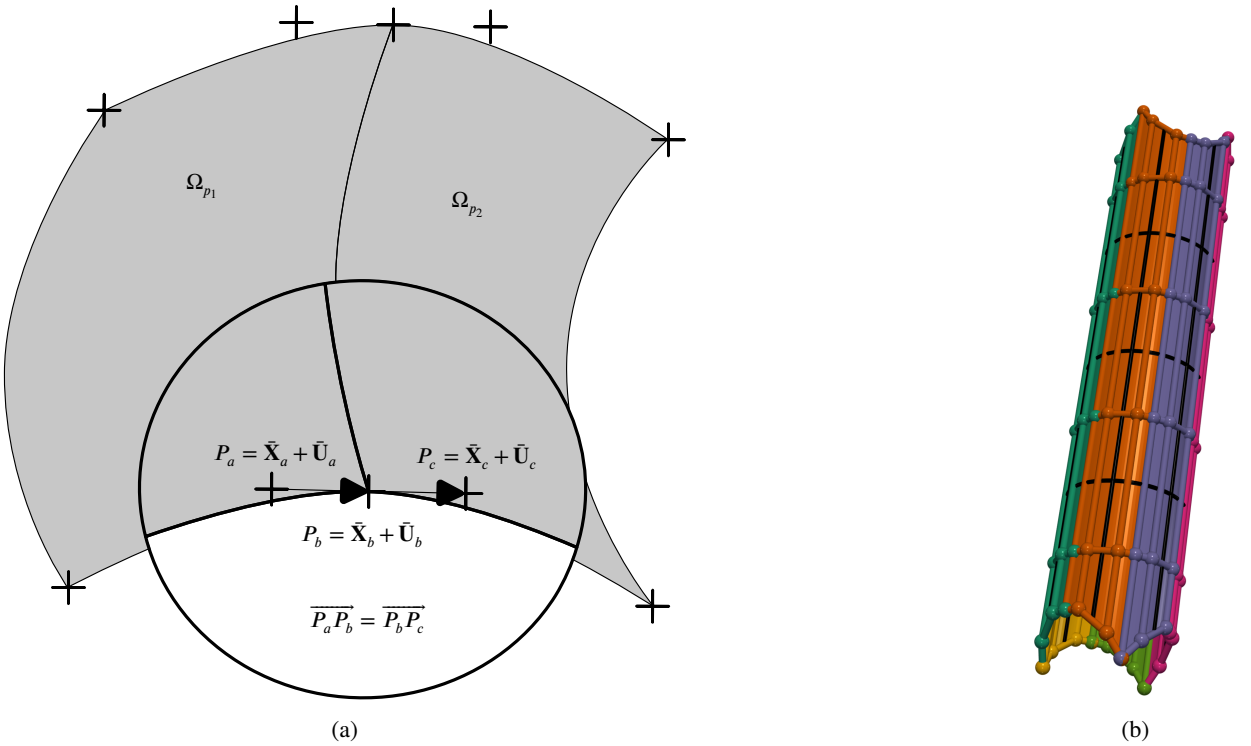


Figure 6: (a) Illustration of the condition to ensure a C^1-continuous transition between two patches. (b) Example of the initial, as-designed surface associated with a lattice strut, taken from Fig. 3b. Note that particular care has been taken to align and equally space the control points on each side of the patch interfaces (except at the strut boundaries where C^0 continuity is necessary to connect the different struts of the lattice structure). Consequently, this condition can be maintained during the shape update to preserve C^1-continuity.

using the Kronecker symbol $\delta_{i,j} = \begin{cases} 1 \text{ for } i=j, \\ 0 \text{ for } i \neq j. \end{cases}$. Finally, we can easily compute the gradient vector and the associated Hessian matrix to minimize this functional. It yields:

$$\nabla_{\bar{\mathbf{U}}} \mathrm{E}_{\mathrm{inter}}(\bar{\mathbf{U}}) = \left(\bar{\mathbf{X}} + \bar{\mathbf{U}}\right)^T \boldsymbol{M} \; ; \tag{58}$$

$$\mathrm{H}_{\bar{\mathbf{U}}} \mathrm{E}_{\mathrm{inter}} \quad = \boldsymbol{M} \; . \tag{59}$$

This concludes the method section, as we now have all the necessary quantities to successfully implement the approach provided in Eqs. (29)-(32).

Remark 3.3. *Note that this inter-patch treatment requires the initial multipatch surface to respect the C^1-condition illustrated in Fig. 6a at specific locations. Therefore, we make an effort, when feasible, to align and equally space the control points on each side of the patch interfaces in the initial, as-designed surface model (see Fig. 6b, which model will be utilized in the forthcoming examples section).*

4. Method validation

4.1. Synthetic case

To evaluate the performance of the proposed method, a first synthetic data-set was considered. It consists in an image of a cylindrical strut generated with a known defect. More precisely, the closed form expression of the indicator function of the imperfect strut is given in the cylindrical coordinate system as the set of points (r, ϕ, z) which satisfy:

$$1_{\mathrm{strut}}(r, \phi, z) = \begin{cases} 255 & \text{if } r < R(z, \phi) \text{ and } |z| < H, \\ 0 & \text{otherwise} \end{cases} \tag{60}$$

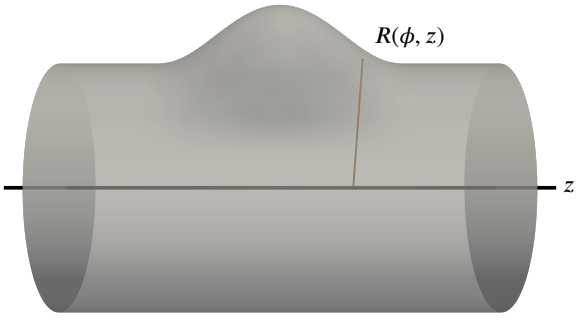
with:

$$R(z,\phi) = \begin{cases} R + \frac{R}{2}\sin^2\left(\frac{z+z_0}{2z_0}\pi\right)\sin^2(2\phi) & \text{if } (\phi, z) \in [0, \pi/2] \times [-z_0, z_0] \\ R & \text{otherwise} \end{cases} . \tag{61}$$

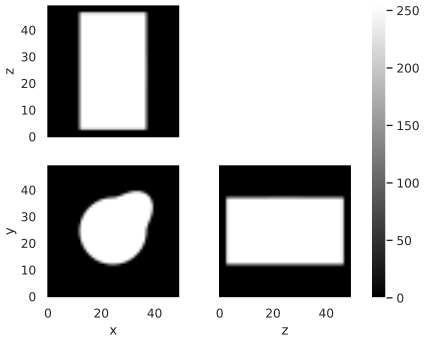
In the following, the parameters $R = 1$, $z_0 = 1$ and $H = 1.75$ were used.

To generate a graylevel image from this closed form, a regular grid of 750^3 sampling points was generated in a $[-2, 2]^3$ cube. The indicator function (60) was evaluated at the sampling points to generate a first 750^3 black/white image. From this binary image, a graylevel image of lower resolution was obtained by binning 15^3 neighboring voxels. This low-resolution graylevel image was used as a synthetic image of the as-manufactured geometry. This provides a synthetic data-set with corresponding ground truth to evaluate the performance of the developed isogeometric fitting method.

The image is shown in Fig. 7. Fig. 7a provides an illustration of the indicator function, and Fig. 7b depicts the resulting synthetic graylevel volumetric image (in terms of orthogonal slices). Note that the size of a voxel with respect to the diameter of the strut has been chosen such that the resolution is similar to that of the real tomographic images which will be taken in Section 4.2. More precisely, since the resolution of the real tomographic images was about 20 μm/voxel and the size of the real lattice struts was approximately 1 mm, the cylinder diameter was set to 25 voxels.

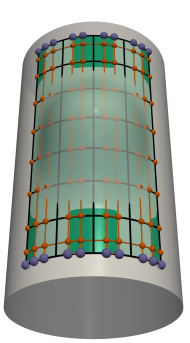


(a) Indicator function of the synthetic domain with sine imperfection.

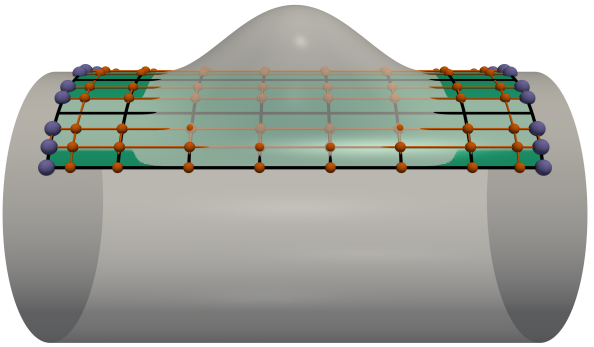


(b) Orthogonal slices of the corresponding synthetic graylevel volumetric image (voxel sclale).

Figure 7: Synthetic volumetric image of a strut with an imperfection created from a closed form.



(a) Top view.



(b) Side view.

Figure 8: B-spline patch representing the as-designed geometry with nominal initial position. The element edges are shown in black and the control mesh in orange. The blue control points at the top and bottom of the cylinder are fixed in the z-direction (i.e., along the cylinder axis).

Since the method is intended to update an initial as-designed CAD model, a defect-free model of a perfect quarter cylinder with a diameter of R was created to initially represent the region where the defect occurs, as shown in Fig 8. More specifically, a single-patch cubic B-spline surface consisting of 10×10 control points was used, so in this example no inter-patch energy E_{inter} had to be included. The top and bottom edges of the cylindrical surface were constrained, i.e. the shape correction field $\bar{\mathbf{U}}$ in the z-direction for the corresponding control points (shown in blue in Fig. 8) was set to 0. This constraint reduced the number of DOFs from 300 to 280.

The virtual image correlation procedure proposed in Section 3.2 and illustrated in Fig.4 was applied to this synthetic data-set.

Step 1: Generation of the virtual image. Given the way the synthetic image was generated, the graylevels of the background bg and the foreground fg were set to 0 and 255 respectively. The parameter controlling the thickness of the transition from white to black was initialized to $\rho = 2.5$ voxels.

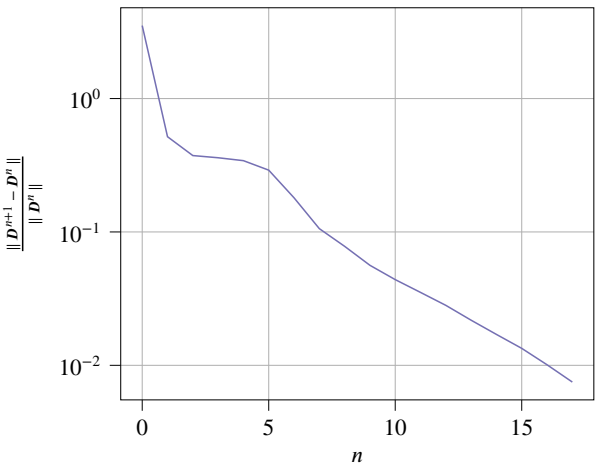
Step 2: In this synthetic example, the calibration of the transformation between the physical and image coordinate systems was considered known. In other words, the parameters of the camera model were not estimated but simply set to their analytical values: $s_P = 12.5$, $\boldsymbol{t_P} = (1.96, 1.96, 1.96)^T$ and no rotations, i.e. $\boldsymbol{R_P} = \boldsymbol{I}_3$.

Step 3: The size h of the virtual image in the normal direction to the surface was chosen so that the layer of the surface neighborhood in the image (physical) space was 10 voxels thick. The sampling of the surface neighborhood was designed to provide approximately 4 points per voxel along the surface and 2 points per voxel along the normal.

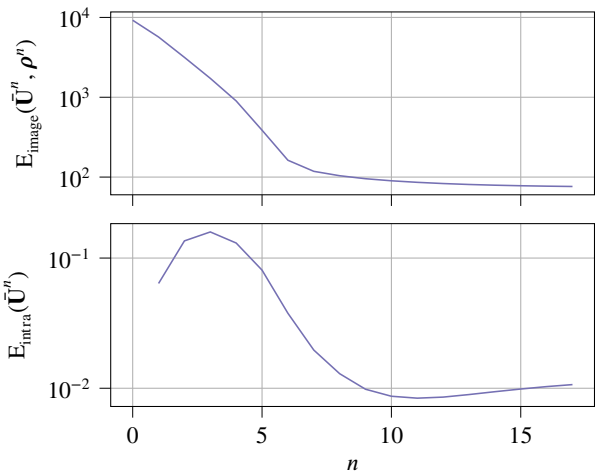
Remark 4.1. *These sampling parameters were used to produce smooth images for the purposes of this article, but in practice, it is possible to use coarser samplings producing the same level of accuracy at a lower computational cost. For example, choosing* 0.75 *points per voxel along the surface and* 1 *point along the normal allowed the algorithm to converge to a quasi-identical solution.*

Step 4: The gray-level value at these sampling points were evaluated using sub-pixel interpolation. From this, it is possible to generate an image representing the transition corresponding to the real boundary of the material in the parametric space, as shown in Fig. 4(step 4).

Step 5: The position of the control points was then updated to minimize the functional E (see Eq. (30)). As mentioned previously, an intra-patch regularization based on the membrane energy of the surface was used to avoid spurious in-plane movements while leaving the out-of-plane components free. In this case, the product of stiffness and membrane thickness was set to $\alpha = 2 \cdot 10^{-2} \times 255^2$. Additionally, we recall that since it is a single-patch model here, there is no need for inter-patch regularization. In practice, at each iteration of the algorithm, the vector of design variables $\boldsymbol{D}$, which gathers the control point displacements $\bar{\mathbf{U}}$ and the graylevel transition parameter ρ of the virtual image, was updated using (29).



(a) Convergence graph of the surface fitting algorithm for the synthetic cylinder case.



(b) Evolution of both the virtual image correlation cost function and the membrane strain energy.

Figure 9: Convergence of the algorithm (a) and evolution of the terms of the cost function (b) over the iterations.

Fig. 9a shows the evolution of the relative stagnation indicator of the vector of design variables $\boldsymbol{D}$ as a function of the iterations. The algorithm converged within 18 iterations to reach a relative increment smaller than 1%. The convergence rate is of the order of 1.3, which is slower than the quadratic convergence of a Newton, but still very efficient. This is attributed to the Gauss-Newton approximation, but also possibly to the discrete nature of digital images. Then, Fig. 9b illustrates the interaction between the two parts of the cost function. Initially, with no deformation, the membrane energy was zero. As long as the deformation remains low (up to iteration 3), the reduction in image energy seems to take priority. As the spline deforms to match the shape of the image, the membrane energy increases. From iteration 3 to iteration 11, the descent directions seek to minimize both energies. From iteration 11 onwards, the membrane energy is very low and only the image energy continues to decrease until the corrections are sufficiently small, indicating that the algorithm has converged.

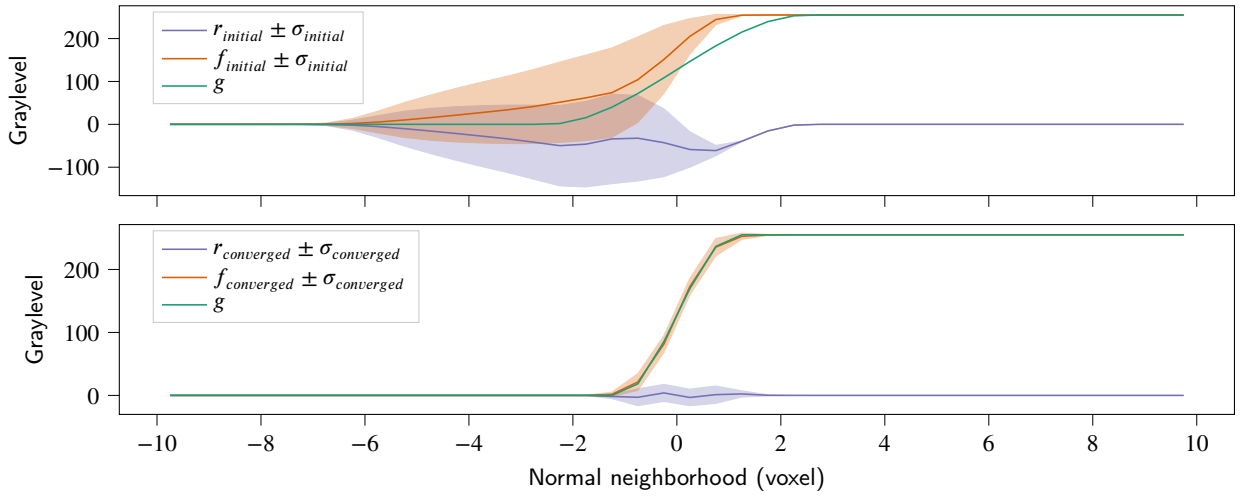
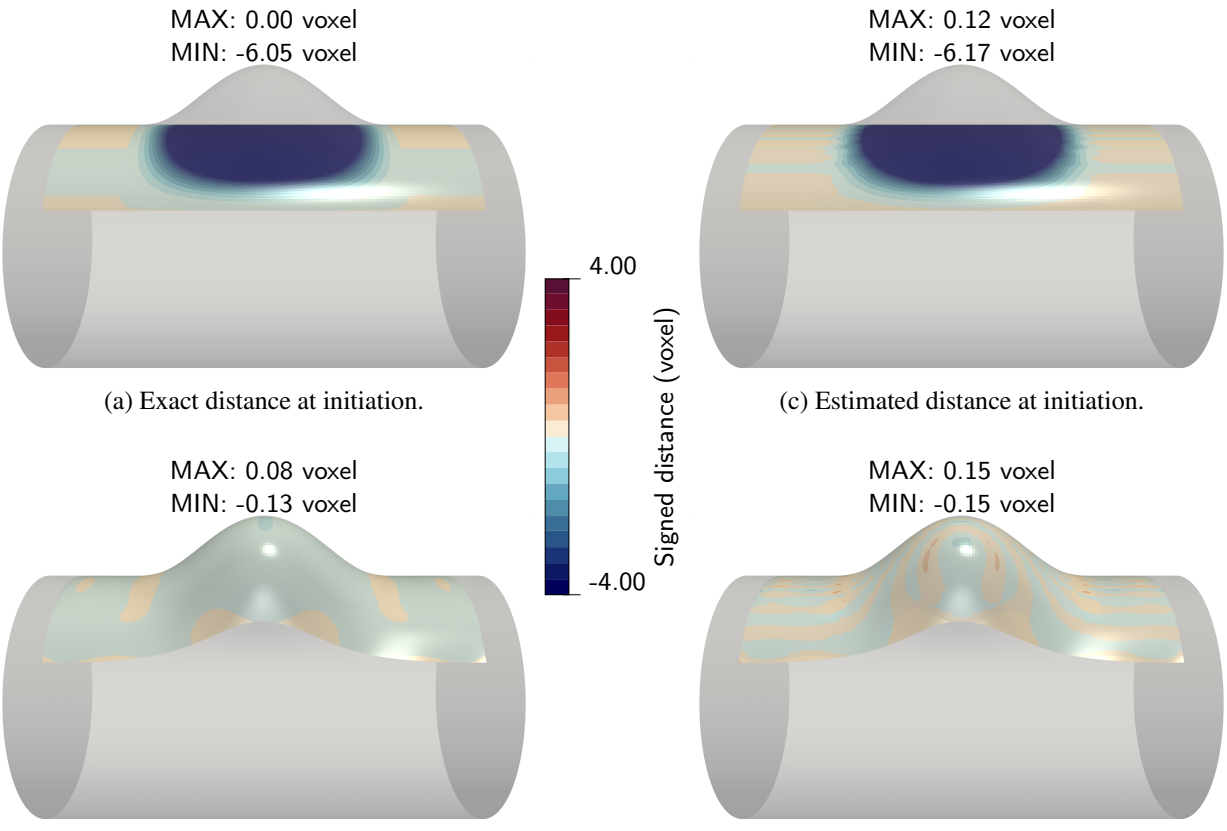


Figure 10: Statistical comparison of the graylevel profiles observed initially and at convergence, on the real image, virtual image, and virtual image correlation residual. The proposed algorithms efficiently captures the black-to-white transition of the real image in addition to the surface geometry.

Let us recall that the image correlation residual between virtual and real images is a 3D scalar function in a parametric space. More specifically, it is made of the cross product of the 2D spline parametric space of the surface and the identity mapping in the out-of plane dimension, as shown in Fig. 4 (steps 1 and 4). To illustrate the capacity of the algorithm to estimate the real black-to-white transition, it is proposed to average the real image along the (ξ, η) directions. It gives a 1D function in the out-of-plane direction representative of the average on the surface of the real black-to-white transition to be compared to the virtual image of Fig. 5. Fig. 10 shows the virtual, initial and optimized transitions along with the VIC residual within the search width $[-h, h]$. Initially, the graylevels of the image (orange) are far from the virtual image (green). As the B-spline surface moves and the virtual image is adjusted, the image becomes closer to the virtual image. The initially high standard deviation of the residual (blue) and that of the real image (orange) indicates a high level of uncertainty, testifying to a rather poor description of the surface geometry and the black-to-white evolution. After convergence, this uncertainty is highly reduced, which demonstrates that the algorithm is able to estimate both the position of the surface and the transition profile. The fact that the standard deviation is not zero at convergence is due to voxelization that the regularity of B-splines smooths out. Additionally, the zero residual outside the interval $[-6, 6]$, even initially, suggests that a smaller search width could have been used to accelerate computation.

Given that the exact geometry is known here, a distance map between the B-spline surface and the actual transition position in the image is generated. As shown in Fig. 11(b), the converged geometry deviates by 0.13 voxels at its furthest point from the exact geometry, compared to an initial maximum distance of 6.05 voxels. This distance map reflects sub-voxel errors arising from both voxelization and surface fitting processes. One advantage of this surface fitting process is that the B-spline representation of the geometry, combined with the incorporation of membrane energy, effectively smooths out the high-frequency angular shape of the voxels.



(a) Exact distance at initiation.

(c) Estimated distance at initiation.

(b) Exact distance at convergence.

(d) Estimated distance at convergence.

Figure 11: Signed distance of the B-spline surface to the geometry. (a) and (b) show the initial and converged surfaces colored according to the signed distance to the exact ground-truth geometry. (c) and (d) show the initial and converged surfaces colored by the signed distance to the real geometry estimated using voxel values only. The strong similarity of these distance maps confirms the reliability of the proposed 1D-DIC distance estimation method as an effective indicator of exact distance to the real surface.

However, this distance is not available in real cases because the exact geometry is not known. The only error available with this method is the VIC residual. But this is a graylevel residual, not a distance. In the following, a new distance map estimation method is proposed solely based on graylevels. This distance is computed through a one-dimensional digital image correlation (DIC) algorithm [28] between the virtual image and the graylevel profile along the normal, using the same discretization as for the virtual image correlation functional. This method involves minimizing a DIC residual with respect to a scalar displacement d at every point on the surface:

$$d^\star(\xi,\eta) = \underset{d\in[-h,h]}{\operatorname{argmin}}\, r_{DIC}(\xi,\eta;d). \tag{62}$$

With the DIC residual computed using the tools from the virtual image correlation cost function:

$$r_{DIC}(\xi,\eta;d) = \int_{-h}^{h} \mathrm{g}(\gamma-d;\boldsymbol{\rho}) - \mathrm{f}(\boldsymbol{P}(\mathbf{x}^{\mathrm{nn}}_{\mathrm{moved}}(\xi,\eta,\gamma;\mathbf{U}))). \tag{63}$$

This method yields a distance in the physical space that can be converted to voxels. This distance in voxels is compared to the real distance in Fig. 11 at the initial and converged states. In this figure, comparing (a) to (c) shows that the estimated distance provides a very good approximation of the exact distance. The discrepancy between the real and estimated distances is always significantly sub-voxel. To the best of our knowledge, such a distance estimation method is new and represents a further contribution to the field, which could also be advantageously used, for instance, to initialize or adjust the neighborhood thickness h in challenging configurations.

In the following, we study the effect of the internal regularization E_{intra} on the solution. To this end, we compare different levels of regularization α and different regularization energies. The membrane energy proposed in this paper is compared to the gradient $\|\nabla \mathbf{u}\|^2$ and Laplacian $\|\Delta \mathbf{u}\|^2$ energies classically encountered in the fields of active contours and image registration. In order to compare regularization levels with regularizations of different nature and order, we adjusted α by the ratio of VIC and internal regularization energies calculated for a white noise-type displacement field. The results are shown in Fig 12. The resulting shapes and distance mappings for the different methods are presented in

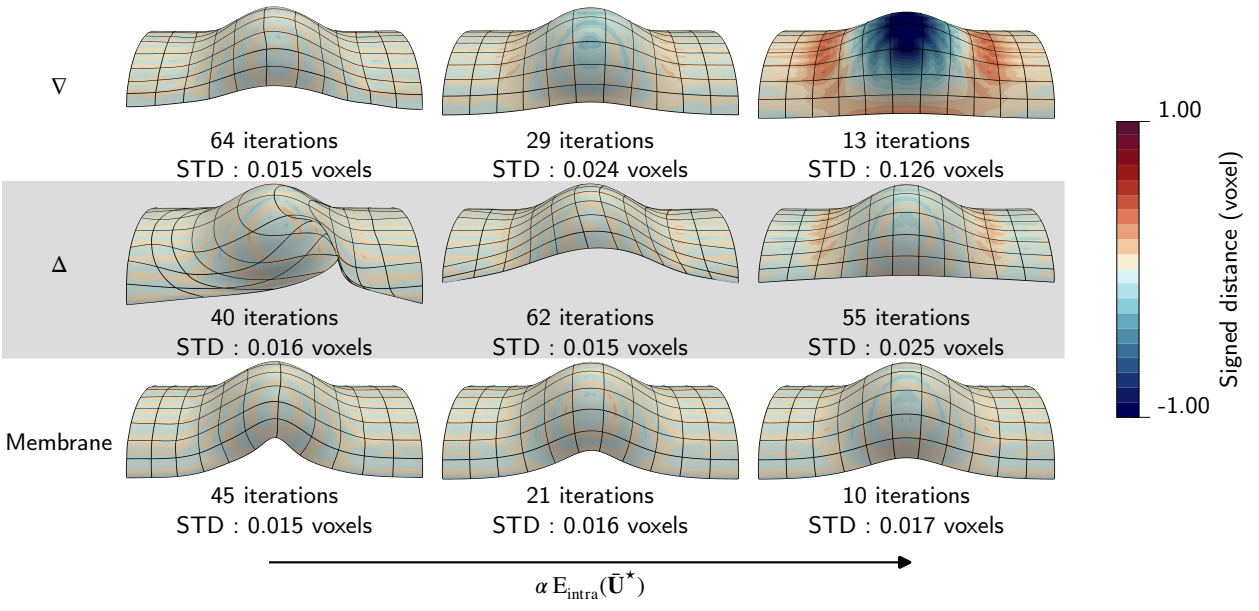


Figure 12: Comparison of the different methods of regularization.

line. From left to right, the penalty parameter is increased. It can be seen that the membrane regularization method is the only consistent one, i.e. it is robust to large variations in α and provides a low number of iterations which reduces as the regularization increases, the functional becoming more convex.

Thanks to this synthetic data-set, it has been possible to quantify the performance of the proposed method. However, working with real lattice images adds a host of additional complexities such as image noise, reconstruction artifacts, estimation of the unknown transformation between physical and image coordinate systems (e.g., camera model calibration) and complex multipatch geometry, which are the subjects of the next section.

4.2. Case of a real lattice unit-cell

To design a more realistic test case to challenge the method, the workflow described in Fig. 13 was followed. The figure is divided into two parts. The top row of Fig. 13 corresponds to the initial as-designed CAD geometry whereas the bottom one corresponds to the generation of an as-manufactured image of the same lattice but with representative geometric defects.

The as-designed geometry of this lattice structure consists of the $3 \times 3 \times 4$ repetition of the Body Centered Cubic (BCC) unit cell. As shown in the middle top image of Fig. 13, the BCC cells are composed of 8 beams of circular cross-section arranged along the 4 long diagonals of a cube and intersecting at its center. In order to create an analysis-suitable multipatch B-spline volume description of this cell, each beam is composed of 6 volume patches. Each patch is $1/6$ of a circle in cross-section with vertices made so that the patches fit together. One of the $3 \times 3 \times 4$ BCC cells of the lattice structure is extracted and shown in orange in Fig. 13. From the analysis-suitable multipatch parametrization of the volume, a multipatch parametrization of the external surface suitable to the proposed fitting method must be created. This step was conveniently done taking advantage of the tensor-product nature of the B-spline volume. Since the volume is analysis-suitable, the surface is also analysis-suitable using the connectivity of the volume.

The other branch of the workflow aims at designing a representative as-manufactured image, i.e. containing all the difficulties inherent to real lattice CT scans: image noise, reconstruction artifacts and CAD-world reference

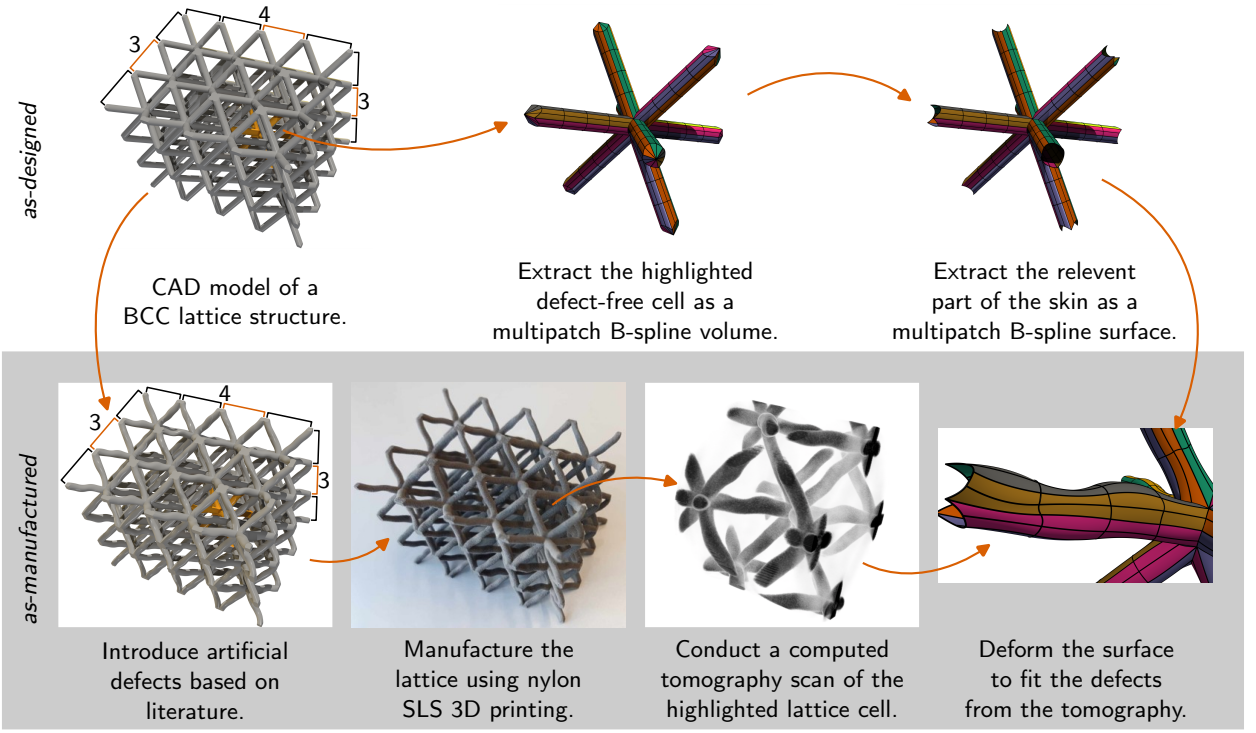


Figure 13: Illustration of the process used to design a realistic case. Starting from a built defect-free CAD model of an as-designed lattice structure, the process splits into two branches. The first branch involves extracting the relevant part of the skin from a volumetric CAD representation of a lattice cell as a multipatch B-spline surface. The second branch involves adding realistic defects to the CAD model, manufacturing the part with 3D printing, and performing a CT scan of a cell of the real sample. Finally, the surface fitting algorithm deforms the as-designed cell surface to match the as-manufactured cell CT scan.

transformation. However, in order to be able to compare the measured defects to a reference, it was decided to generate representative defects on a magnified sample to separate the scale of the generated defects to that of the real manufacturing defects. Synthetic defects were added to the initial CAD according to [40] on Selective Laser Melting (SLM) manufacturing. In [40], 2 types of defects were considered in diagonal beams: the deviation of the strut radius and the strut center axis offset. Since no information about their frequency was given, a rather smooth evolution of the defects was achieved using Free-Form Deformation (FFD) [55] on each beam. Following the workflow of Fig. 13, this as-manufactured geometry was fabricated in nylon using the Selective Laser Sintering (SLS) 3D printer SLS Fuse 1+ from Formlabs. After cleaning the printed part, a computed tomography (CT) scan was acquired in a Lab tomograph (RX Solutions, EasyTom 130). The acquisition parameters of the tomograph are summarized in Table 1. The obtained image was eventually binned, averaging 8^3 neighboring voxels. The corresponding volume image is show in Fig. 14(a).

The fitting algorithm was applied to create a geometric twin of the as-built cell from a real CT scan, starting from the as-designed multipatch B-spline surface. The first difficulty consisted in estimating the transformation (translation, rotation and scaling) between the sample frame and the image frame. In other words, this involved calibrating the camera model, which was known in the previous example. To calibrate these parameters, the position of the cell has to be identified in the image using image analysis tools. The following process was applied:

1. **Masking:** Initially, the exterior region of the cylinder inscribed within the tomographic volume, aligned along the e_z axis, was removed. This area often exhibits high uncertainties due to the rotational artifacts inherent to tomography, and excluding it helped focusing on the cell of interest, see Fig. 14(b).
2. **Gaussian Blur:** Next, a Gaussian blur with a standard deviation of 5 voxels was applied to the image. This choice of 5 voxels was considered small enough not to modify the geometry of the struts which was approximately

Parameters	Information
CT device	EasyTom 130 RX solutions
Source	130kV Hamamatsu Microfocus X-ray Source L9181-02
Voltage	119kV
Current	67μA
Detector	PaxScan Flat Panel X-ray Detector 2520DX
Encoding	16bits
Memory size of the reconstructed scan	8.8GB
Definition	1 816×1 816×1 339
Duration of the scan	25min
Voxel size	15.22μm
Target	Reflexion target
Filter	None
Source to detector	721.10mm
Source to object	86.42mm
Number of projections	1 472 (1 440 + 32 reference projections)
Angular amplitude	360°

Table 1
CT Scaning parameters

equal to 33 voxels in diameter, see Fig. 14(c). This step was necessary to remove outlier graylevels and help segmentation.

3. **Segmentation:** The image was then transformed into a binary image with 1 inside the lattice and 0 outside. The segmentation threshold was taken equal to average of background (air) and sample (lattice) graylevels, see Fig. 14(d).
4. **Skeletonization:** A skeletonization algorithm was then applied to reduce the beams forming the struts of the lattice structure to single voxel-wide lines. This transformation simplifies the complex 3D structures into more manageable forms for analysis, see Fig. 14(e).
5. **Intersection Extraction:** The intersection of beams was simply detected in the skeletonization graph as voxels connected to more than two neighboors, see Fig. 14(f).
6. **K-Means Clustering:** Intersections between multiple beams are often represented by several intersection voxels in the skeletonization graph. Given our knowledge of the initial geometry, the number of intersections is also known. Thus, a K-Means clustering algorithm was used to group these voxels in the same number of clusters than the known number of intersections, see Fig. 14(g).
7. **Least-Squares Optimization:** The parameters of the camera model were then found as the solution to a least-squares problem defined by the minimization of the sum of squared distances between the coordinates of the intersections found in the image (as described before) and the projections of the known physical intersections using the camera model.
8. **Solution Method:** To solve this minimization problem, a Gauss-Newton algorithm was employed, see Fig. 14(h). The focal length was considered fixed during the least-squares optimization. It was set to the unit distance in physical space as accurately provided by the tomographic reconstruction algorithm.

This process enables us to accurately estimate the parameters of the linear camera model, enabling the mapping between physical space and voxel space in 3D tomography.

Using this camera model, the as-designed B-spline surface was positioned in the image. This surface is composed of 48 B-spline patches of degree 2 in both isoparametric directions, with $4 \times 6 = 24$ control points each. Taking into account the conforming connectivity, this leads to $3 \times 786 = 2\,358$ DOFs for the geometry, at which one parameter is added for the learning of the virtual image, making a total of 2 359 DOFs. To compute the integral for the VIC energy, integration points were separated by approximately $1/2$ a voxel in the tangential direction of the surface, for about 1 voxel in the normal direction, for a total of 315 840 sampling points to be compared to the $\approx$ 8 million voxels in the image. For the membrane energy, 64 896 Gauss-Legendre integration points were used. The inter-patch energy was applied to all the patch interfaces except for the central ends of the beams. This is to stay consistent with the regularity of the initial CAD model where the intersection of the beams are C^0 lines. The weight parameter for the membrane strain energy α was set to 100 and the penalization weight of the inter-patch energy was set to $\beta = 10^{10}$.

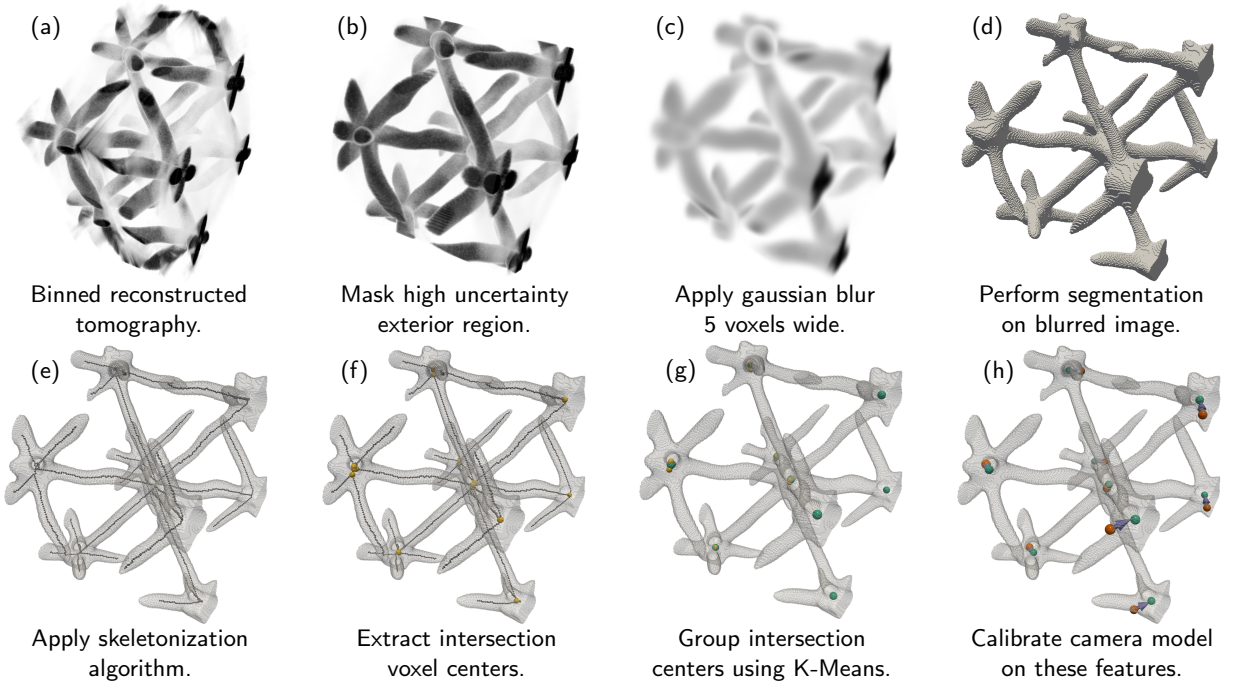


Figure 14: Workflow of camera model parameters calibration. In (a) to (g), lattice beam intersection positions are extracted from the reconstructed tomography. In (h), these features (in green) are compared with the known junctions of the as-designed lattice cell (in orange) to calibrate the camera model. The calibration is computed by a Gauss-Newton algorithm solving a nonlinear least-squares problem.

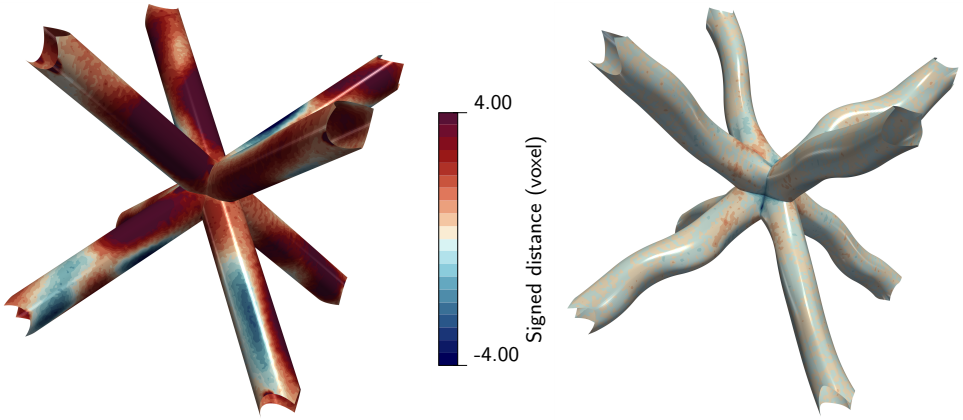
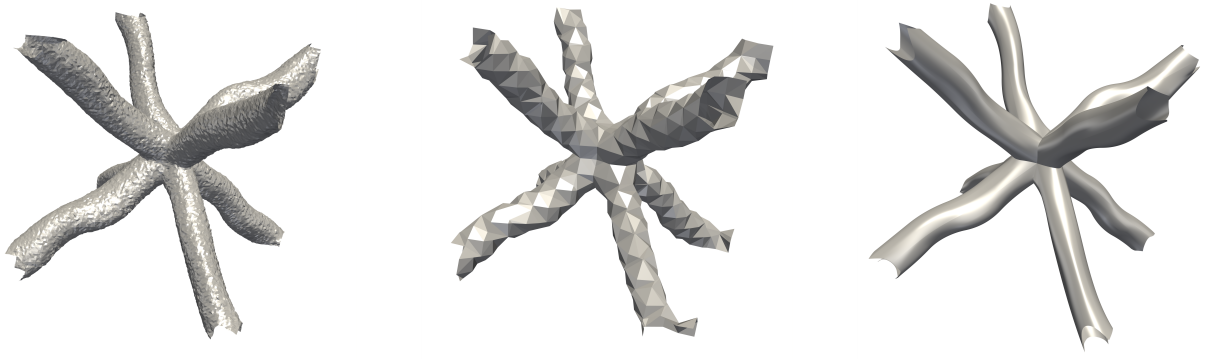


Figure 15: Signed distance of the multipatch B-spline surface to the as-manufactured of the CT scan computed by DIC. The initial as-designed distance map on the left can be compared to the fitted configuration on the right.

With all these parameters set, the surface fitting algorithm was applied to the BCC multipatch B-spline surface. It converged within 27 iterations to a relative error of 1%.

The signed distance to the actual boundary of the lattice is shown in Fig. 15 for the initial as-designed multipatch B-spline model on the left and after fitting on the right. Whereas the maximum initial geometric errors were about 6 voxels, the fitting was able to reduce these errors to a subvoxel value, which demonstrates the effectiveness of this

method on real images with representative defects. The higher values at the crossing of the beams can be explained by the surface neighborhood intersecting the foreground graylevels of the neighboring beams.



(a) Marching cubes triangular mesh of 63 972 nodes.

(b) Marching cubes triangular mesh of 1 031 nodes.

(c) Fitted multipatch B-spline surface of 786 nodes.

Figure 16: The first triangular mesh (a) was created using the marching cubes algorithm on the cropped CT scan from Fig. 14(a). The second one (b) was made by averaging 16^3 voxels in the image before applying the marching cubes algorithm. Both are compared to the fitted multipatch B-spline surface (c). The number of control points of the multipatch B-spline surface is two orders of magnitude below the first triangular mesh, making it more suitable for analysis. At similar number of DOFs, the difference in smoothness, and in fidelity to the data from the CT scan between (b) and (c) is evident here.

Last, the obtained B-spline representation was compared to a surface finite element mesh of linear triangles built using marching cubes [41]. Two different mesh sizes were prescribed, resulting in 63 972 and 1 031 nodes respectively (see Figs. 16a and 16b). In comparison, the fitted multipatch surface, see Fig. 16c, is made of 786 control points only. With a number of DOFs similar to that of the fitted spline representation, it is quite obvious that the regularity and the fidelity of the FE mesh is poor. Whereas to reach a similar level of fidelity, the FE requires two orders of magnitude more nodes in the surface mesh. To compute a mechanical response, a volume mesh would have to be computed and the difference of number of nodes between the FE and B-spline models, and so the computational cost of the corresponding simulations, would drastically increase. These results clearly illustrate the capacity of the proposed method to provide smooth and computationally light fitted B-spline models from real CT scans. In particular, it can be seen that the inter-patch energy $\mathrm{E}_{\mathrm{inter}}$ successfully prescribes C^1-continuity across patches. In addition to a fitted representation of the as-built geometry, the method also provides an explicit representation of the transformation between the as-designed and as-built lattices, as well as a distance map between them. Although beyond the scope of this paper, all these additional outputs represent precious pieces of information that could, for instance, be used to characterize precisely and quantitatively the manufacturing defects.

5. Conclusion

Recapitulating, we developed a CAD-based surface fitting approach that allows for the accurate and efficient measurement of the real boundary of mechanical structures or materials, given by a graylevel distribution in a 3D CT scan. More precisely, our method is capable of deforming a complex, analysis-suitable B-spline multipatch surface to match its actual counterpart observed in a volumetric image. To achieve this, we drew inspiration from the framework of active contour methods developed in computer vision [32, 64, 30], and proposed three main innovations to make it suitable for 3D and the context of solid mechanics. The first key point was to consider a data fitting metric based on the VIC method [23, 3], additionally incorporating virtual image learning as initiated in a previous contribution [46], and computing the integral in the physical domain. The combination of these elements allowed us to achieve both good accuracy and low computational cost, as it involves interrogating the image only within a local, thin layer around the surface to be measured. In this respect, it is worth noting that the proposed strategy extends the original VIC technique [23, 3], which differs from more recent alternatives, also referred to as VIC [14, 11], where standard Digital Image Correlation between a virtual and the complete real image is performed. Secondly, we complemented the VIC-based functional with the membrane strain energy of the surface model as the regularization metric. This approach

breaks from what is typically proposed in active contour methods [1, 64, 30] and template fitting approaches in mechanics [31, 14, 11]. It proved necessary in 3D, particularly given the fine scale of the defects to be measured, to avoid spurious tangential modes while keeping the normal component, driven by the VIC metric, unaffected. Thirdly, we incorporated an additional term in the functional to address the multipatch aspect of our model. Where appropriate, we ensured that the interface and neighboring control points remained aligned and equally spaced during the deformation of the surface, thereby maintaining C^1-continuity at the considered patch interfaces.

The developed approach was applied in this work to measure geometric defects in lattice structures produced by additive manufacturing. In particular, a lattice structure with a BCC pattern, subject to geometric defects typical of those encountered in metallic additive manufacturing [40], was fabricated and imaged, and the resulting data were used to evaluate our approach. After properly estimating the position of the CAD object in the image, we were able to deform the as-designed unit-cell model (composed of 8 struts, each consisting of 6 B-spline patches) to align with its as-manufactured counterpart. We also proposed a strategy for computing a distance indicator map (in voxel units) between the aligned CAD model and the as-manufactured surface described in the image. This indicator seems valuable as it allows for a better assessment of the accuracy of the method compared to relying solely on the VIC residual, as is often done. Based on this indicator, we achieved a fit of the true geometry with sub-voxel precision. Furthermore, we paid special attention to demonstrating the robustness of the membrane strain energy for regularization compared to more standard gradient or Laplacian-based regularizations.

Finally, thanks to the use of a conforming spline model, the developed method provides an explicit and compact representation of the as-manufactured geometry that shares the same support as the initial as-designed geometry, which should facilitate the quantitative assessment of geometric defects [40, 38, 33]. Furthermore, since the (skin) model is analysis-suitable, constructing a volume model adapted to IGA should be rather simple, allowing efficient numerical simulations [26, 42] to study the impact of defects on the mechanical behavior of the structure. These two points constitute interesting perspectives for this work, which will be investigated in the future using data from metallic lattice structures. Lastly, it is worth noting that the developed framework, which combines a VIC-type functional with membrane strain and inter-patch energies for regularization, is relatively generic and could be employed in other fields, particularly during refinement stages to improve accuracy, such as in biomedical applications [7, 62].

Acknowldegments: This research was supported by the French "Agence Nationale de la Recherche" under grant ANR-22-CE46-0007 (AVATAR). ANR is gratefully acknowledged. For the purpose of Open Access, a [CC-BY public copyright license] has been applied by the authors to the present document and will be applied to all subsequent versions up to the Author Accepted Manuscript arising from this submission.

Declaration of generative AI and AI-assisted technologies in the writing process: During the preparation of this work the authors used ChatGPT in order to improve language and readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] Ahlberg, J., Nordberg, K., and Wiklund, J. (2000). 3-D active contours attracted to planar structure. Publisher: Citeseer.

[2] Amani, Y., Dancette, S., Delroisse, P., Simar, A., and Maire, E. (2018). Compression behavior of lattice structures produced by selective laser melting: X-ray tomography based experimental and finite element approaches. *Acta Materialia*, 159:395–407.

[3] Baconnais, M., Réthoré, J., and François, M. (2020). Improvement of the digital image correlation close to the borders of an object. *Strain*, 56(3):e12340.

[4] Baghdadi, L., Steinman, D. A., and Ladak, H. M. (2005). Template-based finite-element mesh generation from medical images. *Computer methods and programs in biomedicine*, 77(1):11–21.

[5] Bartels, R. H., Beatty, J. C., and Barsky, B. A. (1995). *An introduction to splines for use in computer graphics and geometric modeling*. Morgan Kaufmann.

[6] Benedetti, M., Du Plessis, A., Ritchie, R., Dallago, M., Razavi, N., and Berto, F. (2021). Architected cellular materials: A review on their mechanical properties towards fatigue-tolerant design and fabrication. *Materials Science and Engineering: R: Reports*, 144:100606.

[7] Berberoğlu, E., Stoeck, C. T., Kozerke, S., and Genet, M. (2022). Quantification of left ventricular strain and torsion by joint analysis of 3d tagging and cine mr images. *Medical Image Analysis*, 82:102598.

[8] Blakey-Milner, B., Gradl, P., Snedden, G., Brooks, M., Pitot, J., Lopez, E., Leary, M., Berto, F., and Du Plessis, A. (2021). Metal additive manufacturing in aerospace: A review. *Materials & Design*, 209:110008.

[9] Bouclier, R. and Hirschler, T. (2022). *IGA: Non-conforming Coupling and Shape Optimization of Complex Multipatch Structures*. Wiley, 1 edition.

[10] Bouclier, R. and Passieux, J.-C. (2023). *IGA: Non-Invasive Coupling with FEM and Regularization of Digital Image Correlation Problems, Volume 2*. John Wiley & Sons.

[11] Calmettes, L., François, M., and Réthoré, J. (unpublished results https://hal.science/hal-04362475/). Shape measurements of lattice materials from few X-ray radiographs using the 3D Virtual Image Correlation method.

[12] Chen, L.-Y., Liang, S.-X., Liu, Y., and Zhang, L.-C. (2021). Additive manufacturing of metallic lattice structures: Unconstrained design, accurate fabrication, fascinated performances, and challenges. *Materials Science and Engineering: R: Reports*, 146:100648.

[13] Cottrell, J. A., Hughes, T. J. R., Bazilevskij, J. J., and Bazilevs, Y. (2009). *Isogeometric analysis: toward integration of CAD and FEA*. Wiley, Chichester.

[14] de Pastre, M.-A. and Quinsat, Y. (2023). Virtual volume correlation of lattice structures: From volumetric data to geometrical and dimensional defects identification. *Additive Manufacturing*, 61:103347.

[15] de Prenter, F., Verhoosel, C. V., van Brummelen, E., Evans, J., Messe, C., Benzaken, J., and Maute, K. (2020). Multigrid solvers for immersed finite element methods and immersed isogeometric analysis. *Computational Mechanics*, 65:807–838.

[16] Dierckx, P. (1995). *Curve and surface fitting with splines*. Oxford University Press.

[17] Divi, S. C., Verhoosel, C. V., Auricchio, F., Reali, A., and van Brummelen, E. H. (2022). Topology-preserving scan-based immersed isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 392:114648.

[18] Du Plessis, A., Yadroitsava, I., and Yadroitsev, I. (2020). Effects of defects on mechanical properties in metal additive manufacturing: A review focusing on x-ray tomography insights. *Materials & Design*, 187:108385.

[19] Düster, A., Sehlhorst, H.-G., and Rank, E. (2012). Numerical homogenization of heterogeneous and cellular materials utilizing the finite cell method. *Computational Mechanics*, 50:413–431.

[20] Echeta, I., Feng, X., Dutton, B., Leach, R., and Piano, S. (2020). Review of defects in lattice structures manufactured by powder bed fusion. *The International Journal of Advanced Manufacturing Technology*, 106:2649–2668.

[21] Echter, R., Oesterle, B., and Bischoff, M. (2013). A hierarchic family of isogeometric shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 254:170–180.

[22] François, M. L. M. (2022). Uncertainty of the virtual image correlation method. *International Journal for Numerical Methods in Engineering*, 123(18):4367–4390.

[23] François, M., Semin, B., and Auradou, H. (2010). Identification of the Shape of Curvilinear Beams and Fibers. *Applied Mechanics and Materials*, 24-25:359–364.

[24] Geng, L., Wu, W., Sun, L., and Fang, D. (2019). Damage characterizations and simulation of selective laser melting fabricated 3d re-entrant lattices based on in-situ ct testing and geometric reconstruction. *International Journal of Mechanical Sciences*, 157:231–242.

[25] Glaesener, R. N., Kumar, S., Lestringant, C., Butruille, T., Portela, C. M., and Kochmann, D. M. (2023). Predicting the influence of geometric imperfections on the mechanical response of 2d and 3d periodic trusses. *Acta materialia*, 254:118918.

[26] Hirschler, T., Bouclier, R., Antolin, P., and Buffa, A. (2024). Reduced order modeling based inexact FETI-Dp solver for lattice structures. *International Journal for Numerical Methods in Engineering*, 125(8):e7419.

[27] Hirschler, T., Bouclier, R., Duval, A., Elguedj, T., and Morlier, J. (2021). A new lighting on analytical discrete sensitivities in the context of isogeometric shape optimization. *Archives of Computational Methods in Engineering*, 28(4):2371–2408.

[28] Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1):185–203.

[29] Hughes, T. J. R., Cottrell, J. A., and Bazilevs, Y. (2005). Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195.

[30] Husham, S., Mustapha, A., Mostafa, S. A., Al-Obaidi, M. K., Mohammed, M. A., Abdulmaged, A. I., and George, S. T. (2020). Comparative Analysis between Active Contour and Otsu Thresholding Segmentation Algorithms in Segmenting Brain Tumor Magnetic Resonance Imaging. *Journal of Information Technology Management*, 12(Special Issue: Deep Learning for Visual Information Analytics and Management.):48–61.

[31] Jiang, Z., Mayeur, O., Witz, J., Lecomte-Grosbras, P., Dequidt, J., Cosson, M., Duriez, C., and Brieu, M. (2019). Virtual image correlation of magnetic resonance images for 3D geometric modelling of pelvic organs. *Strain*, 55(3):e12305.

[32] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

[33] Khristenko, U., Constantinescu, A., Le Tallec, P., and Wohlmuth, B. (2022). Statistically equivalent surrogate material models: Impact of random imperfections on the elasto-plastic response. *Computer Methods in Applied Mechanics and Engineering*, 402:115278.

[34] Kiendl, J., Bletzinger, K. U., Linhard, J., and Wüchner, R. (2009). Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198(49):3902–3914.

[35] Kiendl, J., Schmidt, R., Wüchner, R., and Bletzinger, K.-U. (2014). Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. *Computer Methods in Applied Mechanics and Engineering*, 274:148–167.

[36] Korkmaz, M. E., Gupta, M. K., Robak, G., Moj, K., Krolczyk, G. M., and Kuntoğlu, M. (2022). Development of lattice structure with selective laser melting process: A state of the art on properties, future trends and challenges. *Journal of Manufacturing Processes*, 81:1040–1063.

[37] Korshunova, N., Alaimo, G., Hosseini, S., Carraturo, M., Reali, A., Niiranen, J., Auricchio, F., Rank, E., and Kollmannsberger, S. (2021a). Bending behavior of octet-truss lattice structures: Modelling options, numerical characterization and experimental validation. *Materials & Design*, 205:109693.

[38] Korshunova, N., Papaioannou, I., Kollmannsberger, S., Straub, D., and Rank, E. (2021b). Uncertainty quantification of microstructure variability and mechanical behavior of additively manufactured lattice structures. *Computer Methods in Applied Mechanics and Engineering*, 385:114049.

[39] Krishnamurthy, V. and Levoy, M. (1996). Fitting smooth surfaces to dense polygon meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 313–324.

[40] Liu, L., Kamm, P., García-Moreno, F., Banhart, J., and Pasini, D. (2017). Elastic and failure response of imperfect three-dimensional metallic lattices: the role of geometric defects induced by Selective Laser Melting. *Journal of the Mechanics and Physics of Solids*, 107:160–184.

[41] Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169.

[42] Ma, K. and Bazilevs, Y. (2024). Isogeometric analysis of architected materials and structures. *Engineering with Computers*, pages 1–15.

[43] Nagy, A. P., IJsselmuiden, S. T., and Abdalla, M. M. (2013). Isogeometric design of anisotropic shells: optimal form and material distribution. *Computer Methods in Applied Mechanics and Engineering*, 264:145–162.

[44] Newman, T. S. and Yi, H. (2006). A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5):854–879.

[45] Passieux, J. and Bouclier, R. (2019). Classic and inverse compositional Gauss-Newton in global DIC. *International Journal for Numerical Methods in Engineering*, 119(6):453–468.

[46] Passieux, J.-C., Bouclier, R., and Weeger, O. (2023). Image-based isogeometric twins of lattices with virtual image correlation for varying cross-section beams. *International Journal for Numerical Methods in Engineering*, 124(10):2237–2260.

[47] Perney, A., Bordas, S., and Kerfriden, P. (2023). Nurbs-based surface generation from 3d images: spectral construction and data-driven model selection. *Journal of Computational Design and Engineering*, 10(4):1856–1867.

[48] Piegl, L. and Tiller, W. (1997). *The NURBS Book*. Monographs in Visual Communication. Springer Berlin Heidelberg, Berlin, Heidelberg.

[49] Qian, X. (2010). Full analytical sensitivities in nurbs based isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 199(29-32):2059–2071.

[50] Riva, L., Ginestra, P. S., and Ceretti, E. (2021). Mechanical characterization and properties of laser-based powder bed–fused lattice structures: a review. *The International Journal of Advanced Manufacturing Technology*, 113:649–671.

[51] Rouwane, A., Bouclier, R., Passieux, J.-C., and Périé, J.-N. (2021). Adjusting fictitious domain parameters for fairly priced image-based modeling: Application to the regularization of digital image correlation. *Computer Methods in Applied Mechanics and Engineering*, 373:113507.

[52] Rouwane, A., Doumalin, P., Bouclier, R., Passieux, J.-C., and Périé, J.-N. (2023). Architecture-Driven Digital Volume Correlation: application to the analysis of in-situ crushing of a polyurethane foam. *Experimental Mechanics*, 63(5):897.

[53] Schaedler, T. A. and Carter, W. B. (2016). Architected Cellular Materials. *Annual Review of Materials Research*, 46(1):187–210.

[54] Schillinger, D. and Ruess, M. (2015). The finite cell method: a review in the context of higher-order structural analysis of cad and image-based geometric models. *Archives of Computational Methods in Engineering*, 22:391–455.

[55] Sederberg, T. W. and Parry, S. R. (1986). Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4):151–160.

[56] Tirvaudey, M., Bouclier, R., Passieux, J.-C., and Chamoin, L. (2020). Non-invasive implementation of nonlinear isogeometric analysis in an industrial fe software. *Engineering Computations*, 37(1):237–261.

[57] Urick, B., Sanders, T. M., Hossain, S. S., Zhang, Y. J., and Hughes, T. J. (2019). Review of patient-specific vascular modeling: template-based isogeometric framework and the case for cad. *Archives of Computational Methods in Engineering*, 26:381–404.

[58] Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., and Yu, T. (2014). scikit-image: image processing in python. *PeerJ*, 2:e453.

[59] Verhoosel, C. V., Van Zwieten, G., Van Rietbergen, B., and de Borst, R. (2015). Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone. *Computer Methods in Applied Mechanics and Engineering*, 284:138–164.

[60] Wang, J., Zhou, G., Hillman, M., Madra, A., Bazilevs, Y., Du, J., and Su, K. (2021). Consistent immersed volumetric nitsche methods for composite analysis. *Computer Methods in Applied Mechanics and Engineering*, 385:114042.

[61] Wang, P., Lei, H., Zhu, X., Chen, H., and Fang, D. (2019). Influence of manufacturing geometric defects on the mechanical properties of alsi10mg alloy fabricated by selective laser melting. *Journal of Alloys and Compounds*, 789:852–859.

[62] Willems, R., Verberne, L., van der Sluis, O., and Verhoosel, C. V. (2024). Echocardiogram-based ventricular isogeometric cardiac analysis using multi-patch fitted nurbs. *Computer Methods in Applied Mechanics and Engineering*, 425:116958.

[63] Wu, Y., Fang, J., Wu, C., Li, C., Sun, G., and Li, Q. (2023). Additively manufactured materials and structures: A state-of-the-art review on their mechanical characteristics and energy absorption. *International Journal of Mechanical Sciences*, 246:108102.

[64] Yang, L., Xin, D., Zhai, L., Yuan, F., and Li, X. (2019). Active Contours Driven by Visual Saliency Fitting Energy for Image Segmentation in SAR Images. In *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 393–397.

[65] Zhang, Y., Bazilevs, Y., Goswami, S., Bajaj, C. L., and Hughes, T. J. (2007). Patient-specific vascular nurbs modeling for isogeometric analysis of blood flow. *Computer methods in applied mechanics and engineering*, 196(29-30):2943–2959.